



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



**Diseño y desarrollo de una arquitectura de Internet  
de las Cosas de Nueva Generación orientada al  
cálculo y predicción de índices compuestos aplicada  
en entornos reales**

Departamento de Comunicaciones  
*Universitat Politècnica de València*

Tesis presentada para la obtención del grado de  
*Doctor por la Universitat Politècnica de València*

Valencia, septiembre de 2022

Ignacio Lacalle Úbeda

Director:  
Prof. Carlos E. Palau Salvador



*A mis padres,  
por guiar el camino que me ha traído hasta aquí.*





# Resumen

El Internet de las Cosas (IoT) ha experimentado un gran crecimiento en los últimos años. El incremento en el número de dispositivos, una mayor miniaturización de la capacidad de computación y las técnicas de virtualización, han favorecido su adopción en la industria y en otros sectores. Asimismo, la introducción de nuevas tecnologías (como la Inteligencia Artificial, el 5G, el Tactile Internet o la Realidad Aumentada) y el auge del *edge computing* preparan el terreno, y formulan los requisitos, para lo que se conoce como Internet de las Cosas de Nueva Generación (NGIoT).

Estos avances plantean nuevos desafíos tales como el establecimiento de arquitecturas que cubran dichas necesidades y a la vez resulten flexibles, escalables y prácticas para implementar servicios que aporten valor a la sociedad. En este sentido, el IoT puede resultar un elemento clave para el establecimiento de políticas y la toma de decisiones. Una herramienta muy útil para ello es la definición y cálculo de indicadores compuestos, que representan un impacto en un fenómeno real a través de un único valor. La generación de estos indicadores es un aspecto promovido por entidades oficiales como la Unión Europea, aunque su automatización y uso en entornos de tiempo real es un campo poco explorado. Este tipo de índices deben seguir una serie de operaciones matemáticas y formalidades (normalización, ponderación, agregación...) para ser considerados válidos.

Esta tesis doctoral plantea la unión de ambos campos en alza, proponiendo una arquitectura de Internet de las Cosas de nueva generación orientada al servicio de cálculo y predicción de indicadores compuestos. Partiendo de la experiencia del candidato en proyectos de investigación europeos y regionales, y construyendo sobre tecnologías *open source*, se ha incluido el diseño, desarrollo e integración de los módulos de dicha arquitectura (adquisición de datos, procesamiento, visualización y seguridad) como parte de la tesis.

Dichos planteamientos e implementaciones se han validado en cinco escenarios diferentes, cubriendo cinco índices compuestos en entornos con requisitos dispares

siguiendo una metodología diseñada durante este trabajo. Los casos de uso están centrados en aspectos de sostenibilidad en entornos urbano y marítimo-portuario, pero se ha destacado que la solución puede ser extrapolada a otros sectores ya que ha sido diseñada de una manera agnóstica.

El resultado de la tesis ha sido, además, analizado desde el punto de vista de transferencia tecnológica. Se ha propuesto la formulación de un producto, así como una posible financiación en fases de madurez más avanzadas y su potencial explotación como elemento comercializable.

# Resum

La Internet de les Coses (IoT) ha experimentat un gran creixement en els últims anys. L'increment en el nombre de dispositius, una major miniaturització de la capacitat de computació i les tècniques de virtualització, han afavorit la seua adopció en la indústria i en altres sectors. Així mateix, la introducció de noves tecnologies (com la Intel·ligència Artificial, el 5G, la Internet Tàctil o la Realitat Augmentada) i l'auge del *edge computing* preparen el terreny, i formulen els requisits, per al que es coneix com a Internet de les Coses de Nova Generació (NGIoT).

Aquests avanços plantegen nous desafiaments com ara l'establiment d'arquitectures que cobrisquen aquestes necessitats i resulten, alhora, flexibles, escalables i pràctiques per a implementar serveis que aporten valor a la societat. Ací, el IoT pot resultar un element clau per a l'establiment de polítiques i la presa de decisions. Una eina molt útil en aquest sentit és la definició i càlcul d'indicadors compostos, que representen un impacte en un fenomen real a través d'un únic valor. La generació d'aquests indicadors és un aspecte promogut per entitats oficials com la Unió Europea, encara que la seua automatització i ús en entorns de temps real és un camp poc explorat. Aquest tipus d'índexs han de seguir una sèrie d'operacions matemàtiques i formalitats (normalització, ponderació, agregació...) per a ser considerats vàlids.

Aquesta tesi doctoral planteja la unió de tots dos camps en alça, proposant una arquitectura d'Internet de les Coses de nova generació orientada al servei de càlcul i predicció d'indicadors compostos. Partint de l'experiència del candidat en projectes d'investigació europeus i regionals, i construïnt sobre tecnologies *open source*, s'ha inclòs el disseny, desenvolupament i integració dels mòduls d'aquesta arquitectura (adquisició de dades, processament, visualització i seguretat) com a part de la tesi.

Aquests plantejaments i implementacions s'han validat en cinc escenaris diferents, cobrint cinc índexs compostos en entorns amb requisits dispersos seguint

una metodologia dissenyada durant aquest treball. Els casos d'ús estan centrats en aspectes de sostenibilitat en entorns urbà i marítim-portuari, però s'ha destacat que la solució pot ser extrapolada a altres sectors ja que ha sigut dissenyada d'una manera agnòstica.

El resultat de la tesi ha sigut, a més, analitzat des del punt de vista de transferència tecnològica. S'ha proposat la formulació d'un producte, així com un possible finançament en fases de maduresa més avançades i la seua potencial explotació com a element comercialitzable.

# Abstract

The Internet of Things (IoT) has experienced tremendous growth in recent years. The increase in the number of devices, greater miniaturization of computing capacity and virtualization techniques have favored its adoption in industry and other sectors. Likewise, the introduction of new technologies (such as Artificial Intelligence, 5G, Tactile Internet or Augmented Reality), together with the rise of *edge computing*, are paving the way, and formulating the requirements, for what is known as the Next Generation Internet of Things (NGIoT).

These advances pose new challenges such as the establishment of proper architectures that meet those needs and, at the same time, are flexible, scalable, and practical for implementing services that bring value to society. In this sense, IoT could be a key element for policy and decision making. A very useful tool for this is the definition and calculation of composite indicators, which represent an impact on a real phenomenon through a single value. The generation of these indicators is an aspect promoted by official entities such as the European Union, although their automation and use in real-time environments is a rather uncharted research field. This type of indexes must follow a series of mathematical operations and formalities (normalization, weighting, aggregation...) to be considered valid.

This doctoral thesis proposes the union of both fields, proposing a new generation Internet of Things architecture oriented to the calculation and prediction of composite indicators. Based on the candidate's experience in European and regional research projects, and building on *open source* technologies, the design, development and integration of the modules of such architecture (data acquisition, processing, visualization and security) has been included as part of the thesis.

These approaches and implementations have been validated in five different scenarios, covering five composite indexes in environments with disparate requirements following a methodology designed during this work. The use cases are focused on sustainability aspects in urban and maritime-port environments, but it has been highlighted that the solution can be extrapolated to other sectors as it

has been designed in an agnostic way.

The result of the thesis has also been analyzed from the point of view of technology transfer. A tentative product definition has been formulated, as well as a possible financing in more advanced stages of maturity and its potential exploitation as a marketable element.

# Agradecimientos

En primer lugar, quiero agradecer al Profesor Carlos E. Palau su eterna dirección profesional y académica. Desde hace más de ocho años he tenido la oportunidad de aprender y formarme bajo su tutela, por la cual me siento enormemente afortunado.

En segundo lugar, agradezco a todos mis compañeros del grupo de investigación SATRD por hacerme sentir siempre como en casa. Y en especial a Andreu, Álex y Rafa, que han sido mi principal soporte durante este periodo y quienes representan todo lo que aspiro ser a nivel profesional.

En el plano personal, deseo dar las gracias a mi hermano Miguel por ser un bastión de complicidad incondicional. A mis padres, Rafael y Vicenta, por haber hecho todo esto posible. Ellos inyectaron en mí la inquietud por el conocimiento, así como el valor de la constancia en cualquier proyecto. Ambas cosas están plasmadas en esta tesis, que espero esté a la altura de la educación que han sabido darme. Al resto de mi familia; a los que están y a los que ya no, muchas gracias por haberme apoyado tanto como lo habéis hecho.

A mis amigos, en especial a Carlos, por su inestimable ayuda en todos los vértices de mi vida, pero también a Nacho, Rosana, Javi, Migue, Laura, Vicente y los demás. Dicen que uno es la suma de dónde nace y de lo que vive. Ellos me han hecho vivir un sinfín de aventuras que han moldeado a la persona que soy hoy. Por tanto, son parte de esta tesis, de todo lo que ha venido y de lo que está por venir.

Por último, a la persona más especial de mi vida, Sonia; culpable directa de cualquier éxito que pueda alcanzar. Gracias por el cariño y la paciencia en todo este proceso, por ser una fuente constante de ánimo y desafío personal aun en momentos tan difíciles como una pandemia mundial.

Para acabar, aunque me temo que nunca entenderá estas líneas, a Luna. Muchas páginas de este documento jamás hubieran sido escritas sin la inspiración encontrada durante sus paseos.





# Índice

|   |           |
|---|-----------|
| Índice de figuras.....  | IV        |
| Índice de tablas .....  | XI        |
| Acrónimos .....   | XIII      |
| <b>1. Introducción .....</b>                                    | <b>1</b>  |
| 1.1. Contexto .....   | 1         |
| 1.2. Motivaciones .....   | 4         |
| 1.3. Objetivos de la tesis .....                                | 6         |
| 1.4. Principales aportaciones.....                              | 7         |
| 1.4.1. Artículos, Congresos y Jornadas.....                     | 7         |
| 1.4.2. Proyectos de investigación.....                          | 10        |
| 1.4.3. Desarrollo software .....                                | 12        |
| 1.5. Organización de la memoria.....                            | 13        |
| <b>2. Estado del arte tecnológico .....</b>                     | <b>15</b> |
| 2.1. Introducción.....  | 15        |
| 2.2. Arquitecturas NGIoT de referencia.....                     | 17        |
| 2.3. Edge computing en NGIoT .....                              | 34        |
| 2.4. Aprendizaje automático para indicadores compuestos .....   | 47        |
| 2.4.1. Predicción de series temporales en NGIoT .....           | 47        |
| 2.4.2. Aprendizaje federado ( <i>Federated Learning</i> ) ..... | 58        |
| 2.5. Otras tecnologías habilitadoras .....                      | 63        |
| 2.5.1. Virtualización y escalabilidad .....                     | 63        |
| 2.5.2. Seguridad en IoT .....                                   | 67        |
| 2.5.3. DevSecOps.....   | 73        |
| 2.5.4. Tactile Internet .....                                   | 79        |

|           |   |            |
|-----------|---|------------|
| <b>3.</b> | <b>Definición e implementación de la arquitectura .....</b> | <b>85</b>  |
| 3.1.      | Introducción.....   | 85         |
| 3.1.1.    | Requisitos .....  | 87         |
| 3.1.1.1.  | Requisitos técnicos .....                                   | 88         |
| 3.1.1.2.  | Requisitos funcionales .....                                | 90         |
| 3.1.1.3.  | Requisitos de servicio .....                                | 92         |
| 3.1.2.    | Metodología .....   | 94         |
| 3.2.      | Visión general .....  | 97         |
| 3.2.1.    | Principios de diseño.....                                   | 97         |
| 3.2.2.    | Estructura modular y vistas .....                           | 107        |
| 3.3.      | Adquisición de datos .....                                  | 111        |
| 3.3.1.    | Preprocesado y armonización de datos .....                  | 112        |
| 3.3.2.    | Concentración de información contextual.....                | 123        |
| 3.4.      | Procesamiento de datos.....                                 | 130        |
| 3.4.1.    | Persistencia de la información .....                        | 131        |
| 3.4.2.    | Ejecución de servicios .....                                | 139        |
| 3.5.      | Visualización .....   | 146        |
| 3.6.      | Seguridad y privacidad .....                                | 157        |
| 3.7.      | Solución integrada.....                                     | 168        |
| <b>4.</b> | <b>Cálculo y predicción de índices compuestos .....</b>     | <b>177</b> |
| 4.1.      | Estado de la técnica .....                                  | 177        |
| 4.1.1.    | Introducción .....  | 178        |
| 4.1.2.    | Construcción de índices compuestos .....                    | 183        |
| 4.1.2.1.  | Creación del marco inicial .....                            | 185        |
| 4.1.2.2.  | Operaciones matemáticas sobre los indicadores.....          | 192        |
| 4.1.2.3.  | Otras consideraciones.....                                  | 203        |
| 4.1.2.4.  | Representación de resultados .....                          | 204        |
| 4.1.3.    | Herramientas software disponibles.....                      | 206        |
| 4.2.      | Implementación de los servicios .....                       | 215        |
| 4.2.1.    | Consideraciones generales .....                             | 215        |
| 4.2.2.    | Cálculo de indicadores compuestos .....                     | 222        |
| 4.2.3.    | Predicción de índices .....                                 | 237        |
| 4.2.3.1.  | Predicción de indicadores cKPIs .....                       | 237        |
| 4.2.3.2.  | Propuesta de aprendizaje federado.....                      | 245        |
| 4.2.4.    | Servicios auxiliares.....                                   | 248        |
| <b>5.</b> | <b>Aplicación en entornos reales .....</b>                  | <b>263</b> |
| 5.1.      | Introducción.....   | 263        |
| 5.2.      | Metodología de aplicación y despliegue de la solución ..... | 267        |
| 5.2.1.    | Visión general .....  | 267        |
| 5.2.2.    | Fase I – Conceptual y de servicio .....                     | 268        |

|   |            |
|---|------------|
| 5.2.3. Fase II – Arquitectura IoT y conectores .....  | 270        |
| 5.2.4. Fase III – Configuración y resultados .....  | 273        |
| 5.3. Instanciación <i>cloud</i> en ciudad-puerto para análisis de congestión de tráfico.....                                      | 275        |
| 5.4. Instanciación <i>cloud</i> en puertos marítimos para análisis de impacto medioambiental.....                                 | 297        |
| 5.5. Instanciación <i>edge</i> para la evaluación rápida de impacto medioambiental a través de un <i>kit</i> IoT .....            | 326        |
| 5.6. Instanciación <i>cloud</i> en un entorno de terminal logística para predicción de distancia social en época de COVID-19..... | 339        |
| 5.7. Instanciación <i>híbrida</i> para la recomendación de asignación en transporte logístico por carretera .....                 | 350        |
| <b>6. Consideraciones hacia la transferencia tecnológica.....</b>   | <b>359</b> |
| 6.1. Perspectiva de financiación .....  | 360        |
| 6.2. Análisis del resultado explotable.....   | 364        |
| 6.3. Esbozo de un modelo de negocio .....   | 370        |
| <b>7. Conclusiones.....</b>   | <b>377</b> |
| 7.1. Grado de cumplimiento de objetivos.....  | 378        |
| 7.2. Conclusiones globales .....  | 382        |
| 7.3. Reflexiones sobre la innovación.....   | 388        |
| 7.4. Trabajos futuros.....  | 390        |
| <b>Referencias .....</b>  | <b>395</b> |
| <b>Apéndices .....</b>  | <b>433</b> |
| Apéndice A – Diagramas de clases del servicio CIC .....   | 434        |
| Apéndice B – Características de las fuentes de datos .....  | 438        |
| Apéndice C – Docker compose de instalación de la arquitectura .....   | 442        |

# Índice de figuras

|   |    |
|---|----|
| Figura 1.1: De IoT a Next Generation IoT. ....                              | 2  |
| Figura 1.2: Estructuración en bloques de los capítulos centrales .....      | 14 |
| Figura 2.1: Estructuración del estado del arte tecnológico en la tesis..... | 16 |
| Figura 2.2: Evolución de plataformas IoT 2015-2021 .....                    | 18 |
| Figura 2.3: Arquitectura FP7-IoT-A. ....                                    | 21 |
| Figura 2.4: Arquitectura WSO2 IoT.....                                      | 22 |
| Figura 2.5: Arquitectura RAMI4.0. ....                                      | 22 |
| Figura 2.6: Arquitectura IIRA. ....   | 23 |
| Figura 2.7: Arquitectura OpenFog.....                                       | 25 |
| Figura 2.8: Arquitectura RAMEC ECC-RA. ....                                 | 26 |
| Figura 2.9: Arquitectura LSP IoT 3D.....                                    | 26 |
| Figura 2.10: Arquitectura FAR EDGE.....                                     | 27 |
| Figura 2.11: Arquitectura AIOTI.....  | 28 |
| Figura 2.12: Arquitectura AIOTI detallada. ....                             | 29 |
| Figura 2.13: Comparación entre patrones de arquitectura.....                | 33 |
| Figura 2.14: Perspectiva de cloud computing (CC) .....                      | 34 |
| Figura 2.15: Continuo de computación (edge-fog-cloud computing) .....       | 35 |
| Figura 2.16: Esquemas EC: MEC, cloudlet y FC.....                           | 37 |
| Figura 2.17: Arquitectura de referencia plataforma OpenFog. ....            | 41 |
| Figura 2.18: Esquema de un nodo <i>fog</i> según OpenFog. ....              | 42 |
| Figura 2.19: Arquitectura LF Edge – EdgeX-Foundry. ....                     | 43 |
| Figura 2.20: Nodo fog Fledge.....   | 44 |
| Figura 2.21: Nodo fog FogAtlas. ....  | 45 |
| Figura 2.22: Nodo fog pasarela H2020 INTER-IoT.....                         | 45 |
| Figura 2.23: Clasificación de tipos de ML. ....                             | 47 |
| Figura 2.24: Tipos de modelos de regresión. ....                            | 49 |
| Figura 2.25: Perspectiva de predicción en series temporales .....           | 50 |
| Figura 2.26: Modelo ARIMA .....   | 53 |

|  |     |
|--|-----|
| Figura 2.27: Modelo Holt-Winter Exponential Smoothing.....                               | 54  |
| Figura 2.28: Modelo Facebook Prophet .....   | 54  |
| Figura 2.29: Modelo LSTM. ....   | 55  |
| Figura 2.30: Resumen del flujo de aprendizaje federado. ....                             | 59  |
| Figura 2.31: Comparación de opciones de virtualización .....                             | 65  |
| Figura 2.32: Ciclo del mecanismo OAuth .....   | 69  |
| Figura 2.33: Pasos en metodología DevOps .....   | 74  |
| Figura 2.34: Evolución hacia DevSecOps .....   | 77  |
| Figura 2.35: Estructura de comunicación en Tactile Internet. Imágenes: Flaticon .....    | 80  |
| Figura 3.1: Estructuración de requisitos para el diseño de la arquitectura .....         | 87  |
| Figura 3.2: Metodología de diseño de la arquitectura .....                               | 95  |
| Figura 3.3: Utilización de la metodología DevSecOps en el desarrollo de la tesis .....   | 96  |
| Figura 3.4: Patrones de diseño de arquitecturas (remarcado el elegido en la tesis) ..... | 98  |
| Figura 3.5: Subparadigmas basados en servicios (remarcado el elegido en la tesis) .....  | 99  |
| Figura 3.6: Perspectiva híbrida de despliegue de la solución .....                       | 103 |
| Figura 3.7: Multi-tenancia versus multi instancia en un sistema software.....            | 106 |
| Figura 3.8: Diseño de la arquitectura (vista Funcional) .....                            | 109 |
| Figura 3.9: Vista Informacional de la arquitectura .....                                 | 110 |
| Figura 3.10: Módulo de adquisición de datos dentro de la arquitectura propuesta.....     | 111 |
| Figura 3.11: Subcomponentes en el módulo de adquisición de datos .....                   | 112 |
| Figura 3.12: Perspectiva de elementos en NGSI .....                                      | 115 |
| Figura 3.13: Estructura de los agentes en la arquitectura IoT .....                      | 115 |
| Figura 3.14: Tipo de agente activo .....   | 116 |
| Figura 3.15: Tipo de agente pasivo .....   | 116 |
| Figura 3.16: Modelo de datos KeyPerformanceIndicator (FIWARE) .....                      | 119 |
| Figura 3.17: Ejemplo de modelo de datos propuesto - cKPI.....                            | 120 |
| Figura 3.18: Instalación de un agente mediante comandos .....                            | 122 |
| Figura 3.19: Agentes instalados funcionando según indicaciones .....                     | 122 |
| Figura 3.20: Flujo instalación de agentes sobre la arquitectura .....                    | 122 |
| Figura 3.21: Agentes instalados funcionando según indicaciones .....                     | 123 |
| Figura 3.22: Comparación entre concentradores de información.....                        | 126 |
| Figura 3.23: Esquema de gestor de contexto ORION. ....                                   | 127 |
| Figura 3.24: Dualidad <i>edge-cloud</i> del módulo de adquisición de datos .....         | 128 |
| Figura 3.25: Módulos de procesamiento de datos en la arquitectura propuesta.....         | 130 |
| Figura 3.26: Módulos de procesamiento de datos: división .....                           | 131 |
| Figura 3.27: Alternativas tecnológicas propuestas para almacenamiento de datos .....     | 136 |
| Figura 3.28: Alternativas tecnológicas propuestas para almacenamiento de datos .....     | 138 |
| Figura 3.29: Submódulo de ejecución de servicios de la arquitectura propuesta.....       | 139 |
| Figura 3.30: Resumen del análisis de opciones para la ejecución de servicios .....       | 143 |
| Figura 3.31: Perspectiva de las opciones de ejecución de servicios analizadas .....      | 143 |
| Figura 3.32: Orquestación simplificada propuesta en la arquitectura (imagen) .....       | 144 |
| Figura 3.33: Instrucción (comando) para la ejecución planificada del servicio.....       | 145 |

|   |     |
|---|-----|
| Figura 3.34: Módulo de visualización dentro de la arquitectura propuesta.....   | 146 |
| Figura 3.35: Vistas plataformas IoT industriales (I): Azure e IBM Watson IoT. Fuente: páginas oficiales de las compañías..... | 148 |
| Figura 3.36: Vistas plataformas IoT industriales (II): AWS IoT y MindSphere. Fuente: páginas oficiales de las compañías.....  | 149 |
| Figura 3.37: Vistas plataformas IoT industriales (III): Grafana y Freeboard. Fuente: páginas oficiales de las compañías.....  | 150 |
| Figura 3.38: Vistas plataformas IoT industriales (IV): Graphite y Wirecloud. Fuente: páginas oficiales de las compañías.....  | 151 |
| Figura 3.39: Herramienta de visualización de datos escogida: Kibana .....   | 151 |
| Figura 3.40: Estructura del módulo de visualización de la arquitectura .....  | 152 |
| Figura 3.41: Visualización específica para servicio CI (1) .....  | 154 |
| Figura 3.42: Visualización específica para servicio CI (2) .....  | 154 |
| Figura 3.43: Visualización específica para servicio CI (3) .....  | 155 |
| Figura 3.44: Visualización específica para servicio CI (4) .....  | 155 |
| Figura 3.45: Visualización específica para servicio CI (5) .....  | 155 |
| Figura 3.46: Visualización específica para servicio CI (6) .....  | 156 |
| Figura 3.47: Visualización específica para servicio CI (7) .....  | 156 |
| Figura 3.48: Módulo de visualización dentro de la arquitectura propuesta.....   | 157 |
| Figura 3.49: Esquema de componentes del módulo de seguridad y privacidad .....  | 157 |
| Figura 3.50: Seguridad en acceso de usuario: roles y permisos.....  | 158 |
| Figura 3.51: Esquema de funcionamiento de PBMS. ....  | 159 |
| Figura 3.52: Seguridad en acceso de usuario: roles y permisos.....  | 164 |
| Figura 3.53: Esquema modular detallado de la arquitectura propuesta.....  | 169 |
| Figura 3.54: Diagrama de secuencia de flujo del dato.....   | 170 |
| Figura 3.55: Diagrama de secuencia de datos desde gestor de contexto.....   | 171 |
| Figura 3.56: Diagrama de secuencia de carga de datos <i>batch</i> desde agente .....  | 171 |
| Figura 3.57: Diagrama de secuencia de la persistencia en el sistema .....   | 172 |
| Figura 3.58: Diagrama de secuencia de instalación y ejecución de servicios .....  | 172 |
| Figura 3.59: Diagrama de secuencia de la configuración de seguridad .....   | 173 |
| Figura 3.60: Diagrama de secuencia de la petición de acceso a recurso .....   | 173 |
| Figura 3.61: Exposición de diferentes escenarios de despliegue .....  | 174 |
| Figura 4.1: Semana de indicadores compuestos 2021 – JRC K4P.....  | 182 |
| Figura 4.2: Pasos para la construcción de un índice compuesto .....   | 184 |
| Figura 4.3: Flujo de decisión para la inclusión de indicadores .....  | 186 |
| Figura 4.4: Representación del CI – Ejemplo I .....   | 205 |
| Figura 4.5: Representación del CI – Ejemplo II.....   | 205 |
| Figura 4.6: Herramienta CACIS de la OCDE. ....  | 206 |
| Figura 4.7: Herramienta EViews de IHS Global Inxc. ....   | 207 |
| Figura 4.8: Herramienta CIM. ....   | 207 |
| Figura 4.9: Herramienta del proyecto CITYKeys. ....   | 208 |
| Figura 4.10: Herramienta Compind. ....  | 209 |

|  |     |
|--|-----|
| Figura 4.11: Herramienta CIAO Tool. ....   | 210 |
| Figura 4.12: Herramienta COIN (Excel). ....  | 211 |
| Figura 4.13: Herramienta CIF. ....   | 212 |
| Figura 4.14: Plataforma y dashboard Thinking Cities. ....                                | 213 |
| Figura 4.15: Herramienta COIN-R. ....  | 213 |
| Figura 4.16: Comparación del CIC con herramientas disponibles. ....                      | 214 |
| Figura 4.17: Acuerdo de datos de entrada y salida. ....                                  | 215 |
| Figura 4.18: Esquema del modelo de datos acordado de entrada (desde agentes) ....        | 219 |
| Figura 4.19: Variabilidad en relaciones número de entradas: salidas en agentes. ....     | 220 |
| Figura 4.20: Resumen de consideraciones en agentes para la ejecución del servicio. ....  | 221 |
| Figura 4.21: Mecanismos de extracción e inserción de datos por/desde el servicio. ....   | 223 |
| Figura 4.22: Personalización estructura del árbol cálculo del CI. ....                   | 224 |
| Figura 4.23: Espacio de configuración de datos técnicos para ejecutar el servicio. ....  | 225 |
| Figura 4.24: Instalación del servicio por comando. ....                                  | 226 |
| Figura 4.25: Estructura del servicio de cálculo de indicadores compuestos (CIC). ....    | 227 |
| Figura 4.26: Comprobación de la estructura de nodos e indicadores. ....                  | 228 |
| Figura 4.27: Construcción de la estructura del árbol en la lógica del servicio. ....     | 229 |
| Figura 4.28: Avance iterativo de ponderación y agregación en niveles del árbol. ....     | 230 |
| Figura 4.29: Avance iterativo de ponderación y agregación en niveles del árbol. ....     | 231 |
| Figura 4.30: Formato JSON de resultado de cálculo en cada nivel de nodo. ....            | 234 |
| Figura 4.31: Diagrama de clases (simplificado) del servicio desarrollado. ....           | 236 |
| Figura 4.32: Diagrama de secuencia de la actuación del servicio. ....                    | 236 |
| Figura 4.33: Diagrama de secuencia dentro del módulo de procesamiento de datos. ....     | 237 |
| Figura 4.34: Ámbitos de acción del servicio de predicción de índices compuestos. ....    | 238 |
| Figura 4.35: Descomposición de servicios de predicción. ....                             | 239 |
| Figura 4.36: Configuración del usuario para el servicio de predicción. ....              | 240 |
| Figura 4.37: Configuración de desarrollador de servicio de predicción. ....              | 241 |
| Figura 4.38: Estructura de funcionamiento de servicios de predicción. ....               | 242 |
| Figura 4.39: Instalación de servicios de predicción. ....                                | 243 |
| Figura 4.40: UI resultante para la observación del servicio (I). ....                    | 244 |
| Figura 4.41: UI resultante para la observación del servicio (II). ....                   | 244 |
| Figura 4.42: Propuesta de integración de aprendizaje federado en la arquitectura. ....   | 246 |
| Figura 4.43: Flujo de integración de servicios adicionales. ....                         | 249 |
| Figura 4.44: Servicio externo de normalización desde entorno remoto. ....                | 251 |
| Figura 4.45: Servicio de recomendaciones basada en reglas sobre indicadores (I). ....    | 253 |
| Figura 4.46: Servicio de recomendaciones basada en reglas sobre indicadores (II). ....   | 254 |
| Figura 4.47: Opciones de implementación servicio auxiliar índice de fiabilidad. ....     | 256 |
| Figura 4.48: Metodología de cálculo del porcentaje relativo de fiabilidad de datos. .... | 258 |
| Figura 4.49: Ejemplo del formato de la fiabilidad relativa de fuentes de datos. ....     | 259 |
| Figura 4.50: Interfaz de resultados del índice de fiabilidad (I). ....                   | 260 |
| Figura 4.51: Interfaz de resultados del índice de fiabilidad (II). ....                  | 261 |
| Figura 5.1: Temporalidad de los escenarios de validación. ....                           | 266 |

|  |     |
|--|-----|
| Figura 5.2: Fases de la metodología de aplicación y despliegue de la solución.....   | 267 |
| Figura 5.3: Actividades de la fase I en la metodología de despliegue.....  | 268 |
| Figura 5.4: Fases de la metodología de aplicación y despliegue de la solución.....   | 270 |
| Figura 5.5: Composición del árbol para el CI en los escenarios de aplicación.....  | 273 |
| Figura 5.6: Selección de métodos matemáticos en la interfaz.....   | 274 |
| Figura 5.7: Configuración de periodicidad de ejecución del servicio.....   | 274 |
| Figura 5.8: Vistas del puerto de Tesalónica y su posición respecto a la ciudad.....  | 275 |
| Figura 5.9: Actividades portuarias relacionadas con el escenario de validación.....  | 276 |
| Figura 5.10: Datos procedentes de los sensores RFID en las puertas del puerto (a) arcos RFID en la entrada al puerto de la puerta de camiones (16); (b) etiqueta RFID y placa de detección de vehículos en la puerta de vehículos comunes (10A)..... | 278 |
| Figura 5.11: Información aportada por la web TrafficThess para el cálculo del TCI contextualizando la posición de las puertas 16 y 10A del puerto de Tesalónica.....   | 279 |
| Figura 5.12: Información de barcos procesados por el Puerto de Tesalónica, ofrecida por la Autoridad Portuaria para el cálculo del TCI.....  | 280 |
| Figura 5.13: Distribución de estaciones Stratus Meteo Grecia y selección de la más apropiada para alimentar una fuente de datos para el cálculo del TCI.....   | 280 |
| Figura 5.14: EDA de los datos de puertas en el puerto de Tesalónica. (a) por mes y (b) por día de la semana.....   | 282 |
| Figura 5.15: Gráfica de número de vehículos entrantes por puerta 10A en TCI.....   | 282 |
| Figura 5.16: Gráfica de número de vehículos entrantes por puerta 16 en TCI.....  | 283 |
| Figura 5.17: Identificación de subíndices en el escenario TCI.....   | 284 |
| Figura 5.18: Arquitectura desplegada en el escenario TCI.....  | 286 |
| Figura 5.19: Modelo de datos de entrada del número de vehículos atravesando las puertas del puerto en el escenario TCI.....  | 287 |
| Figura 5.20: Representación de los datos procesados por el agente sobre tráfico en calles circundantes al puerto de Tesalónica en el escenario TCI.....  | 288 |
| Figura 5.21: Modelo de datos recibido por el anuncio de buques en el puerto y CSV preprocesado por parte del agente (para caso histórico).....   | 289 |
| Figura 5.22: Selección de métodos matemáticos en el escenario TCI.....   | 292 |
| Figura 5.23: Árbol de composición del indicador TCI usando la solución.....  | 292 |
| Figura 5.24: Precisión expresada en MAE del modelo predictivo usado en el TCI.....   | 293 |
| Figura 5.25: Precisión expresada en MAE del modelo predictivo usado en el TCI.....   | 294 |
| Figura 5.26: Resultado de visualización en escenario TCI (1).....  | 295 |
| Figura 5.27: Resultado de visualización en escenario TCI (2).....  | 296 |
| Figura 5.28: Resultado de visualización en escenario TCI (3).....  | 296 |
| Figura 5.29: Planteamiento contextual de PIXEL de cara al escenario PEI.....   | 298 |
| Figura 5.30: Puertos de validación PEI (Pireo, Tesalónica, Burdeos, Monfalcone).....   | 299 |
| Figura 5.31: Flujo de recopilación de literatura para el diseño del índice PEI.....  | 301 |
| Figura 5.32: Aspectos medioambientales en un entorno marítimo portuario.....   | 302 |
| Figura 5.33: Gráfica que muestra los aspectos medioambientales más relevantes para los puertos marítimos según encuestas del proyecto PIXEL.....   | 302 |



|  |     |
|--|-----|
| Figura 5.34: Estructura de subíndices e índice en el escenario PEI .....   | 303 |
| Figura 5.35: Fuente de datos sobre barcos a procesar por el puerto escenario PEI .....   | 304 |
| Figura 5.36: Fuente de datos vertido de residuos escenario PEI.....  | 305 |
| Figura 5.37: Formulario web datos de consumo eléctrico PEI .....   | 305 |
| Figura 5.38: Gestión de residuos terminal portuaria escenario PEI .....  | 306 |
| Figura 5.39: Reportes anuales de ruido escenario PEI .....   | 306 |
| Figura 5.40: Estación sensores ruido y luz para escenario PEI .....  | 307 |
| Figura 5.41: Contextualización de índices .....  | 311 |
| Figura 5.42: Encaje de la solución sobre plataforma PIXEL escenario PEI.....   | 313 |
| Figura 5.43: Integración de la solución sobre plataforma PIXEL escenario PEI.....  | 314 |
| Figura 5.44: Formato de datos agente de barcos en el escenario PEI .....   | 315 |
| Figura 5.45: Formato de datos residuos de barcos escenario PEI .....   | 316 |
| Figura 5.46: Datos de consumo eléctrico y gas para emisiones PEI .....   | 317 |
| Figura 5.47: Datos de residuos generados por terminal en escenario PEI .....   | 317 |
| Figura 5.48: Opciones de conectividad para implementar el agente de ruido y luz .....  | 318 |
| Figura 5.49: Integración de la solución sobre plataforma PIXEL escenario PEI.....  | 320 |
| Figura 5.50: Selección de métodos matemáticos para cálculo del CI PEI .....  | 321 |
| Figura 5.51: Configuración del árbol de cálculo subíndices e índice PEI .....  | 322 |
| Figura 5.52: Ejecución del servicio CI en el escenario PEI (muestra 2020) .....  | 323 |
| Figura 5.53: Ejecución del servicio CI índice de fiabilidad en el escenario PEI .....  | 323 |
| Figura 5.54: Resultado de generación de reporte con recomendaciones del PEI .....  | 324 |
| Figura 5.55: Utilización del servicio CI en escenario PEI en puerto .....  | 325 |
| Figura 5.56: Contexto del planteamiento del escenario de validación $\mu$ PEI .....  | 326 |
| Figura 5.57: Sensores seleccionados para el escenario de validación $\mu$ PEI. De izquierda a derecha: (a) sensor de CO <sub>2</sub> MG-811, (b) sensor de ruido KY-038, (c) sensor de iluminación KY-018..... | 329 |
| Figura 5.58: Artefactos físicos para la simulación de demostración escenario $\mu$ PEI .....   | 329 |
| Figura 5.59: Diseño de arquitectura de despliegue para el escenario $\mu$ PEI .....  | 331 |
| Figura 5.60: Montaje de estación de sensores para escenario de validación $\mu$ PEI.....   | 332 |
| Figura 5.61: Montaje y componentes del escenario $\mu$ PEI.....  | 334 |
| Figura 5.62: Despliegue de la arquitectura usando Kubernetes escenario $\mu$ PEI.....  | 335 |
| Figura 5.63: Despliegue de la arquitectura usando Kubernetes escenario $\mu$ PEI.....  | 336 |
| Figura 5.64: Selección de métodos matemáticos para cálculo del CI $\mu$ PEI .....  | 337 |
| Figura 5.65: Estructura del árbol para el cálculo del $\mu$ PEI.....   | 337 |
| Figura 5.66: Operaciones simuladas para la recreación del escenario $\mu$ PEI Aproximación del barco (luz, CO <sub>2</sub> , sonido de sirena).....  | 338 |
| Figura 5.67: Operaciones simuladas para la recreación del escenario $\mu$ PEI .....  | 338 |
| Figura 5.68: Resultado ejecución índice $\mu$ PEI.....   | 338 |
| Figura 5.69: Contexto de escenario CEDI - Puerto de Monfalcone.....  | 340 |
| Figura 5.70: Diseño de arquitectura de despliegue para el escenario CEDI .....   | 344 |
| Figura 5.71: Sensorización CO <sub>2</sub> escenario CEDI .....  | 345 |
| Figura 5.72: Esquema funcionamiento servicio PAS .....   | 346 |

|   |     |
|---|-----|
| Figura 5.73: Formularios de introducción de datos - PAS .....                     | 347 |
| Figura 5.74: Flujo de servicios en escenario CEDI.....                            | 348 |
| Figura 5.75: Estructura del árbol para el cálculo del CEDI.....                   | 348 |
| Figura 5.76: Interfaz específica de visualización - CEDI.....                     | 349 |
| Figura 5.77: Utilización de la simulación planteada – escenario CEDI.....         | 349 |
| Figura 5.78: Fases completadas en el escenario OAI.....                           | 350 |
| Figura 5.79: Representación del contexto del escenario OAI .....                  | 351 |
| Figura 5.80: Ciclo de servicios en el escenario OAI.....                          | 354 |
| Figura 5.81: Árbol tentativo de construcción de índice OAI.....                   | 356 |
| Figura 5.82: Relación de diseño del Gemelo Digital con la solución propuesta..... | 357 |
| Figura 5.83: Interfaz relacionada con el escenario OAI (1) .....                  | 358 |
| Figura 5.84: Interfaz relacionada con el escenario OAI (2).....                   | 358 |
| Figura 6.1: Niveles de madurez tecnológica - TRL .....                            | 360 |
| Figura 6.2: El valle de la muerte de la financiación tecnológica .....            | 362 |
| Figura 6.3: Instrumentos recientes de financiación de la Unión Europea.....       | 363 |
| Figura 6.4: Esbozo de producto explotable “CIC kit”.....                          | 365 |
| Figura 6.5: Esquema tentativo de comercialización por <i>tiers</i> .....          | 376 |
| Figura A.1: Diagrama de clases del servicio CIC - Nodee.....                      | 434 |
| Figura A.2: Diagrama de clases del servicio CIC - Edge.....                       | 435 |
| Figura A.3: Diagrama de clases del servicio CIC - KPI.....                        | 435 |
| Figura A.4: Diagrama de clases del servicio CIC - Tree .....                      | 436 |
| Figura A.5: Diagrama de clases del servicio CIC - Global.....                     | 437 |

# Índice de tablas

|   |     |
|---|-----|
| Tabla 2.1: Comparación entre arquitecturas monolíticas, SOA y microservicios.....       | 32  |
| Tabla 2.2: Diferencias en características entre MEC, cloudlets y FC.....                | 38  |
| Tabla 2.3: Factores para el análisis de modelos de ML en el estado del arte .....       | 51  |
| Tabla 2.4: Diferentes dimensiones de escalabilidad .....                                | 63  |
| Tabla 2.5: Mecanismos de seguridad de plataformas IoT .....                             | 71  |
| Tabla 2.6: Pasos, herramientas y métodos DevSecOps y su encaje en la tesis .....        | 77  |
| Tabla 3.1: Requisitos técnicos de la arquitectura IoT .....                             | 89  |
| Tabla 3.2: Requisitos funcionales de la arquitectura IoT .....                          | 91  |
| Tabla 3.3: Requisitos del servicio a ejecutar sobre la arquitectura IoT .....           | 92  |
| Tabla 3.4: Vistas y grupos de interés en la arquitectura .....                          | 108 |
| Tabla 3.5: Comparación entre bases de datos candidatas.....                             | 135 |
| Tabla 3.6: Tecnologías elegidas para el sistema de gestión de políticas.....            | 162 |
| Tabla 3.7: Reglas recomendadas relacionadas con la privacidad .....                     | 166 |
| Tabla 3.8: SW y HW recomendado para el despliegue de la arquitectura.....               | 175 |
| Tabla 4.1: Ventajas y desventajas de los índices compuestos. ....                       | 179 |
| Tabla 4.2: Tipos de valores ausentes .....  | 189 |
| Tabla 4.3: Métodos singulares de gestión de datos ausentes.....                         | 190 |
| Tabla 4.4: Tipos de métodos de ponderación.....   | 197 |
| Tabla 4.5: Campos establecidos en el modelo de datos acordado <i>cKPI</i> .....         | 217 |
| Tabla 4.6: Extracto de comparación entre modalidades de captura de datos.....           | 257 |
| Tabla 5.1: Escenarios de validación de la arquitectura propuesta .....                  | 266 |
| Tabla 5.2: Acceso a datos IoT cubierto en los casos de uso .....                        | 266 |
| Tabla 5.3: Agentes NGSI desarrollados en los escenarios de validación .....             | 272 |
| Tabla 5.4: Tabla resumen de datos en el escenario TCI.....                              | 281 |
| Tabla 5.5: Resumen de correlación entre las fuentes de datos en el escenario TCI        | 283 |
| Tabla 5.6: Elección de métodos matemáticos para el escenario TCI .....                  | 284 |
| Tabla 5.7: Equipamiento <i>hardware</i> utilizado en el escenario de validación TCI ... | 291 |
| Tabla 5.8: Tecnología <i>software</i> utilizada en el escenario de validación TCI.....  | 291 |

|   |     |
|---|-----|
| Tabla 5.9: Métricas de precisión en el escenario TCI sobre el total del data set ....   | 294 |
| Tabla 5.10: Tabla resumen de datos en el escenario PEI .....                            | 308 |
| Tabla 5.11: Elección de métodos matemáticos para el escenario PEI.....                  | 309 |
| Tabla 5.12: Resultado primera pregunta BAM en escenario PEI.....                        | 312 |
| Tabla 5.13: Equipamiento <i>hardware</i> virtualizado utilizado en escenario PEI.....   | 321 |
| Tabla 5.14: Tecnología <i>software</i> utilizada en el escenario de validación TCI..... | 321 |
| Tabla 5.15: Elección de métodos matemáticos para el escenario $\mu$ PEI.....            | 330 |
| Tabla 5.16: Reflexión sobre formalización del índice CEDI.....                          | 342 |
| Tabla 6.1: Análisis DAFO del producto explotable.....                                   | 368 |
| Tabla 6.2: Análisis PESTEL del producto explotable .....                                | 369 |
| Tabla 6.3: BMC – Esbozo de modelo de negocio.....                                       | 372 |
| Tabla 7.1: Grado de cumplimiento del objetivo 1.....                                    | 378 |
| Tabla 7.2: Grado de cumplimiento del objetivo 2.....                                    | 378 |
| Tabla 7.3: Grado de cumplimiento del objetivo 3.....                                    | 378 |
| Tabla 7.4: Grado de cumplimiento del objetivo 4.....                                    | 379 |
| Tabla 7.5: Grado de cumplimiento del objetivo 5.....                                    | 379 |
| Tabla 7.6: Grado de cumplimiento del objetivo 6.....                                    | 379 |
| Tabla 7.7: Grado de cumplimiento del objetivo 7.....                                    | 380 |
| Tabla 7.8: Grado de cumplimiento del objetivo 8.....                                    | 380 |
| Tabla 7.9: Grado de cumplimiento del objetivo 9.....                                    | 381 |
| Tabla 7.10: Grado de cumplimiento del objetivo 10.....                                  | 381 |
| Tabla 7.11: Publicación de artículos por secciones – validación de la innovación ..     | 389 |

# Acrónimos

|                |  |  |
|----------------|--|--|
| <b>1.5xIQR</b> | 1.5 x Inter Quartile Ranking               | 1,5 x Clasificación intercuartil                       |
| <b>5G</b>      | 5th Generation Mobile Communications       | Comunicaciones Móviles de 5 <sup>a</sup> generación    |
| <b>ABAC</b>    | Attribute Based Access Control             | Control de acceso basado en atributos                  |
| <b>ACID</b>    | Atomicity, Isolation, Durability           | Atomicidad, Consistencia, Aislamiento, Durabilidad     |
| <b>ACL</b>     | Access Control List                        | Lista de control de acceso                             |
| <b>ACP</b>     | Access Control Policies                    | Políticas de control de acceso                         |
| <b>ACR</b>     | Access Control Rules                       | Reglas de control de acceso                            |
| <b>AHP</b>     | Analytical Hierarchy Process               | Proceso de jerarquía analítica                         |
| <b>AI</b>      | Artificial Intelligence                    | Inteligencia Artificial                                |
| <b>AIOTI</b>   | Alliance for Internet of Things Innovation | Alianza para la Innovación en el Internet de las Cosas |
| <b>AIS</b>     | Automatic Identification System            | Sistema de identificación automática                   |
| <b>AMQP</b>    | Advanced Message Queuing Protocol          | Protocolo avanzado de colas de mensajes                |
| <b>ANSI</b>    | American National Standards Institute      | Instituto Nacional de Normalización de Estados Unidos  |
| <b>AP</b>      | Autoridad Portuaria                        | Autoridad Portuaria                                    |
| <b>API</b>     | Application Programming Interface          | Interfaz de programación de aplicaciones               |
| <b>AQL</b>     | Arango Query Language                      | Lenguaje de consulta de Arango                         |
| <b>ARIMA</b>   | AutoRegressive Integrated Moving Average   | Media móvil integrada autorregresiva                   |
| <b>ARM</b>     | Advanced RISC Machine                      | Máquina RISC avanzada                                  |

|                  |  |  |
|------------------|--|--|
| <b>BAM</b>       | Budget Allocation Method                         | Método de asignación del presupuesto             |
| <b>BBDD</b>      | Base(s) de Datos                                 | Base(s) de Datos                                 |
| <b>BLE</b>       | Bluetooth Low Energy                             | Bluetooth de baja energía                        |
| <b>BMC</b>       | Business Model Canvas                            | Lienzo del Modelo de Negocio                     |
| <b>BoD</b>       | Benefit of the Doubt                             | Beneficio de la duda                             |
| <b>BSON</b>      | Binary JSON                                      | JSON binario                                     |
| <b>CA</b>        | Conjoint Analysis                                | Análisis conjunto                                |
| <b>CC</b>        | Cloud Computing                                  | Computación en la nube                           |
| <b>CDTI</b>      | -  | Centro para el Desarrollo Tecnológico Industrial |
| <b>CE</b>        | Comisión Europea                                 | -  |
| <b>CEDI</b>      | Composed Estimated Density Indicator             | Indicador de densidad estimada compuesta         |
| <b>CI</b>        | Composite Indicator/Index                        | Indicador/Índice Compuesto                       |
| <b>CI/CD</b>     | Continuous Integration/Continuous Development    | Integración continua/desarrollo continuo         |
| <b>CIC</b>       | Composed Index Calculator                        | Calculador de índices compuestos                 |
| <b>CIF</b>       | Composite Index Framework                        | -  |
| <b>CIM</b>       | Composite Index Modeller                         | -  |
| <b>cKPI</b>      | Composed KPI                                     | KPI compuesto                                    |
| <b>CNN</b>       | Convolutional Neural Network                     | Red neuronal convolucional                       |
| <b>COAP</b>      | Constrained Application Protocol                 | Protocolo de aplicación restringido              |
| <b>COIN</b>      | CO-mposite IN-dicator                            | -  |
| <b>CORS</b>      | Cross-Origin Resource Sharing                    | Compartición de recursos entre orígenes cruzados |
| <b>CQL</b>       | Cassandra Query Language                         | Lenguaje de consulta de Cassandra                |
| <b>CRUD</b>      | Create, Read, Update, Delete                     | Crear, Leer, Actualizar, Borrar                  |
| <b>CSS</b>       | Cascading Style Sheets                           | Hojas de estilo en cascada                       |
| <b>CSV</b>       | Comma Separated Values                           | Valores separados por comas                      |
| <b>DAFO</b>      | Debilidades, Amenazas, Fortalezas, Oportunidades | -  |
| <b>DAG</b>       | Directed Acyclic Graph                           | Grafo acíclico dirigido                          |
| <b>DAST</b>      | Dynamic Application Security Testing             | Pruebas de seguridad de aplicaciones dinámicas   |
| <b>DB</b>        | Database   | Base de datos                                    |
| <b>DBN</b>       | Deep Belief Networks                             | Redes de creencia profundas                      |
| <b>DevOps</b>    | Development-Operations                           | Desarrollo-Operaciones                           |
| <b>DevSecOps</b> | Development-Security-Operations                  | Desarrollo-Seguridad-                            |

|             |                                     |                                  |
|-------------|-------------------------------------|----------------------------------|
|             |                                     | Operaciones                      |
|             |                                     | Tecnología de Libro Contable     |
| <b>DLT</b>  | Distributed Ledger Technology       | Distribuido                      |
| <b>DT</b>   | Digital Twin                        | Gemelo Digital                   |
| <b>DoS</b>  | Denial of Service                   | Denegación de servicio           |
| <b>EBT</b>  | -                                   | Empresa de Base Tecnológica      |
| <b>EC</b>   | Edge Computing                      | Computación de borde             |
| <b>EDA</b>  | Exploratory Data Analysis           | Análisis exploratorio de datos   |
|             | Environmental Impact Assessments of | Evaluación del impacto           |
| <b>EIAP</b> | Ports                               | ambiental de los puertos         |
| <b>ELK</b>  | Elasticsearch Logstash Kibana       | Elasticsearch Logstash Kibana    |
| <b>EMA</b>  | Exponential Moving Average          | Media móvil exponencial          |
|             |                                     | Consejo europeo de               |
| <b>ERC</b>  | European Research Council           | investigación                    |
|             |                                     | Planificación de los recursos de |
| <b>ERP</b>  | Enterprise Resource Planning        | la empresa                       |
|             | European Telecommunications         | Instituto Europeo de Normas de   |
| <b>ETSI</b> | Standards Institute                 | Telecomunicaciones               |
| <b>EWC</b>  | European Waste Catalogue            | Catálogo europeo de residuos     |
| <b>FC</b>   | Fog Computing                       | Computación en la niebla         |
| <b>FL</b>   | Federated Learning                  | Aprendizaje federado             |
|             |                                     | Protocolo de transferencia de    |
| <b>FTP</b>  | File Transfer Protocol              | archivos                         |
|             |                                     | Reglamento General de            |
| <b>GDPR</b> | General Data Protection Regulation  | Protección de Datos              |
|             |                                     | Servicio general de              |
| <b>GPRS</b> | General Packet Radio Service        | radiocomunicación por paquetes   |
|             |                                     | Sistema de posicionamiento       |
| <b>GPS</b>  | Global Positioning System           | global                           |
| <b>GUI</b>  | Graphic User Interface              | Interfaz gráfica de usuario      |
| <b>HE</b>   | Horizon Europe                      | Horizonte Europa                 |
|             |                                     | Lenguaje de marcado de           |
| <b>HTML</b> | HyperText Markup Language           | hipertexto                       |
|             |                                     | Protocolo de transferencia de    |
| <b>HTTP</b> | Hyper-Text Transfer Protocol        | hipertexto                       |
| <b>I/O</b>  | Input/Output                        | Entrada/Salida                   |
| <b>I+D</b>  | Investigación y Desarrollo          | -                                |
| <b>IA</b>   | Inteligencia Artificial             | -                                |
| <b>IaaS</b> | Infrastructure as a Service         | Infraestructura como servicio    |
|             | Interactive Application Security    | Pruebas interactivas de          |
| <b>IAST</b> | Testing                             | seguridad de las aplicaciones    |
| <b>ICT</b>  | Information and Communications      | Tecnologías de la Información y  |

|                |   |                                 |
|----------------|---|---------------------------------|
|                | Technologies                            | Comunicaciones                  |
| <b>IDE</b>     | Integrated Development Environment      | Entorno de desarrollo integrado |
|                | International Electrotechnical          | Comisión Electrotécnica         |
| <b>IEC</b>     | Commission                              | Internacional                   |
|                | Institute of Electrical and Electronics | Instituto de Ingenieros         |
| <b>IEEE</b>    | Engineers                               | Eléctricos y Electrónicos       |
|                |   | Consortio del Internet          |
| <b>IIC</b>     | Industrial Internet Consortium          | Industrial                      |
|                |   | Organización Marítima           |
| <b>IMO</b>     | International Maritime Organization     | Internacional                   |
| <b>IoT</b>     | Internet of Things                      | Internet de las Cosas           |
| <b>IP</b>      | Internet Protocol                       | Protocolo de Internet           |
|                |   | Organización Internacional de   |
| <b>ISO</b>     | International Standards Organization    | Normalización                   |
|                |   | Proveedor de servicios de       |
| <b>ISP</b>     | Internet Service Provider               | Internet                        |
| <b>JRC</b>     | Joint Research Centre                   | -                               |
|                |   | Notación de objetos de          |
| <b>JSON</b>    | JavaScript Object Notation              | JavaScript                      |
| <b>JWT</b>     | JSON Web Token                          | Testigo web JSON                |
| <b>K4P</b>     | Knowledge4Policy                        | -                               |
|                |   | Vecinos más cercanos (K-        |
| <b>KNN</b>     | K-Nearest Neighbours                    | Nearest Neighbours)             |
| <b>KPI</b>     | Key Performance Indicator               | Indicador clave de rendimiento  |
|                |   | Director (de referencia),       |
| <b>LCS</b>     | Lead, Composite, Sentiment              | Compuesto, de Sentimiento       |
| <b>LOPD</b>    | Ley Orgánica de Protección de Datos     | -                               |
| <b>LSTM</b>    | Long Short-Term Memory                  | Memoria a corto plazo           |
| <b>LTS</b>     | Long Term Storage                       | Almacenamiento a largo plazo    |
|                |   | Protocolo OMA Lightweight       |
| <b>LwM2M</b>   | OMA Lightweight M2M Protocol            | M2M                             |
| <b>LXC/LXD</b> | LinuX Container                         | Contenedor LinuX                |
| <b>MA</b>      | Moving Average                          | Media móvil                     |
| <b>MAE</b>     | Mean Absolute Error                     | Error Medio Absoluto            |
| <b>MAR</b>     | Missing At Random                       | Desaparecido al azar            |
|                |   | Desaparecido completamente al   |
| <b>MCAR</b>    | Missing Completely At Random            | azar                            |
|                |   | Computación de borde            |
| <b>MEC</b>     | Mobile/Multi-Access Edge Computing      | móvil/multi-acceso              |
| <b>ML</b>      | Machine Learning                        | Aprendizaje Automático          |
| <b>MLP</b>     | Multi-Layer Perceptron                  | Perceptrón multicapa            |
| <b>MQTT</b>    | Message Queing Telemetry Transport      | Transporte de telemetría de     |



|               |   |  |
|---------------|---|--|
|               |   | cola de mensajes   |
| <b>MVP</b>    | Minimum Value Product   | Producto de valor mínimo                                       |
| <b>NFC</b>    | Near Field Communication  | Comunicación de campo cercano                                  |
| <b>NFV</b>    | Network Function Virtualization                                 | Virtualización de funciones de red                             |
| <b>NGI</b>    | Next Generation Internet  | Internet de Nueva Generación                                   |
| <b>NGIoT</b>  | Next Generation Internet of Things                              | Internet de las Cosas de Nueva Generación                      |
| <b>NGSI</b>   | Next Generation Service Interfaces                              | Interfaces de servicio de próxima generación                   |
| <b>NIST</b>   | National Institute of Standards and Technology                  | Instituto Nacional de Normas y Tecnología                      |
| <b>NMAR</b>   | Not Missing At Random   | No se ha perdido al azar                                       |
| <b>OAI</b>    | Optimal Assignment Indicator                                    | Indicador de asignación óptima                                 |
| <b>OCDE</b>   | Organización para la Cooperación y el Desarrollo Económicos     | Organización para la Cooperación y el Desarrollo Económicos    |
| <b>OGC</b>    | Open Geospatial Consortium                                      | Consortio Geoespacial Abierto                                  |
| <b>OMA</b>    | Open Mobile Alliance  | Alianza móvil abierta  |
| <b>OPC-UA</b> | OPC Unified Architecture  | Arquitectura unificada de la fundación OPC                     |
| <b>OSM</b>    | OpenStreetMap   | -  |
| <b>OT</b>     | Operational Tools   | Herramientas operativas  |
| <b>PaaS</b>   | Platform as a Service   | Plataforma como servicio                                       |
| <b>PAP</b>    | Policy Administration Point                                     | Punto de administración de la política                         |
| <b>PAS</b>    | Port Activity Scenario  | Escenario de Actividades Portuarias                            |
| <b>PCA</b>    | Principal Component Analysis                                    | Análisis de componentes principales                            |
| <b>PCS</b>    | Port Community System   | Sistema comunitario portuario                                  |
| <b>PDP</b>    | Policy Decision Point   | Punto de decisión política                                     |
| <b>PEI</b>    | Port Environmental Index  | Índice medioambiental portuario                                |
| <b>PEP</b>    | Policy Enforcement Point  | Punto de aplicación de la política                             |
| <b>PESTEL</b> | Political, Economical, Social, Technological, Ecological, Legal | Político, económico, social, tecnológico, ecológico y jurídico |
| <b>PIP</b>    | Policy Information Point  | Punto de información política                                  |
| <b>PRP</b>    | Policy Retrieval Point  | Punto de recuperación de la política                           |

|                 |   |   |
|-----------------|---|---|
| <b>PUF</b>      | Physical Unclonable Function  | Función física inclonable   |
| <b>PYME</b>     | -   | Pequeña y Mediana Empresa   |
| <b>QoE</b>      | Quality of Experience   | Calidad de la experiencia   |
| <b>RAM</b>      | Random Access Memory  | Memoria de acceso aleatorio   |
| <b>RAN</b>      | Radio Access Network  | Red de acceso radioeléctrico  |
| <b>RASP</b>     | Runtime Application Self Protection   | Autoprotección de aplicaciones<br>en tiempo de ejecución                    |
| <b>RBAC</b>     | Role Based Access Control   | Control de acceso basado en<br>roles  |
| <b>REST</b>     | Representational State Transfer   | Transferencia de estados<br>Representacional                                |
| <b>RFID</b>     | Radio Frequency Identification  | Identificación por<br>radiofrecuencia                                       |
| <b>RIA</b>      | Research and Innovation Action  | Acción de investigación e<br>innovación                                     |
| <b>RNC</b>      | Radio Network Controller  | Controlador de la red de<br>radiocomunicación                               |
| <b>RNN</b>      | Recurrent Neural Network  | Red neuronal recurrente   |
| <b>RR</b>       | Reliability Rating  | Ratio de fiabilidad   |
| <b>SaaS</b>     | Software as a Service   | Software como servicio  |
| <b>SAML</b>     | Security Assertion Markup Language  | Lenguaje de marcado de<br>aserción de seguridad                             |
| <b>SARIMA</b>   | Seasonal AutoRegressive Integrated<br>Moving Average                            | Media móvil integrada<br>autorregresiva estacional                          |
| <b>SARIMAX</b>  | Seasonal Auto-Regressive Integrated<br>Moving Average with eXogenous<br>factors | Media móvil integrada<br>autorregresiva estacional con<br>factores eXógenos |
| <b>SAST</b>     | Static Application Security Testing   | Pruebas estáticas de seguridad<br>de las aplicaciones                       |
| <b>SCA</b>      | Software Composition Analysis   | Análisis de la composición del<br>software                                  |
| <b>SDN</b>      | Software-Defined Network  | Red definida por software   |
| <b>SGD</b>      | Stochastic Gradient Descent   | Descenso de gradiente<br>estocástico  |
| <b>SMA</b>      | Simple Moving Average   | Media móvil simple  |
| <b>SO</b>       | Sistema Operativo   | -   |
| <b>SOA</b>      | Service-Oriented Architecture   | Arquitectura orientada a<br>servicios                                       |
| <b>SOSA/SSN</b> | Semantic Sensor Network   | Red de sensores semánticos  |
| <b>SQL</b>      | Structured Query Language   | Lenguaje de consulta<br>estructurado  |
| <b>SRM</b>      | Sparse Representation Model   | Modelo de representación  |

|              |   |   |
|--------------|---|---|
|              |   | dispersa  |
| <b>SSH</b>   | Secure Shell                                    | Shell seguro  |
| <b>SSL</b>   | Secure Socket Layers                            | Capas de sockets seguros                                    |
| <b>SSO</b>   | Single Sign-On                                  | Inicio de sesión único                                      |
| <b>STH</b>   | Short Term History                              | Historia a corto plazo                                      |
| <b>STLF</b>  | Short Term Load Forecasting                     | Previsión de carga a corto plazo                            |
|              | Simple/Streaming Text Oriented                  | Protocolo de mensajería orientada al texto simple/de flujo. |
| <b>STOMP</b> | Messaging Protocol.                             |   |
| <b>SVM</b>   | Support Vector Machine                          | Máquina de vector de soporte                                |
| <b>SVR</b>   | Support Vector Regression                       | Regresión de vector de soporte                              |
| <b>TCI</b>   | Traffic Congestion Index                        | Índice de congestión del tráfico                            |
|              |   | Protocolo de Control de Transmisión                         |
| <b>TCP</b>   | Transmission Control Protocol                   | Tecnologías Habilitadoras Digitales                         |
| <b>THD</b>   | -   | Autoridad Portuaria de Tesalónica                           |
| <b>THPA</b>  | Thessaloniki Port Authority                     |   |
| <b>TI</b>    | Tactile Internet                                | Tactile Internet  |
|              |   | Tecnologías de la Información y Comunicaciones              |
| <b>TIC</b>   | -   | Seguridad de la capa de transporte                          |
|              | Transport Layer Security                        | Transparencia, precisión, credibilidad, compromiso          |
| <b>TLS</b>   | Transparency, Accuracy, Credibility, Engagement | Nivel de preparación tecnológica                            |
| <b>TRACE</b> |   | Redes sensibles al tiempo                                   |
|              | Technology Readiness Level                      | -   |
| <b>TRL</b>   |   | Conector universal de IoT                                   |
| <b>TSN</b>   | Time Sensitive Networking                       | Lenguaje de modelado unificado                              |
| <b>UE</b>    | Unión Europea                                   | Localizador uniforme de recursos                            |
| <b>UIC</b>   | Universal IoT Connector                         | Bus serie universal   |
| <b>UML</b>   | Unified Modeling Language                       | Tiempo universal coordinado                                 |
|              | Uniform Resource Locator                        | Formato de transformación                                   |
| <b>URL</b>   |   | Unicode de 8 bits   |
| <b>USB</b>   | Universal Serial Bus                            | Banda Ultra Ancha   |
| <b>UTC</b>   | Coordinated Universal Time                      | Red privada virtual   |
|              |   | Realidad virtual  |
| <b>UTF-8</b> | 8-bit Unicode Transformation Format             | Consorcio de la World Wide Web                              |
| <b>UWB</b>   | Ultra Wide Band                                 |   |
| <b>VPN</b>   | Virtual Private Network                         |   |
| <b>VR</b>    | Virtual Reality                                 |   |
|              |   |   |
| <b>W3C</b>   | World Wide Web Consortium                       | Web   |

|              |   |   |
|--------------|---|---|
| <b>WEF</b>   | World Economic Forum                      | Foro Económico Mundial                              |
| <b>WMA</b>   | Weighted Moving Average                   | Media móvil ponderada                               |
| <b>WSDL</b>  | Web Services Description Language         | Lenguaje de descripción de servicios web            |
| <b>XACML</b> | eXtensible Access Control Markup Language | Lenguaje de marcado de control de acceso eXtensible |
| <b>XAI</b>   | Explainable AI                            | IA explicable                                       |

# Capítulo 1

## Introducción

### 1.1.Contexto

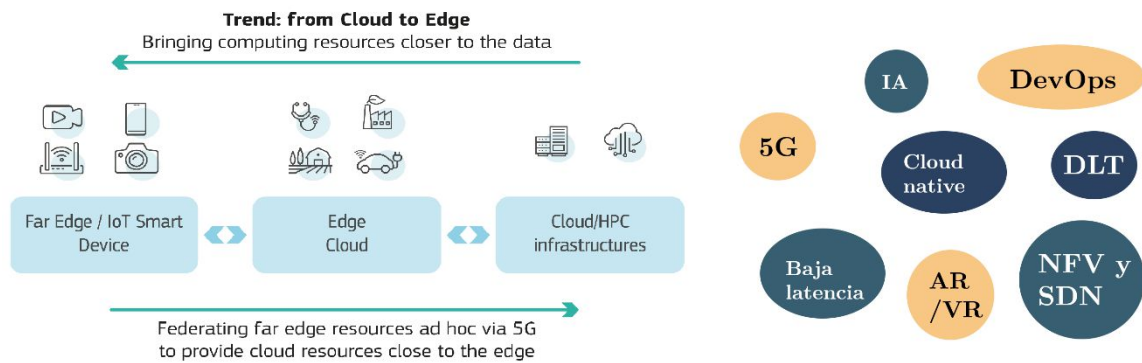
Alrededor del 90% de la información existente ha sido generada en los últimos tres años [4]. Buena parte de este crecimiento exponencial se debe al auge del denominado Internet de las Cosas (de ahora en adelante, IoT, del inglés *Internet of Things*). Este concepto representa el paradigma tecnológico en el que dispositivos heterogéneos (sensores y actuadores –“cosas”, pasarelas, aparatos inteligentes, etiquetas RFID/NFC, equipos de computación local, servidores en la nube, etc.), y componentes software forman un ecosistema que coopera para proveer de servicios físicos avanzados a los seres humanos.

El IoT (como tecnología y como campo de mercado) ha experimentado un enorme crecimiento durante los últimos años y en estos momentos se encuentra en una fase de consolidación y evolución. De hecho, en relación al volumen de datos provenientes de IoT, se estima un total de más de 850ZB procedentes de más de 40.000 millones de dispositivos para 2025 [1].

Además de la cantidad, el tipo (tanto estructurados como, en su mayoría, no

estructurados [2]) y volumen de estos datos está en aumento a medida que se enriquecen, debido a la disrupción de nuevas tecnologías como el acceso de banda ancha ubicuo (gracias al 5G), la realidad aumentada, los mecanismos ágiles DevOps, el Tactile Internet, los progresos en robótica e inteligencia artificial y el nuevo equipamiento háptico [17]. Estas tecnologías, junto al avance en la miniaturización, el aumento de la capacidad computacional en equipos cada vez más pequeños, la asequibilidad de la tecnología y la evolución de las capacidades virtualización, preparan el terreno, y formulan los requisitos, para lo que se conoce como Internet de las Cosas de Nueva Generación [18].

Este paso adelante en el contexto del IoT trae consigo nuevos desafíos y retos que están estudiándose en la actualidad, centrados en primer plano en el desarrollo de nuevos paradigmas de arquitecturas escalables. La mayoría de las plataformas están basadas en la nube (*cloud*), pocas de ellas admiten arquitecturas de *edge computing* o en el continuo de computación. Hoy en día no existe una clara referencia de implementación de dichos sistemas para entornos descentralizados.



**Figura 1.1:** De IoT a Next Generation IoT. Fuente: [549]

Por otro lado, existe también la tendencia a una digitalización basada en IoT para comprender eventos de la realidad y tomar decisiones en consecuencia. Se ha observado (perspectiva compartida por el doctorando) que, en la actualidad, debido a la gran disponibilidad de equipamiento “generador” de datos (p.ej., sensores), existe una cantidad de información desestructurada y sin conjugar que debe ser puesta en común para establecer políticas de interacción entre diferentes agentes. Disponiendo de dichas políticas, sería factible mejorar las sinergias entre ellos, siendo capaces de atacar de una manera conjunta los desafíos actuales y futuros. Sin embargo, existen ciertas barreras que dificultan la adopción de estas nuevas tecnologías IoT para la derivación de políticas y tomas de decisiones, como la inexistencia de regulaciones comunes, metodologías claras y transparentes o la reactancia por parte de entidades a compartir datos necesarios.

Conviene destacar que estas reflexiones contextuales se encuentran considerablemente marcadas por el sector donde se han desarrollado los escenarios de validación de la tesis (urbano y marítimo-portuario, así como la interfaz entre ellos). No obstante, estas premisas son igualmente aplicables a otros entornos (públicos y privados). De hecho, el objetivo de esta tesis ha sido plantear una

arquitectura IoT agnóstica al caso de uso y al dominio de aplicación.

Para poder establecer políticas efectivas que superen las barreras anteriores, una herramienta muy útil es la definición de indicadores sobre los que apoyarse. Para ello, el candidato propone el uso de indicadores compuestos (representado por las siglas CI, del inglés *Composite Indicator*). Los CIs se basan en la combinación matemática de indicadores individuales para representar diferentes dimensiones, modelando un evento complejo donde no existe una manera obvia de agregar y ponderar en un único valor ilustrativo las diferentes partes que lo componen [3]. Mientras que la simplificación de fenómenos complejos en un solo valor puede suponer una pérdida de información (y puede enviar mensajes engañosos y poco sólidos si está mal construida y se interpreta de forma errónea), permitirá a entidades (público, organismos gubernamentales, ayuntamientos, etc.) tener una visión clara del impacto actual, estableciendo clasificaciones, análisis oportunos, comparación entre diferentes escenarios y entre entidades y construir políticas sobre dicho valor único. Según estudios liderados por entidades oficiales, este tipo de soluciones son útiles no solo para monitorizar el progreso y rendimiento sobre un área concreta, sino para establecer dichas políticas [4]. De hecho, el uso de estos indicadores para la toma de decisiones y la comparación de rendimiento o impacto es una tendencia en alza durante los últimos años. Tanto es así que la Comisión Europea (a través de su *Joint Resarch Centre*) ha creado un grupo específico para elaborar guías, metodologías e incluso herramientas para el establecimiento y cálculo de índices compuestos [5].

En este punto, el Internet de las Cosas está llamado a ejercer de elemento habilitador clave. Sería muy útil disponer de una arquitectura que contemplara las premisas anteriores y permitiera la definición, cálculo, análisis y predicción de dichos indicadores para establecer políticas, baremos, medidas de control o de evolución temporal de impacto o rendimiento de cara a aportar un valor a la sociedad. Hoy en día, no existe una solución escalable que ataque este aspecto. Es por ello que la arquitectura propuesta se nutre de información recogida de fuentes de datos heterogéneas (IoT), basada en tecnologías *open source* y en una estructura tecnológica robusta para poder calcular dicho índice correctamente.

Por otro lado, un elemento contextual relevante en esta tesis doctoral ha sido el análisis de la situación actual de transferencia tecnológica en la Unión Europea. Se ha constatado la existencia de brechas significativas a la hora de trasladar esfuerzos de investigación (principalmente académica) en productos explotables en el mercado, reduciendo la capacidad (por parte de la investigación) de aportar un valor a la sociedad y a las empresas. En especial, los organismos oficiales han de realizar esfuerzos, especialmente en lo concerniente a la financiación (clave para este emprendimiento) en determinadas fases de madurez tecnológica.

De hecho, existe consenso en la actualidad en que la excelencia de investigación en la Unión Europea no traslada (transfiere) a un suficiente nivel de adopción de la innovación y de impacto en el mercado [6], llegando a haberse acuñado como término “la paradoja europea”.

## 1.2.Motivaciones

La reciente explosión de sistemas *Big Data* y la toma de consciencia del valor que aportan dichas herramientas ha llevado a una creciente utilización de gran cantidad de ellos para la toma de decisiones (empresariales, de gobernanza...). En este sentido, el Internet de las Cosas (IoT) está llamado a jugar un papel clave, ya que en su propia definición se encuentra patente la provisión de datos heterogéneos, de diferente índole y origen para su explotación de diversas maneras. No obstante, en muchas ocasiones se hace difícil “distinguir el grano de la paja” (valga la expresión); impidiendo en algunas ocasiones tomar en consideración la evolución de muchos factores al mismo tiempo en complejos cuadros de mando. Es por ello que una solución como la planteada en esta tesis podría facilitar esta tarea, aportando a la vez una serie de ventajas que se listan a continuación. Debe entenderse el listado inferior como las motivaciones para la creación de una arquitectura y servicio como el que articula esta tesis: “*Diseño y desarrollo de una arquitectura de Internet de las Cosas de Nueva Generación orientada al cálculo y predicción de índices compuestos aplicada en entornos reales*”:

- Habilitar una toma de decisiones ágil basada en la interpretación de un único valor (índice compuesto). Del mismo modo, estos índices permitirían el análisis de la efectividad de dichas medidas tomadas para mejorarlo, estableciendo así una suerte de sonda de control de la evolución temporal. Asimismo, proporciona la oportunidad para observar cómo impactaría una acción particular en dicho índice, introduciendo datos para su simulación.
- Adicionalmente, permitiría la comparación entre diferentes entidades y el establecimiento de rankings de rendimiento. Por otro lado, la adquisición de datos para popular dichos índices podría efectuarse de tal manera que pudiera cuantificarse *ex-ante* qué efecto tendría sobre dicho índice la introducción de una determinada medida/política. Finalmente, la introducción de la predicción de los índices podría ayudar a las entidades a actuar preventivamente ante una posible deriva del valor estimado de la métrica.
- Dotar de la posibilidad de utilización de estos índices para la concesión de licitaciones, realización de auditorías de una forma más ágil y continua... así como aplicaciones similares que podrían contribuir para asegurar y fomentar políticas (por ejemplo, relacionadas con los objetivos de desarrollo sostenible) por parte de administraciones públicas.
- Evolucionar desde un modelo basado en informes estáticos, documentación y evaluaciones cualitativas o basadas en encuestas a un modelo automatizado, tecnológico, mucho más fiable y preciso, con una tasa de refresco mucho más alta y menos costoso tanto inicialmente como a largo plazo.

Desde el punto de vista técnico, la NO existencia de las siguientes herramientas impulsó el planteamiento de esta tesis doctoral, haciendo uso del



conocimiento del candidato de sistemas de Internet de las Cosas:

- Herramienta especializada en cálculo, predicción e interpretación de índices compuestos. Existe una gran cantidad de plataformas IoT disponibles en el mercado, incluso accesibles en abierto para la comunidad, pero ninguna de ellas se encuentra diseñada y orientada en su conjunto para el cálculo y predicción de dichos índices compuestos.
- Herramienta integral, cubriendo todas las acciones necesarias para dicha interpretación, desde la recogida de datos, hasta su concentración, definición del indicador, aplicación del cálculo y observación en un entorno amigable. Se han hallado utilidades parciales para los procesos matemáticos necesarios (p.ej. normalización), así como plataformas de concentración de datos heterogéneos de propósito general, pero no existe un sistema que aúne todo lo anterior en un mismo entorno tecnológico.
- Una metodología, acompañada con herramienta de soporte, para la definición de indicadores compuestos y los datos que deben nutrirlos. Estos procesos son, actualmente, manuales y muy dependientes de cada caso, requiriendo formas únicas de manejar datos e índices, no existiendo un planteamiento agnóstico como el que se ha desarrollado en esta tesis doctoral.

Por último, pueden enumerarse también las motivaciones académicas que llevaron al candidato a emprender esta tesis doctoral, y que se han mantenido vigentes durante los tres años de desarrollo de la misma:

- Aplicación de los conocimientos adquiridos dentro del grupo de investigación SATRD (Sistemas y Aplicaciones en Tiempo Real Distribuidos). Establecimiento de nuevas líneas de investigación dentro del grupo tales como Tactile Internet, *Edge Computing* o índices compuestos.
- Especialización en el diseño de arquitecturas de IoT y especialmente de IoT de nueva generación (NGIoT – *Next Generation IoT*) para casos prácticos, cubriendo necesidades expresadas por grupos de interés, incluyendo nativamente aspectos como DevSecOps, AI o conceptos nativos de *cloud* en entornos edge. Continuación de aplicación de técnicas de interoperabilidad en el contexto de Internet de las Cosas.
- Profundización de la utilización de análisis y predicción de series temporales sobre datos heterogéneos dentro de sistemas IoT. Aplicación de métodos predictivos sobre datos recogidos en tiempo real dentro de un mismo entorno tecnológico (plataforma).
- Fortalecimiento de la capacidad de transferencia tecnológica por parte del candidato y del grupo de investigación, orientando el resultado de investigaciones técnicas (bajo nivel de madurez) a una posible explotación más allá del entorno académico-universitario.

## 1.3. Objetivos de la tesis

### Hipótesis

La hipótesis que se planteó al comienzo de la tesis doctoral es la siguiente: “El Internet de las Cosas puede y debe convertirse en el habilitador ideal para la comprensión de eventos complejos mediante índices compuestos. Las nuevas técnicas y plataformas surgidas por el auge del Internet de las Cosas, tanto a nivel de adquisición de datos heterogéneos (sensorización), como a nivel de técnicas de procesamiento y explotación de los mismos, son catalizadores para automatizar la recopilación y representación de información útil y con criterio”.

### Objetivo general

El propósito general de esta tesis doctoral es el de diseñar y desarrollar un sistema abierto basado en el Internet de las Cosas capaz de procesar información en tiempo real desde diferentes fuentes y diferentes lugares que se centre en el cálculo de índices compuestos. Este objetivo persigue el diseño de un sistema escalable que pueda ser usado en diversos contextos manteniendo condiciones de coherencia, escalabilidad, flexibilidad y facilidad de integración. Es, por último, objetivo de esta tesis, la aplicación y validación en escenarios reales de dicho sistema. En particular, se ha integrado y desplegado principalmente en entornos de interacción puerto-ciudad, enmarcados dentro de los proyectos europeos H2020 PIXEL y H2020 ASSIST-IoT, así como en escenarios más regionales/locales a pequeña escala (AVI-2021-ASIGNAT).

### Objetivos específicos

Los objetivos específicos de esta tesis doctoral son los siguientes:

- 1) Analizar las diferentes arquitecturas IoT preexistentes, sus diferentes componentes, opciones de integración y tendencias tecnológicas para los próximos años.
- 2) Conocer el estado del arte respecto a la adopción de índices compuestos en aplicaciones multi-dominio, especialmente centradas en la interacción puerto-ciudad.
- 3) Explorar las diferentes técnicas de integración de modelos de predicción en el “*Edge computing*” para el cálculo distribuido de un índice compuesto, prestando especial atención al aprendizaje federado.
- 4) Estudiar modelos de caracterización de nivel de “precisión tecnológica” de un índice compuesto y descubrir el más apropiado para el uso en el sistema a diseñar.
- 5) Obtener un conocimiento profundo sobre las posibilidades de computación ubicua en un entorno aplicado de Internet de las Cosas.
- 6) Aplicar las más avanzadas técnicas de ingeniería del software para el diseño de sistemas complejos distribuidos.

- 7) Diseñar un sistema software modular que cubra el objetivo general del proyecto acorde a unos requisitos definidos por el doctorando según experiencias en proyectos de investigación.
- 8) Desarrollar una herramienta útil y usable de cálculo automático de índices compuestos
- 9) Desarrollar una metodología de diseño, implementación, configuración y despliegue de dicho sistema para su uso en entornos reales.
- 10) Implementar todos los componentes necesarios:
  - Adquisición de datos,
  - Elemento de capa de interconexión “*edge*” (gateway) y los conectores necesarios,
  - Base de datos de persistencia a largo plazo,
  - Índice compuesto y su cálculo, incluyendo métodos de normalización y agregación,
  - Interfaz de visualización para la interpretación del índice compuesto para evaluar la solución implementada.
- 11) Despliegue y validación de la solución en, al menos, tres casos de uso, principalmente relacionados con la interfaz ciudad puerto.

## 1.4. Principales aportaciones

### 1.4.1. Artículos, Congresos y Jornadas

Artículos de revista:

- *Framework and Methodology for Establishing Port-City Policies Based on Real-Time Composite Indicators and IoT: A Practical Use-Case.* **Lacalle, I.**; Belsa, A.; Vaño, R.; Palau, C. E. *Sensors* 2020, 20, 4131. [DOI: 10.3390/s20154131](https://doi.org/10.3390/s20154131)
- *A novel approach for assessing the port's environmental impacts in real time - the IoT based Port Environmental Index.* Matija Široka; Stjepan Piličić; Teodora Milošević; **Ignacio Lacalle**; Luka Traven. *Ecological Indicators* (Elsevier). [DOI: 10.1016/j.ecolind.2020.106949](https://doi.org/10.1016/j.ecolind.2020.106949)

Actas de congreso:

- *ASSIST-IoT: A Reference Architecture for Next Generation Internet of Things.* Alejandro Fornés-Leal, **Ignacio Lacalle**, Carlos E. Palau, Paweł Szmeja, Maria Ganzha y Marcin Paprzycki (2022). The 21st International Conference on Intelligent Software Methodologies, Tools, and Techniques. [DOI: 10.3233/FAIA220243](https://doi.org/10.3233/FAIA220243)
- *Efficiency of REST and gRPC realizing communication tasks in microservice-based ecosystems.* Marek Bolanowski, Kamil Żak, Andrzej

- Paszkiwicz1, Maria Ganzha, Marcin Paprzycki, **Ignacio Lacalle**, Carlos E. Palau (2022). The 21st International Conference on Intelligent Software Methodologies, Tools, and Techniques. [DOI: 10.3233/FAIA220242](https://doi.org/10.3233/FAIA220242)
- *Reviewing SDN adoption strategies for Next Generation Internet of Things networks*, 2022, César López, **Ignacio Lacalle**, Andreu Belsa, Zbigniew Kopertowski, Carlos E. Palau, Manuel Esteve, [DOI: 10.1007/978-981-16-2877-1\\_57](https://doi.org/10.1007/978-981-16-2877-1_57)
  - *Leveraging IoT and prediction techniques to monitor COVID-19 restrictions in port terminals*, Rafael Vaño, **Ignacio Lacalle**, Benjamín Molina, Carlos E. Palau. November 2021, IEEEExplore, ISSN 978-1-6654-4431-6 (on-line), [DOI: 10.1109/WF-IoT51360.2021.9595919](https://doi.org/10.1109/WF-IoT51360.2021.9595919)
  - *DevSecOps Methodology for Ng-IoT Ecosystem Development Lifecycle – ASSIST-IoT Perspective*, Óscar López, Jordi Blasi, Mikel Uriarte, **Ignacio Lacalle**, et. al., October 2021, <https://vjs.ac.vn/index.php/jcc/article/view/16245>, ISSN 1813-9663 (on-line), Journal of Computer Science and Cybernetics, Vol 37 No 3, [DOI: 10.15625/1813-9663/37/3/16245](https://doi.org/10.15625/1813-9663/37/3/16245)
  - *Functioning prototype of IoT and composite indicators for smart port environmental monitoring*, **I. Lacalle** et al., 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), 2021, pp. 1-2, [DOI: 10.1109/MeditCom49071.2021.9647608](https://doi.org/10.1109/MeditCom49071.2021.9647608).
  - *A Novel Approach for Calculating Real-Time Composite Indicators Relying on Internet of Things and Industrial Data Spaces*. Andreu Belsa, Rafael Vaño, Ignacio Lacalle, Matilde Julián, Fernando Boronat, Carlos E. Palau (2022). IDC 2021. Studies in Computational Intelligence, vol 1026. Springer, Cham. [DOI: 10.1007/978-3-030-96627-0\\_5](https://doi.org/10.1007/978-3-030-96627-0_5)
  - *Advantage Of a Green Port Of The Future*, Joao Pita Costa, **Ignacio Lacalle**, et. al., September 2021, [www.witpress.com](http://www.witpress.com), ISSN 1743-3509 (on-line), WIT Transactions on The Built Environment, Vol 204, Pages 220-234, [DOI: 10.2495/UT210171](https://doi.org/10.2495/UT210171)
  - *Towards Environmental Impact Reduction Leveraging IoT Infrastructures: The PIXEL Approach*. **Lacalle Úbeda, Ignacio**; Llorente, Miguel; Palau, Carlos. (2019). [DOI: 10.1007/978-3-030-34914-1\\_4](https://doi.org/10.1007/978-3-030-34914-1_4)
  - *Small and medium ports' activities modelling: Introduction to the PIXEL approach*. Simon, Erwan; Garnier, Charles; **Lacalle Úbeda, Ignacio**; Costa, Joao; Palau, Carlos. (2019). 149-163. [DOI: 10.2495/MT190141](https://doi.org/10.2495/MT190141)
  - *Tactile Internet in Internet of Things Ecosystems*. **Ignacio Lacalle**, César López, Rafael Vaño, Carlos E. Palau, Manuel Esteve, Maria Ganzha, Marcin Paprzycki, Paweł Szmeja. ICEEE 2022. Lecture Notes in Electrical

Engineering, vol 894. Springer, Singapore. [DOI: 10.1007/978-981-19-1677-9\\_69](https://doi.org/10.1007/978-981-19-1677-9_69)

- *DevSecOps Methodology for Ng-IoT Ecosystem Development Lifecycle – ASSIST-IoT Perspective*, Óscar López, Jordi Blasi, Mikel Uriarte, **Ignacio Lacalle**, et. al., October 2021, <https://vjs.ac.vn/index.php/jcc/article/view/16245>, ISSN 1813-9663 (online), Journal of Computer Science and Cybernetics, Vol 37 No 3, [DOI: 10.15625/1813-9663/37/3/16245](https://doi.org/10.15625/1813-9663/37/3/16245)
- *Enabling Industrial Data Space Architecture for Seaport Scenario*. Sarabia, David; **Lacalle Úbeda, Ignacio**; Palau, Carlos; Esteve, Manuel. (2019). [DOI: 10.1109/WF-IoT.2019.8767216](https://doi.org/10.1109/WF-IoT.2019.8767216)
- *Efficient Deployment of Predictive Analytics in Edge Gateways: Fall Detection Scenario*. Sarabia, David; **Lacalle Úbeda, Ignacio**; Palau, Carlos; Esteve, Manuel. (2019). [DOI: 10.1109/WF-IoT.2019.8767231](https://doi.org/10.1109/WF-IoT.2019.8767231)

Además, los siguientes artículos de presentación en congreso han sido aceptados y están pendientes de publicación (esperada para los próximos meses):

- *Evolution of MANO towards the Cloud-Native Paradigm for the Edge Computing*. Alejandro Fornés-Leal, **Ignacio Lacalle**, Rafael Vaño, Carlos E. Palau, Fernando Boronat, Maria Ganzha y Marcin Paprzyck. (2022). Presentado en el congreso: 2022 International conference on Advanced Computing and Intelligent Technologies (ICACIT 2022).
- *Design of a next-generation interoperable Cognitive Port Solution*. Andreu Belsa, Matilde Julián, Achilleas Marinakis, Anastasios Nikolakopoulos, Vrettos Moulos, Héctor Iturria, José Antonio Clemente, Xhulja Shahini, Tristan Kley, Andreas Metzger, Miguel Bravo, Paolo Calciati, Christos A. Gizelis, Filippos Nikolopoulos-Gkamatsis, Konstantinos Nestorakis, **Ignacio Lacalle**, Carlos E. Palau y Santiago Cáceres. (2022). Presentado en el congreso IoTWeek-2022 (Dublín, Irlanda).
- *Maritime terminals' cargo handling equipment cooperation leveraging IoT and edge computing: The ASSIST-IoT approach*. Eduardo Garro, **Ignacio Lacalle**, Francisco Blanquer, Adrian Ramos, Angel Martinez, Piotr Sowinski, Miguel Angel Llorente, Carlos E. Palau (2022). Aceptado para presentación en el congreso: Transport Research Arena (TRA) 2022.

## 1.4.2. Proyectos de investigación

### PIXEL (H2020 - G.A. No 769355):

PIXEL (*Port IoT for Environmental Leverage*) fue un proyecto financiado por la Comisión Europea (programa Horizonte 2020 - H2020) que comenzó en mayo de 2018 y finalizó en septiembre de 2021. El objetivo del proyecto fue crear la primera solución modular que combinara metodología y tecnología avanzada (*smart*) con especial orientación a puertos marítimos de tamaño mediano y pequeño que permitiera optimizar las operaciones gracias al Internet de las Cosas y reducir el impacto medioambiental de los mismos. El proyecto planteó una arquitectura de IoT completa y una serie de servicios validados en cuatro puertos europeos (Burdeos, Monfalcone, Tesalónica y El Pireo). El objetivo de estos servicios es cubrir las lagunas y los retos actuales del sector marítimo, como la predicción de la demanda de energía, la previsión de la congestión del tráfico, la armonización de la logística intermodal y, especialmente, la reducción del impacto medioambiental.



### Relevancia e impacto del proyecto PIXEL en la tesis doctoral:

El candidato participó de forma activa en este proyecto de investigación durante toda su duración, incluyendo la fase de concepción o propuesta. El rol ejercido por el candidato fue transversal, participando principalmente en el diseño y desarrollo de un índice compuesto para el reflejo del impacto medioambiental de los puertos. No obstante, colaboró en diferentes partes, desarrollando software y siendo partícipe de las acciones de difusión y explotación de los resultados obtenidos.

En este contexto, puede decirse que el proyecto PIXEL ha tenido una gran relevancia dentro de la presente tesis. El desarrollo principal (dicho índice compuesto) supuso una de las principales motivaciones para iniciar la tesis doctoral, al descubrirse que no había ninguna herramienta automatizada que hiciera uso de tecnologías IoT para dar solución a desafíos de esa naturaleza. En particular, el impacto de este proyecto en la solución de la tesis son los siguientes:

- La arquitectura IoT creada en PIXEL (en cuyo diseño colaboró el candidato) sirvió como referencia e inspiración para la arquitectura propuesta en la tesis (sección 3).
- El índice de impacto medioambiental de puertos marítimos (PEI – del inglés *Port Environmental Index*) sirvió como punto de partida para el planteamiento del servicio de valor añadido propuesto en esta tesis. Adicionalmente, este índice compuso inspiró la metodología e implementación de los desarrollos avanzados expuestos en la sección 4.
- Tres de los casos de validación de la solución de esta tesis se llevaron a cabo dentro del contexto del proyecto PIXEL (subsecciones 5.3, 5.4 y 5.6), inspirando también el planteamiento de otro de ellos (5.5).

**ASSIST-IoT (H2020 - G.A. No 957258):**

ASSIST-IoT (*Architecture for Scalable, Self-\*, human-centric, Intelligent, Secure, and Tactile next generation IoT*) es un proyecto financiado por la Comisión Europea (programa Horizonte 2020 – H2020)



que comenzó en noviembre de 2020, con fecha de finalización prevista para octubre de 2023. ASSIST-IoT tiene como objetivo proporcionar una arquitectura de referencia innovadora para el Internet de las Cosas de Nueva Generación (NGIoT). El proyecto propone un diseño descentralizado, considerando diferentes elementos de computación en el continuo *edge-cloud* (de más cerca a más lejos del origen del dato) por medio de funciones virtualizadas, con una clara separación de los planos de control y de datos, facilitando una eficiente programabilidad de la infraestructura. Incluye la incorporación de nuevas tecnologías como interoperabilidad semántica, seguridad de transacciones gracias a una *Distributed Ledger*, mecanismos (semi)autónomos, Tactile Internet para aplicaciones que requieran baja latencia, realidad artificial/mixta/virtual (AR/MR/VR), entre otras. El proyecto cuenta con pilotos sobre: (i) automatización de puertos marítimos; (ii) seguridad de los trabajadores en entornos de construcción, y (iii) supervisión y diagnóstico de vehículos.

**Relevancia e impacto del proyecto ASSIST-IoT en la tesis doctoral:**

El candidato se encuentra actualmente participando activamente en este proyecto. El rol ejercido se centra en la supervisión global de la ejecución del proyecto, así como en el diseño de la arquitectura de alto nivel. Se trata de un proyecto con una gran cantidad de tecnologías investigadas, donde el candidato ha podido participar de forma extensiva en lo relacionado con *edge computing*, la orquestación de servicios y el Tactile Internet, y en menor medida en la exploración de arquitecturas de aprendizaje federado.

En particular, el impacto de este proyecto en la solución de la tesis son los siguientes:

- Varios principios de diseño de la arquitectura de la tesis han sido obtenidos gracias a la participación en este proyecto (sección 3), tales como escalabilidad, virtualización y “containerización”.
- La propuesta dual *cloud-edge* de la solución se ha inspirado en el diseño descentralizado y orientado a computación en borde (*edge computing*) de ASSIST-IoT.
- La inclusión de nuevas tecnologías como el aprendizaje federado y el Tactile Internet (aunque finalmente no embebidas de forma directa) se han planteado gracias a la participación en ASSIST-IoT.

**ASIGNAT (AVI-2021-Expediente INNEST-2021/175):**

ASIGNAT (*Arquitectura de Gemelos Digitales y Servicios de Aprendizaje Automático para la Planificación en el Transporte Terrestre de Mercancías*) es un proyecto financiado por la Agencia Valenciana de Innovación (AVI) que comenzó en septiembre de 2021 y plantea ejecutarse hasta septiembre de 2023. El objetivo del proyecto es proporcionar una solución para la mejora de la asignación del transporte en carretera de contenedores desde y hacia el puerto de Valencia. se divide en tres pilares: (1) cálculo y estimación del tiempo esperado de llegada (ETA del inglés *Estimated Time of Arrival*) de camiones a puntos de interés, (2) aprendizaje automático para la asignación óptima de rutas multimodales y (3) implementación de un gemelo digital para el control de la flota de vehículos de transporte y la modelización de la asignación del tándem conductor-camión para cada viaje.



**Relevancia e impacto del proyecto ASIGNAT en la tesis doctoral:**

El candidato se encuentra participando en este proyecto de forma tangencial. La colaboración se contextualiza dentro del tercer objetivo previamente mencionado, en el diseño y desarrollo de un gemelo digital de la flota de vehículos para modelar e interactuar en el proceso de asignación.

En particular, el impacto de este proyecto en la solución de la tesis es como sigue:

- El gemelo digital desarrollado (diseño propuesto por el candidato) ha sido basado en la arquitectura diseñada para la solución de la tesis.
- Se ha planteado la toma de decisiones de asignación en función de un índice compuesto, cuyo diseño e implementación ha seguido la metodología planteada en esta tesis doctoral.
- Uno de los casos de validación de la solución de esta tesis se llevó a cabo dentro del contexto del proyecto ASIGNAT (subsección 5.7).

### **1.4.3. Desarrollo software**

A lo largo de la memoria se hace referencia a una serie de componentes software elaborados por el doctorando, haciendo hincapié en el grado de participación en los mismos. El candidato apuesta firmemente por políticas de código abierto y compartición de software libre, no obstante, la participación en varios proyectos de investigación hace que esto no siempre sea posible. Como referencia, estos repositorios contienen tanto algunos de los artefactos desarrollados como portales de documentación:

- <https://github.com/pixel-ports/pixel-platform>
- <https://github.com/pixel-ports/pixel-data-models>



- [https://github.com/pixel-ports/thpa\\_traffic\\_prediction](https://github.com/pixel-ports/thpa_traffic_prediction)
- <https://gitpixel.satrdlab.upv.es/>
- <https://assist-iot-enablers-documentation.readthedocs.io/en/latest/index.html>
- <https://pixel-ports.readthedocs.io/en/latest/>

## 1.5. Organización de la memoria

En este punto se realiza una breve justificación de la estructura del documento. Acorde con los objetivos de la tesis doctoral, el trabajo reflejado se ha abordado desde varias perspectivas, estableciendo un flujo de análisis, diseño, implementación y validación de la solución. Al tratarse de un trabajo autocontenido, pero que se ha nutrido de los avances realizados dentro de varios proyectos de investigación, se ha decidido mencionarlos a lo largo del texto sin la necesidad de establecer capítulos específicos para los mismos. Adicionalmente, conviene mencionar que el orden expuesto a continuación no se corresponde necesariamente con la ejecución temporal llevada a cabo. Se ha establecido el siguiente índice para facilitar la lectura y ofrecer una visión completa de la creación y validación de la arquitectura.

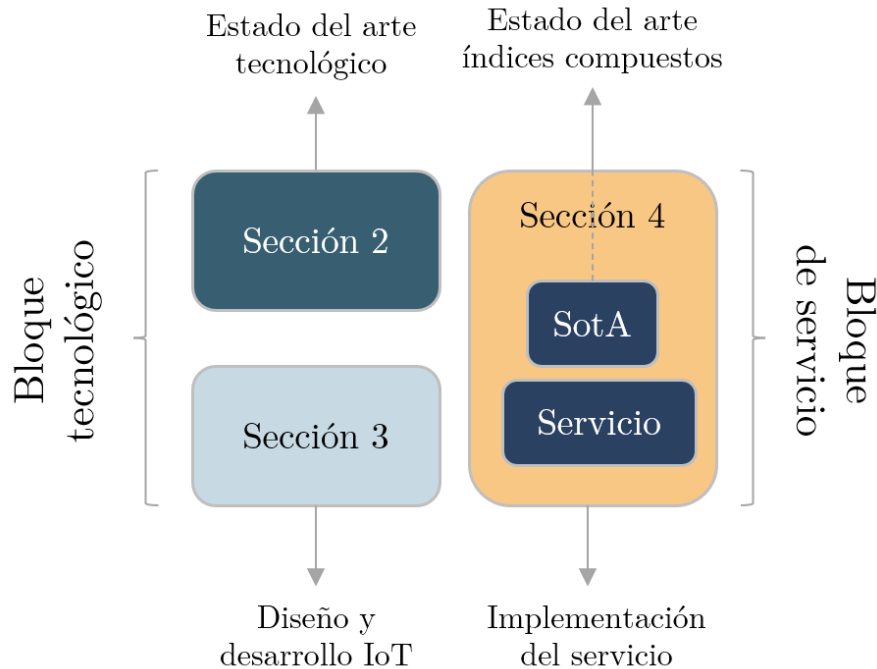
En la sección 1 se expone el contexto de la tesis doctoral, sus objetivos y las principales aportaciones en forma de artículos, presentaciones en congresos y software en abierto.

Es conveniente recordar en este punto que esta tesis doctoral consta de dos elementos principales que han supuesto el grueso de la investigación. Por una parte, la generación de una arquitectura de Internet de las Cosas (**bloque puramente tecnológico**), que comprende la definición, diseño y testeo de componentes software que forman un sistema completo; base sobre el cual pueden desarrollarse servicios de valor añadido sobre datos heterogéneos. Por otra parte, uno de esos “posibles servicios” se ha desarrollado de manera específica. En particular, una herramienta de cálculo de indicadores compuestos. Esta segunda contribución (**bloque de servicio**), que hace uso de la primera para ejecutarse y proveer valor añadido, va más allá de un diseño y desarrollo software, ya que contiene metodología, procesos matemáticos y asunciones orientadas a lógica de negocio.

Por tanto, se ha decidido seguir una estructura de secciones fiel al planteamiento anterior, dividida en dos grandes bloques:

La sección 2 expone los trabajos de búsqueda en la literatura y análisis del estado del arte para la investigación sobre el **bloque tecnológico**. Este bloque se completa con la sección 3, donde se evidencian las acciones de diseño y desarrollo de la arquitectura base de la solución, incluyendo el contexto en que se enmarca, así como un análisis de los requisitos capturados. Se estructura en consiguientes subsecciones acorde a la composición modular de la arquitectura.

Por su parte, el **bloque de servicio** se ha concentrado en la Sección 4, detallando por una parte el análisis teórico de la construcción de índices compuestos y, por otra, la implementación de dicho servicio sobre la arquitectura propuesta.



**Figura 1.2:** Estructuración en bloques de los capítulos centrales

Inmediatamente después, las actividades de validación y puesta en práctica de ambos bloques integrados se reflejan en la sección 5, que se compone de los cinco casos de uso que se han llevado a cabo en el contexto de la tesis doctoral.

Durante toda la ejecución de la tesis doctoral, el candidato ha tratado de concebir la solución como un elemento potencialmente explotable en términos de transferencia tecnológica. Por ello, se ha incorporado en la sección 6 un esbozo preliminar de un posible modelo de negocio basado en la tecnología propuesta.

La sección 7 reflexiona sobre las conclusiones obtenidas tras la compleción de los objetivos, así como propone una serie de líneas futuras de investigación.

Por último, se describen las referencias utilizadas.

Como elementos adicionales, se ha incluido una serie de documentación complementaria que ayuda a refrendar las evidencias aportadas pero que dificultaría la lectura dentro del cuerpo del documento.

# Capítulo 2

## Estado del arte tecnológico

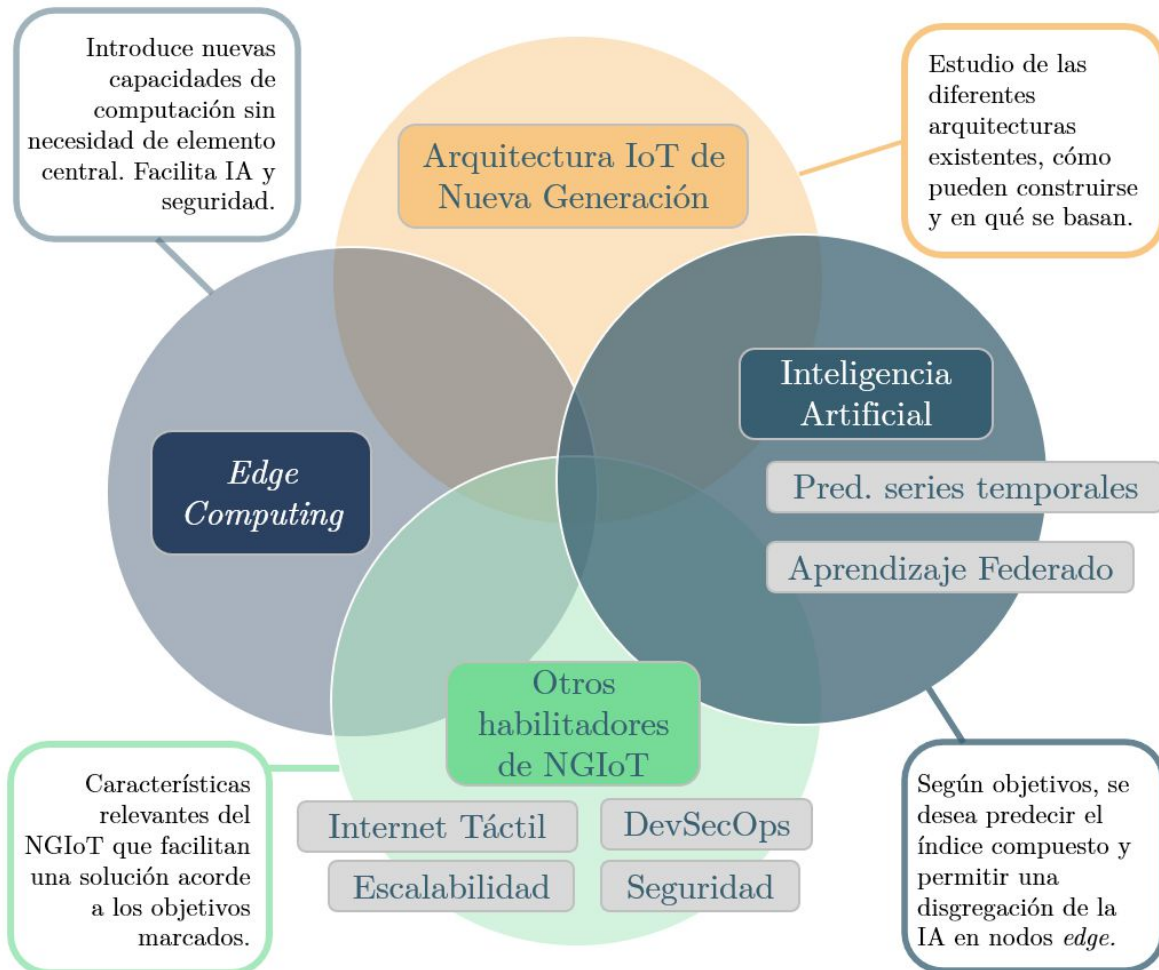
### 2.1. Introducción

Este capítulo tiene como objetivo describir el estado del arte actual de las diferentes técnicas necesarias para construir la solución tecnológica (arquitectura IoT) de la tesis doctoral. Estos análisis fueron acometidos al comienzo de la tesis con la intención de guiar las acciones durante el resto del programa.

Para ello se partió del concepto que subyace al objetivo técnico principal: desarrollar una arquitectura de IoT de Nueva Generación con un propósito concreto. Tal y como fue definido por el proyecto NG-IoT [7], una arquitectura *Next Generation IoT* (de ahora en adelante, NGIoT) es aquella que evoluciona las características tradicionales de los despliegues IoT proporcionando nuevas capacidades a través de “habilitadores” (*enablers*) digitales. La diferencia principal con el enfoque clásico es que estos habilitadores son ahora incluidos como parte

integral de las mismas, sin necesidad de sustituciones o reemplazos. El listado completo de estos *enablers* es: *Edge Computing*, 5G, Inteligencia Artificial (IA), Seguridad y privacidad enriquecida (incluyendo DLT), Realidad Aumentada, Tactile Internet y Gemelos Digitales (*Digital Twins*).

Sin embargo, para que una arquitectura sea considerada NGIoT no es necesario que incorpore de forma nativa todas las tecnologías anteriores, basta con que integre alguno de dichos conceptos modernos en función de su objetivo de aplicación. Por tanto, decidió estructurarse el estado del arte de esta tesis partiendo de dicho principio, seleccionando las tecnologías necesarias para cubrir los objetivos planteados. Así, la estructura respondería al siguiente diagrama, que incluye una breve explicación de su influencia en la tesis doctoral:



**Figura 2.1:** Estructuración del estado del arte tecnológico en la tesis

Conviene destacar que la profundidad de análisis de cada técnica está directamente relacionada con la relevancia de la misma para la solución objetivo del trabajo de la tesis. Por ejemplo, las subsecciones centradas en arquitecturas NGIoT y *edge computing* han sido cruciales para el desarrollo de la tesis doctoral, mientras que las tecnologías de aprendizaje federado o Tactile Internet, debido también a su grado de madurez actual, son expuestas de una manera más sucinta.

## 2.2. Arquitecturas NGIoT de referencia

El concepto de arquitectura de referencia en este trabajo se refiere a aquella especificación que exponga una disposición de los componentes y servicios y unas buenas prácticas de un sistema de IoT que vaya a implementarse de forma recurrente con objetivos similares pero con diferentes contextos o limitaciones. El objetivo de esta subsección es repasar sucintamente las arquitecturas de referencia más comunes en IoT, así como la diferencia entre éstas y el término de “plataformas IoT”. Además, este capítulo explora cómo el término habitual de IoT está evolucionando hacia Next Generation IoT (NGIoT) y, por lo tanto, cómo las arquitecturas de referencia más modernas incluyen esos conceptos dentro de sus disposiciones, distinguiendo los patrones existentes para su diseño.

Dado que el objetivo principal de este doctorado es diseñar una arquitectura IoT (no una plataforma) orientada a la aplicación, resultó apropiado explorar tales arquitecturas de referencia con el fin de mantener una consistencia tecnológica y funcional común cuando los despliegues y los requisitos varíen (véase la sección 5). Además, alinear la arquitectura de la solución con las RA (del inglés *Reference Architecture*) consolidadas (o, al menos, seguir sus principios fundamentales) puede sentar las bases para un futuro crecimiento y explotación del sistema (véase la sección 6), al tiempo que proporciona estabilidad y fiabilidad a la solución de cara a múltiples escenarios y a lo largo del tiempo.

### Plataformas

Según IoT Analytics, existen más de 613 plataformas IoT disponibles en el año 2022 (los datos para 2023 aún no están listos) [8]. Partiendo de unas 260 formales en 2015, está claro que ha habido una proliferación de las mismas, en su mayoría impulsadas por *start-ups* o pequeñas empresas tecnológicas (sin embargo, como nota relevante, el 10% del total de proveedores son empresas con ingresos multimillonarios pero que copan una gran parte del mercado).

Una plataforma de IoT es un concepto bastante amplio, definido a veces (por ejemplo, por Gartner [9]) como “*un conjunto de software local o un servicio en la nube que supervisa y puede gestionar y controlar varios tipos de puntos finales, a menudo a través de aplicaciones que las unidades de negocio despliegan en la plataforma*”. También puede entenderse, según la definición de la acción financiada por la UE IoT-EPI [10] como “*una capa inteligente que conecta las cosas a la red y que abstrae las aplicaciones de las cosas con el objetivo de permitir el desarrollo de servicios. Una plataforma de IoT facilita la comunicación, el flujo de datos, la gestión de dispositivos y la funcionalidad de las aplicaciones. Una plataforma de IoT permite que las aplicaciones conecten máquinas, dispositivos, aplicaciones y personas a centros de control y datos*”. En lo concerniente a este doctorado, la definición que se ha considerado es la siguiente: una plataforma IoT es un conjunto de determinadas características de software interconectadas que permiten ciertas capacidades dentro de un ecosistema de Internet de las Cosas, como la gestión de

dispositivos, la conectividad, el análisis de datos, la simulación y las herramientas de visualización. Además, las plataformas IoT proporcionan interfaces para interactuar con los datos y objetos que forman parte de dicho ecosistema y también permiten gestionar la conectividad y la red subyacente, convirtiéndose en un entorno cerrado que se requiere para que los sensores (normalmente específicos para las mismas) funcionen y sean accesibles.

Dentro del panorama de plataformas IoT, éstas pueden abordar diferentes objetivos y ser de naturaleza variada, potencialmente clasificables como (i) de código abierto o (ii) comerciales, o, desde el punto de vista del despliegue, como (i) centradas en servicios *en la nube*, (ii) centradas en la Industria 4.0, (iii) centradas en la gestión de dispositivos, (iv) centradas en la comunicación y la seguridad o (v) de propósito general. En algunos trabajos, como [11] o [12] se puede consultar un análisis completo de las características de las plataformas IoT, así como una revisión de las opciones e implementaciones actuales (fuera del alcance de este doctorado).

Algunos proyectos de investigación centrados en conseguir interoperabilidad entre plataformas del Internet de las Cosas ahondaron en las principales referencias en que fijarse. El proyecto INTER-IoT [13] basó su clasificación desde un modelo de capas, agnóstico al dominio de aplicación. El proyecto ACTIVAGE [14], en cambio, realizó un análisis más específico para aplicaciones de salud (alejado del objetivo de esta tesis), pero aportando una interesante perspectiva: por niveles de dispositivo, conectividad, nube y aplicación.

La mencionada proliferación de las plataformas de IoT en el periodo 2014-2021 puede emanar de que éstas plantean una menor distancia a la fase de implementación en comparación con las arquitecturas globales de IoT, que suelen incluir software *on-purpose* preseleccionado y predesplegado, o de que poseen una usabilidad más directa. Sin embargo, durante los últimos años, el crecimiento en la publicación de estas plataformas se ha detenido [15].

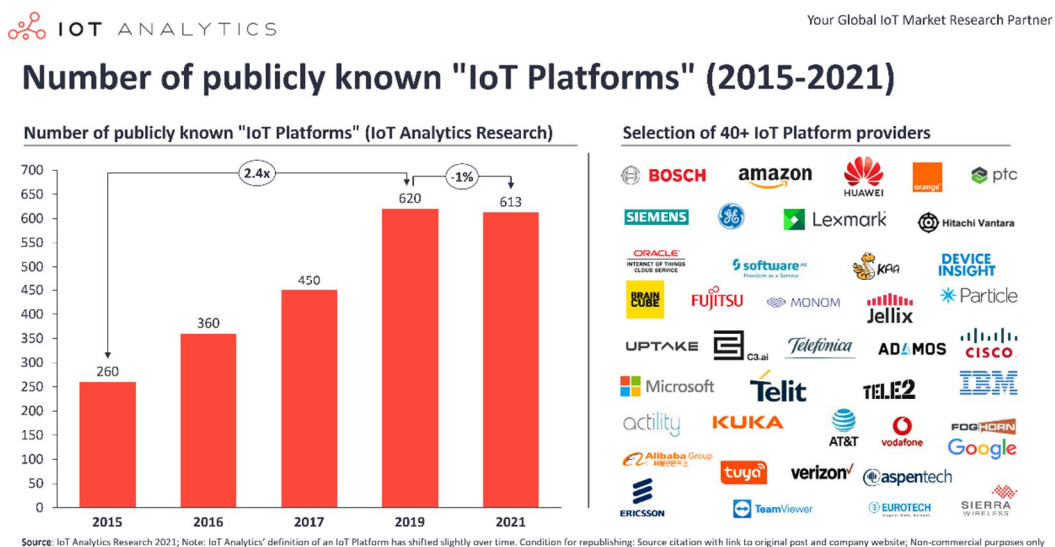


Figura 2.2: Evolución de plataformas IoT 2015-2021

Desde 2019, los proveedores de tecnología IoT parecen estar pivotando sus productos hacia soluciones más amplias de IoT o específicas de seguridad en lugar de plataformas. Una posible explicación es que el auge de las plataformas de IoT se debió al principio de “*comprar algo hecho en lugar de hacerlo uno mismo*”. En los inicios del IoT, no existían muchas tecnologías de código abierto relacionadas ni directrices disponibles, y los sensores y protocolos eran más bien cerrados y específicos, por lo que el uso (mediante la compra) de plataformas era el enfoque más natural, garantizando un funcionamiento adecuado a expensas de la flexibilidad o la personalización. Hoy en día, protocolos relevantes (como MQTT, LoRa, NB-IoT), tecnologías (como el middleware, los brokers, las aplicaciones) y principios generales de diseño (modularización, estandarización de interfaces, microservicios) abiertos y fácilmente accesibles están haciendo que las compañías aprovechen la oportunidad de adaptar una pila de IoT a sus necesidades sin tener que depender de una plataforma de IoT específica (y no adecuada) proporcionada por un proveedor particular [8].

### Arquitecturas

Tal y como se ha analizado, tradicionalmente las plataformas de IoT se han centrado en la monitorización y la actuación, estando en su mayoría vinculadas a un proveedor y, muy a menudo, a un caso de uso específico. Aplicadas al ecosistema IoT, las arquitecturas de referencia (de ahora en adelante RA, del inglés *Reference Architecture*) tienen como objetivo definir un conjunto más amplio de instrucciones, recomendaciones y directrices para construir un despliegue IoT específico (que puede ser una plataforma). Su objetivo consiste en evaluar una serie de requisitos para proporcionar un conjunto de funcionalidades, estructuras de información y mecanismos [16] que sirvan de modelo para desarrollar e implementar sistemas IoT. Las arquitecturas de referencia suelen tener un alto nivel de abstracción, por lo que pueden adaptarse a diferentes dominios o aplicaciones. Una arquitectura IoT también se define como la herramienta que espera el arquitecto de sistemas para crear las bases de un despliegue IoT fiable, seguro y resistente. Esto es, de forma modular a través de bloques y opciones de diseño flexibles, capaz de cubrir y describir sistemas específicos que abarcan los requisitos funcionales, el rendimiento y el despliegue, la estandarización de las interfaces, la seguridad y la conectividad con su entorno. A diferencia de las plataformas, una arquitectura se crea para que sirva de referencia para construir un sistema específico; no impone una tecnología concreta, sino que sugiere diferentes opciones; no viene preconstruida ni cerrada, sino que pretende ser abierta y fácilmente interpretable; no se centra en la gestión de dispositivos o redes, sino en la incorporación de principios y metodologías de diseño. Da más libertad a los arquitectos de sistemas en lugar de ofrecer un producto cerrado y listo para usar a los usuarios de alto nivel.

Como breve resumen, cualquier arquitectura IoT de referencia debe incluir, al menos, los siguientes aspectos (definidos por el proyecto IoT-A [17]):

- Define una forma uniforme de interactuar (monitorización y actuación) con los dispositivos IoT y otras fuentes de datos.
- Aborda el problema de cómo se trata la información y cómo se distribuye en el sistema
- Proporciona indicaciones claras sobre la escalabilidad, explicando cómo un posible despliegue de la arquitectura podría crecer horizontal y verticalmente.
- Incluye disposiciones de seguridad, indicando cómo proteger los dispositivos y los usuarios en el ecosistema.
- Especifica las formas de adaptación a los diferentes escenarios.
- Incluye opciones de programabilidad e interconexión con el mundo exterior a través de APIs u otros estándares específicos.
- Proporciona tecnologías recomendadas (preferiblemente, de código abierto), pero no impone cuáles deben utilizarse.
- Especifica los aspectos de integración, tanto de los módulos internos como de los artefactos externos.

Debido a la heterogeneidad de conceptos y tecnologías, una arquitectura de referencia de IoT es más compleja que una arquitectura ICT tradicional debido a las relaciones entre las diferentes tecnologías utilizadas. Siguiendo los modelos de concepción anteriores, y suponiendo que se usan, al menos, cinco módulos diferentes, el número de combinaciones es mayor de 1,4 millones [18], pudiendo variar los elementos funcionales, la interacción entre ellos, la gestión de la información, las características operacionales y de despliegue y las tecnologías recomendadas. Por ello, para la solución de esta tesis será necesario proponer una arquitectura con unas bases sólidas, que se apoye en referencias consolidadas en el estado del arte. Así, el sistema mantendrá la consistencia en futuros despliegues, aunque varíen los requerimientos funcionales en diversos casos.

Se han propuesto numerosas arquitecturas, muchas de ellas definidas en contextos específicos y que aportan soluciones a una parte del “mundo de las cosas” [19]. Durante los últimos años, algunas iniciativas han aparecido como resultado de una serie de acciones de investigación (FIWARE, OM2M/OpenCDM, IoT-EPI, AIOTI) y comerciales (AWS IoT, CISCO, IBM Watson), para idear arquitecturas IoT avanzadas, modernas y agnósticas al dominio (y, a veces, incluso de código abierto) más que plataformas. En las próximas páginas, se repasan las arquitecturas de referencia más importantes en las primeras etapas del IoT. Posteriormente, se expone la llegada de NGIoT y las consiguientes propuestas de arquitectura de referencia.

### FP7 - IoT-A

IoT-A [20] fue un proyecto financiado por la CE (programa FP7) que estableció un modelo de referencia de arquitectura para el ámbito del Internet de las Cosas [21]. El proyecto se desarrolló desde 2010 hasta 2013 y puede considerarse



la base de todos los esfuerzos de la UE realizados en este ámbito desde entonces. Este modelo de referencia propuso una arquitectura IoT que cubriera completamente las cuestiones de seguridad y privacidad, así como la escalabilidad y la interoperabilidad, entre otros aspectos, estructurando sus disposiciones en puntos de vista (“*diferentes ángulos para ver una arquitectura que se pueden utilizar cuando se diseña y se implementa*”) y perspectivas (“*conjunto de tareas, tácticas, directivas y decisiones arquitectónicas para garantizar que un sistema concreto cumple con uno o más atributos de calidad*”).

Estas definiciones resultaron muy útiles a la hora de plantear el diseño modular de la arquitectura de esta tesis, como puede verse en la sección 3.2.2. IoT-A estructura los componentes modulares en grupos funcionales, a saber: Gestión de procesos IoT, organización de servicios, entidad virtual, comunicación IoT, seguridad y gestión.

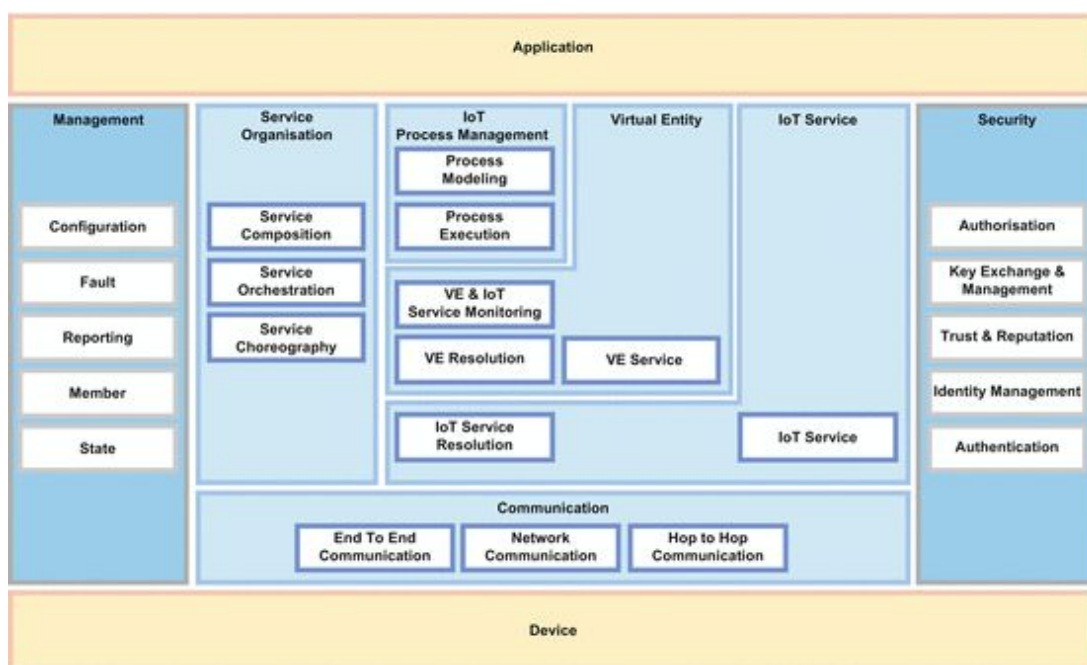
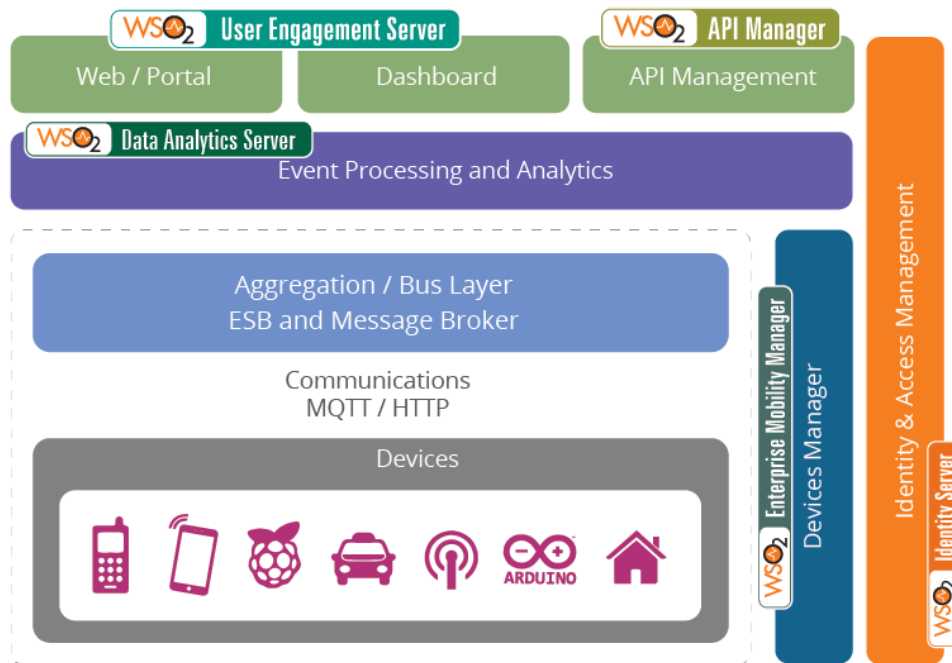


Figura 2.3: Arquitectura FP7-IoT-A. Fuente: [20]

### WSO2

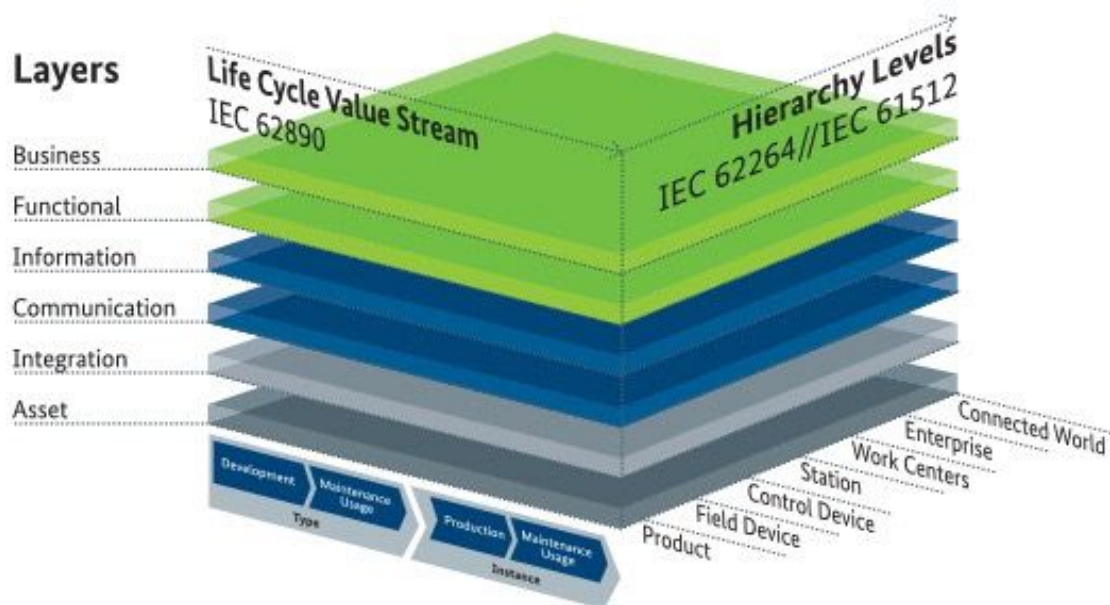
La compañía WSO2, con una larga tradición en la creación de productos IT e IoT, elaboró una arquitectura de referencia compuesta por varias capas basadas en sus soluciones, centrada especialmente en la seguridad y la interoperabilidad, con su *API Manager* como principal elemento concentrador sobre el que debería orbitar la arquitectura [22]



**Figura 2.4:** Arquitectura WSO2 IoT. Fuente: [22]

### RAMI4.0

En 2015, la organización alemana I4.0 [23] propuso la primera versión del modelo arquitectónico de referencia de la industria 4.0 (RAMI 4.0) [24] que, a diferencia de la mayoría de las arquitecturas mencionadas en este apartado, se convirtió en una prenormativa internacional (IEC PAS 63088 [25]).



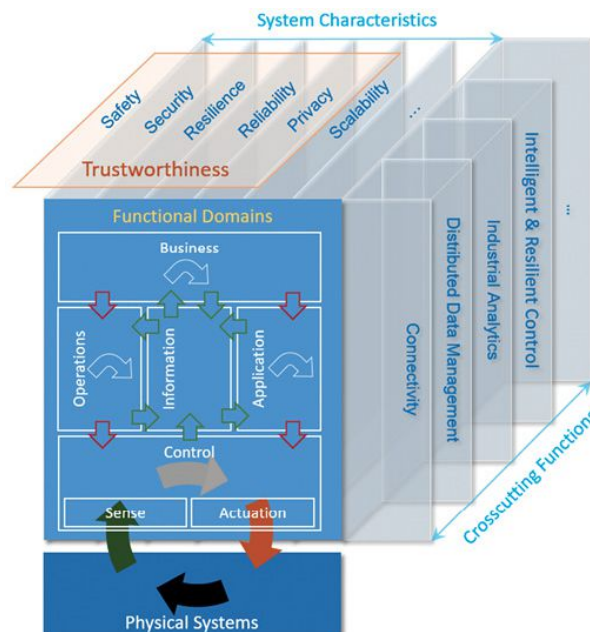
**Figura 2.5:** Arquitectura RAMI4.0. Fuente: [23]

Define una arquitectura orientada a servicios en la que cada uno de los componentes/módulos proporciona servicios a los demás componentes mediante un protocolo de comunicación a través de la red. RAMI es representada como un modelo tridimensional para ajustarse al ecosistema de la industria, ya que apuesta claramente por los principales aspectos de la Industria 4.0: (i) la interoperabilidad entre máquinas, personas y datos, (ii) la adquisición y el almacenamiento en tiempo real, (iii) la orientación a servicio -centrada en el cliente-, y (iv) la modularidad. Uno de los principales objetivos de RAMI 4.0 es asegurarse de que todos los participantes en los debates y actividades de la Industria 4.0 dispongan de un marco común para entenderse.

Es conveniente destacar que esta arquitectura, a pesar de haber sido diseñada en 2015 (relativamente antigua) y estar altamente enfocada en despliegues industriales (principalmente de manufactura) ha resultado una de las referencias más importantes a la hora de diseñar la arquitectura de esta solución. El proyecto PIXEL, donde el candidato participó intensamente, adoptó gran parte de los principios de composición y despliegue expuestos por RAMI 4.0.

### IIRA

También en 2015 (actualización más reciente: 2019), la organización *Industrial Internet Consortium* (IIC) propuso una arquitectura de referencia también enfocada al IoT Industrial (IIoT) basada en estándares ISO/IEC/IEEE. IIRA [26] tiene como objetivo proporcionar una guía y ayuda en el desarrollo, documentación, comunicación y despliegue de sistemas IIoT, teniendo como principal valor su rápida aplicabilidad.



**Figura 2.6:** Arquitectura IIRA. Fuente: [26]

En cuanto a la diferencia entre las dos anteriores, esta arquitectura de referencia se centra en la comunalidad e interoperabilidad entre industrias (es decir,

hacer que las cosas funcionen), mientras que RAMI4.0 se dirige principalmente a la fabricación (es decir, hacer cosas).

Alrededor de dicho periodo (previo al advenimiento de NGIoT en 2019), otras arquitecturas de referencia fueron publicadas por diferentes entidades. Éstas tuvieron mucho menor grado de impacto (al menos en lo referente a proyectos publicados y en relación a la industria europea). Pueden destacarse, entre ellas: la especificación del *Korean Study Group* establecida en 2016 desde el punto de vista de la comunicación y funcional [27] la especificación de la *China Communications Standard Association* (CCSA) propuesta en 2014 (que consta de tres capas: (i) capa de detección, (ii) capa de red y negocio y (iii) capa de aplicación [28]) y, desde otra perspectiva, el Modelo de Siete Niveles del *IoT World Forum* (que, en un ejercicio abstracto de división de las características de los despliegues de la IoT en siete capas (similar a OSI) proporcionó varias recomendaciones [29]).

Llegados a este punto, se habían realizado varios intentos de establecer una referencia única de arquitectura IoT, lo que se tradujo también en la existencia de iniciativas más formales hacia el objetivo de la estandarización. Aparte de la investigación y las entidades privadas, algunos organismos de normalización trabajaron en este ámbito. La ITU propuso un modelo de referencia completo en el entorno IoT: la recomendación ITU-T Y.2060 [30] que aclara el concepto y el alcance del IoT, identifica las características fundamentales y los requisitos de alto nivel y describe el modelo de referencia. Además, más adelante, en consonancia con lo que se explica en las próximas páginas (sobre NGIoT), la norma ISO/IEC/IEEE 42010 [31] que se utiliza en muchas RA modernas [32] especifica un modelo conceptual para ayudar en la descripción de arquitecturas.

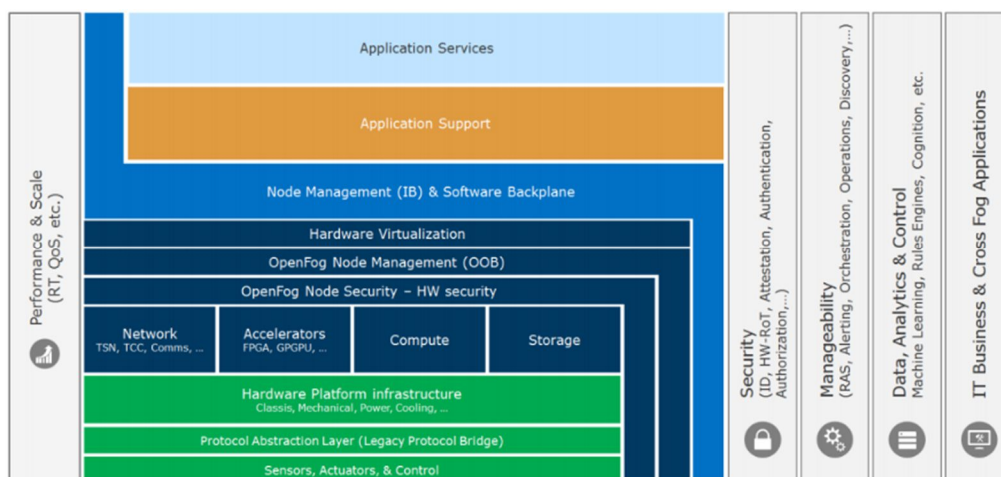
Sin embargo, la reciente generación de nuevos requisitos originados en diferentes áreas (por ejemplo, Big Data Analytics, AI, Cloud/Edge Computing, conectividad 5G, *Industrial Data Spaces* o nuevas interfaces táctiles como la Realidad Aumentada o Virtual), están sentando las bases para transformar el panorama IoT en un cúmulo de facilitadores de la transformación digital integral, apoyando la visión de la Internet de Nueva Generación (NGI, del inglés *Next Generation Internet*) [33].

Además, hoy en día, los despliegues de IoT son cada vez más complejos: se incluyen continuamente nuevas funcionalidades, se amplían las cargas de trabajo y aumenta el número de componentes. Por otra parte, la complejidad también se origina en el hecho de que los nuevos ecosistemas de IoT surgen como resultado de la fusión de los despliegues de IoT existentes, para llevar la próxima generación de servicios a los usuarios. En este contexto, los patrones de arquitectura son un buen punto de partida para abordar las complejidades, apoyando la organización de los componentes del sistema [34]. Sin embargo, algunos de ellos pierden su fuerza dentro de los despliegues reales, creando una brecha que suele dar lugar a más soluciones *vendor-lock*, sin tener en cuenta las preocupaciones existentes analizadas por las RAs o los grupos de estandarización.

Reconociendo lo anterior, las últimas arquitecturas de referencia han evolucionado para incluir características más avanzadas, sobre todo para abordar la computación de borde (*edge computing*) y la virtualización, como OpenFog, ECC RA 2.0, LSP IoT 3D, FAR-EDGE o AIOTI HLA [35].

### OpenFog RA

En 2017, se esbozó la RA de OpenFog [36] fusionando los conceptos de IoT y *edge computing*, estableciendo la primera referencia real para los despliegues de *fog computing*. En la sección 2.3 se profundiza en las disposiciones de OpenFog.



**Figura 2.7:** Arquitectura OpenFog. Fuente: [36]

Para la consideración de esta arquitectura de referencia como elemento relevante para la tesis, resultó interesante su orientación a los microservicios, el papel de la ubicuidad y la adición de módulos como la escalabilidad y la seguridad.

### RAMEC ó ECC-RA

El modelo de arquitectura de referencia Edge Computing (RAMEC) [37] fue presentado por el *European Edge Computing Consortium* (EECC) en 2019 con el objetivo de acelerar la adopción de infraestructuras TIC industriales basadas en software, interoperables, programables, seguras y fáciles de usar.

ECC-RA (o RAMEC) está fuertemente arraigada a los conceptos esbozados por RAMI 4.0 y cuenta con el apoyo de sólidas empresas de Europa y de fuera de ella, centradas en las necesidades de la industria hacia la NGIoT, con el objetivo de proporcionar orientación y situar actividades como iniciativas, normas, publicaciones e implementaciones en un contexto más amplio.

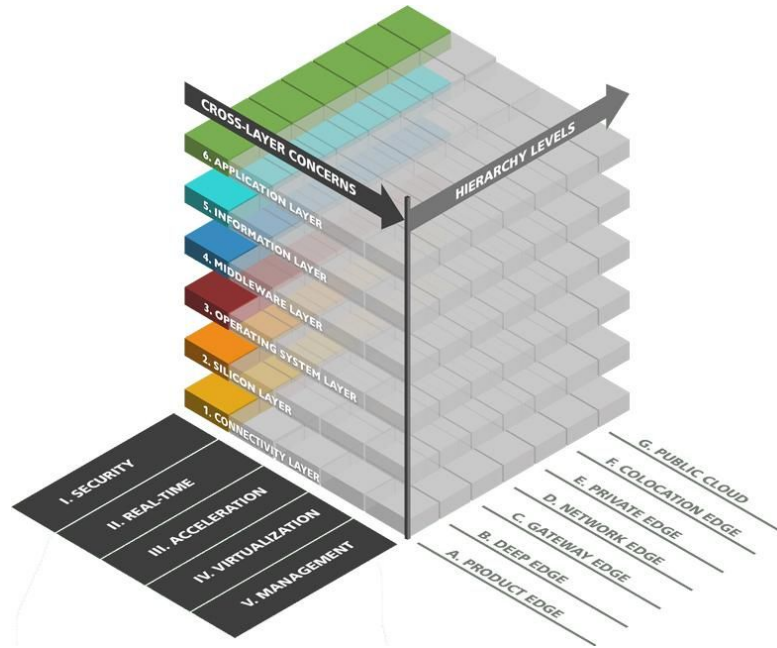


Figura 2.8: Arquitectura RAMEC ECC-RA. Fuente: [37]

*H2020-LSP IoT 3D*

El proyecto CREATE-IoT del programa europeo de pilotos a gran escala de IoT (LSP, *Large Scale Pilots*) presentó en 2018 una arquitectura de referencia de IoT 3D agnóstica. Dado que muchos de los proyectos implicados propusieron RAs que compartían muchos puntos en común, se combinaron para presentar la arquitectura 3D de IoT del LSP [38] que se basa en capas, funciones transversales y propiedades no funcionales, teniendo así una estructura similar a la de la IIRA.

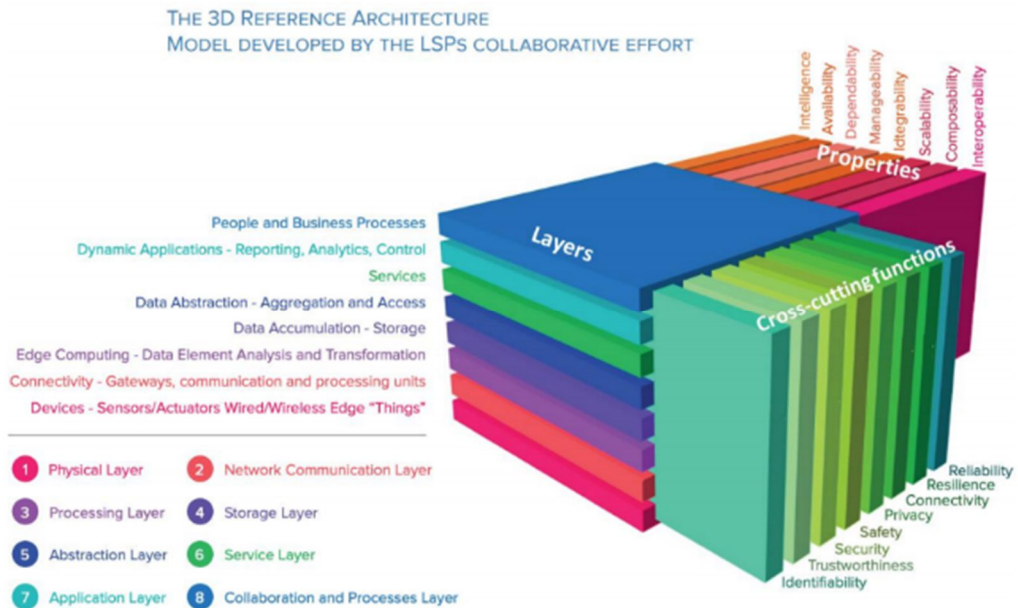


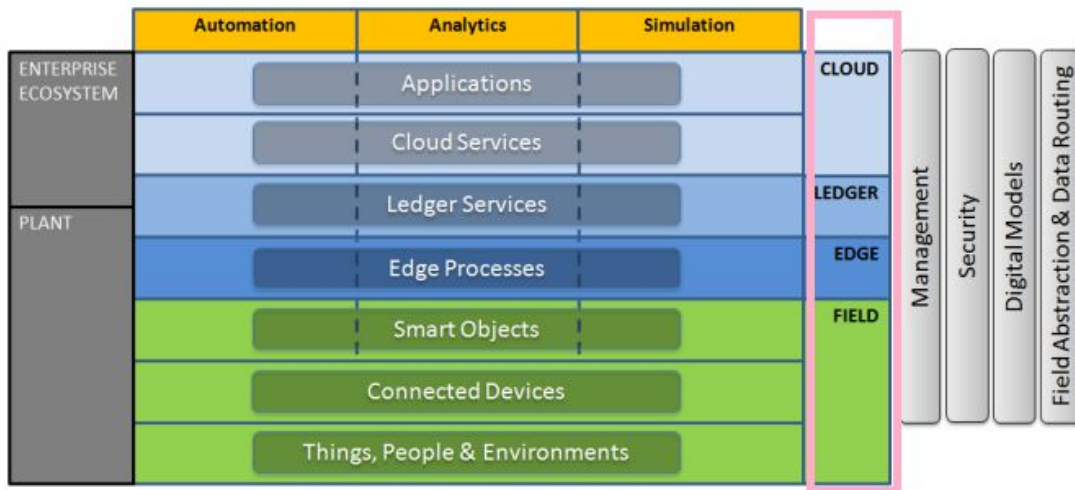
Figura 2.9: Arquitectura LSP IoT 3D. Fuente: [38]



Las ocho capas definidas, junto con las funciones y propiedades transversales, componen la arquitectura tridimensional, con responsabilidades claramente separadas y un alto grado de modularidad. Como se puede observar, las capacidades de Edge Computing se abordan en una capa dedicada, lo que demuestra la importancia que este habilitador ha ido ganando recientemente en el IoT.

***FAR EDGE***

En 2017, el proyecto financiado por la CE FAR EDGE publicó su arquitectura que pretendía convertirse en una referencia para los despliegues de *edge-IoT* en el ámbito de la fabricación inteligente [39]. A pesar de estar dirigido a un sector específico, FAR EDGE fue una referencia muy interesante debido a la amplia documentación y al conjunto bien elaborado de protocolos, estándares y metodologías, teniendo en cuenta la mayoría de las preocupaciones de NGIoT. Sobre la base de RAMI 4.0, IIRA y OpenFog, FAR EDGE establece mayores niveles de automatización, cooperación entre nodos e introduce el no repudio, privacidad y protección de datos con el uso de tecnologías Blockchain.



**Figura 2.10:** Arquitectura FAR EDGE. Fuente: [39]

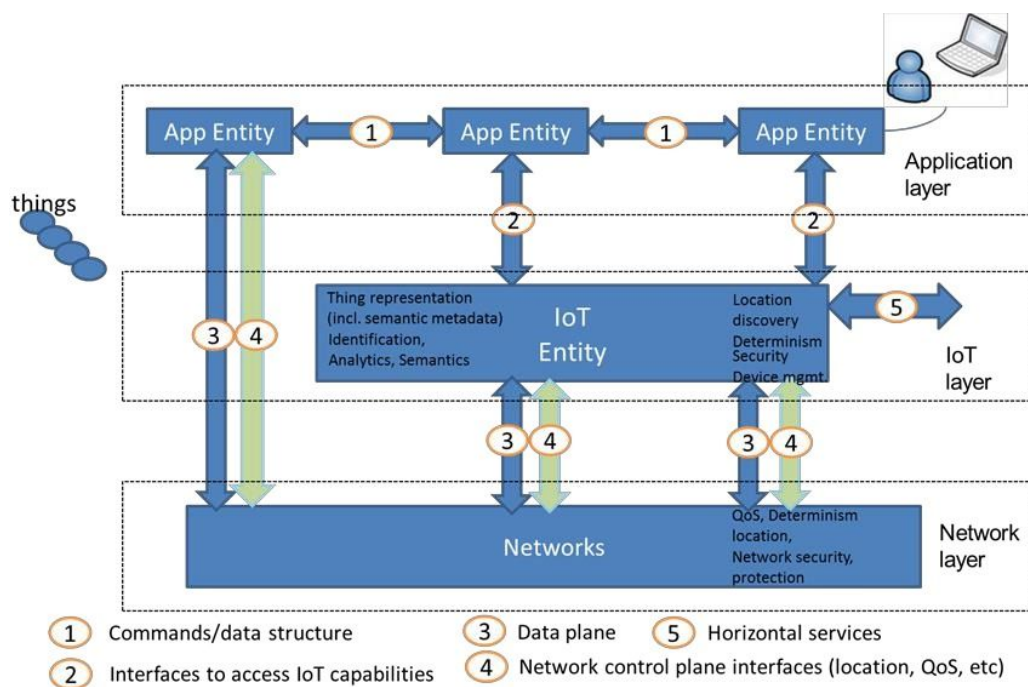
Está concebido como un diseño de 4 niveles, en el que tres niveles principales (Field, Edge y Cloud) se apilan jerárquicamente según su alcance con respecto a los procesos físicos de la fábrica, y se incluye uno adicional para explicitar el trabajo del DLT (*Distributed Ledger Technology*) en la arquitectura.

Ha sido muy relevante para el estudio del estado del arte de esta tesis, ya que incluye un ejemplo ilustrativo de paso de RA a plataforma real implementada, explicando la transferibilidad del *edge*, el encaje de las tecnologías NGIoT, las APIs abiertas, el almacenamiento, la seguridad, etc. Además, tiene unos requisitos muy bien estructurados que han contribuido a los enfoques adoptados en este trabajo (véase la sección 3.1.1). Asimismo, FAR EDGE RA se asienta en un diseño de microservicios, de forma similar a lo considerado en este trabajo. Finalmente, la implementación sugerida también incluye habilitadores FIWARE, alineando su

perspectiva con la mayoría de las decisiones finalmente tomadas para la solución de este doctorado (ver sección 3).

### AIOTI HLA

La *Alliance of Internet of Things Innovation* (AIOTI) fue iniciada por la Comisión Europea en 2015, con el objetivo de fortalecer la interacción entre los actores de la Internet de las Cosas (IoT) en Europa y acelerar la creación de un ecosistema europeo dinámico de IoT. Entre sus actividades, AIOTI tiene como objetivo la creación de una Arquitectura de Alto Nivel (HLA, del inglés *High Level Architecture*), que se centra en la especificación y consolidación de marcos de referencia, arquitecturas y estilos de diseño en el IoT y el Edge. La versión actual disponible del HLA de AIOTI es la 5, publicada en diciembre de 2020 [40]



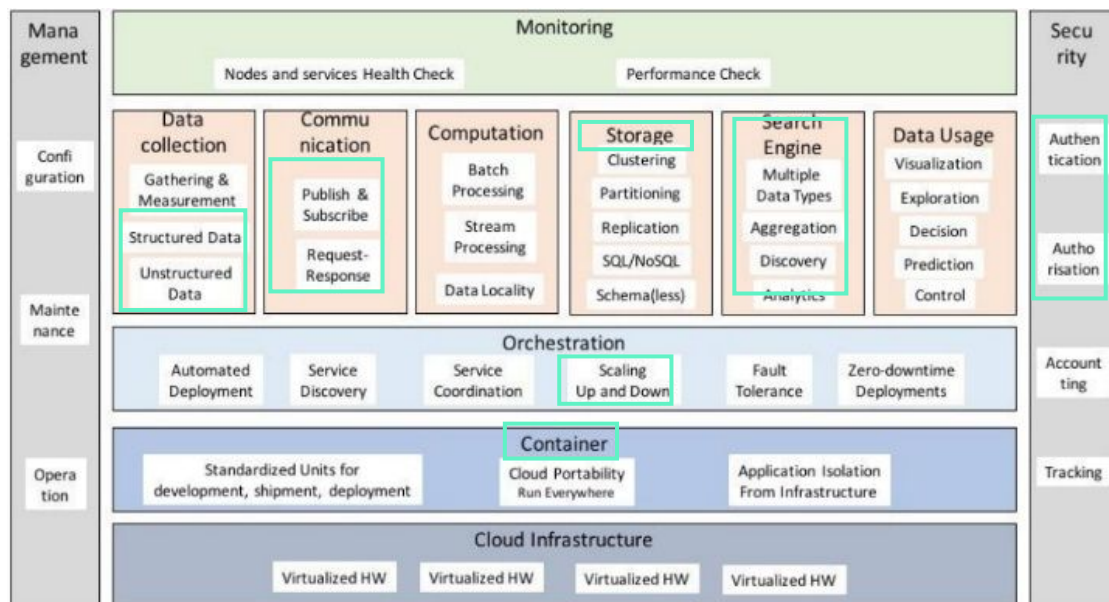
**Figura 2.11:** Arquitectura AIOTI. Fuente: [40]

La figura anterior representa el modelo funcional del HLA de AIOTI, que consta principalmente de tres *capas*. Se basa en la norma ISO/IEC/IEEE 42010 aborda preocupaciones como la privacidad, la virtualización, los Big Data y la IA en el IoT. La seguridad y la gestión también se tienen en cuenta, pero de forma intrínseca a las especificaciones de la interfaz en lugar de representarse como capas transversales. Además, el HLA de AIOTI está muy centrado en “la cosa”, mientras que el trabajo de este doctorado está más centrado en la ejecución de un servicio concreto: el cálculo de índices compuestos.

Un aspecto relevante para la tesis es la perspectiva software del HLA. A diferencia de otras referencias (por ejemplo, LSP IoT 3D), el HLA no se centra en la estructuración de capas para la construcción de la RA, sino en los procesos de software que deben tener lugar (*App Entity*, *IoT Entity*, interconexiones) y se ocupa



de ellos sin asumir que se ejecutan en un mismo nodo. Esto se relaciona directamente con los mecanismos de interconexión establecidos (vía APIs REST, exponiendo puertos o vía acuerdos o convenciones específicas entre módulos que deben comunicarse) -muy útiles para la arquitectura en este trabajo. AIOTI aborda el *edge computing* solo de forma superficial, mientras que explica que la seguridad y la privacidad en los despliegues modernos de NGIoT son “*packaged gifts*” manejados por mecanismos de seguridad estándar y DLT para la confidencialidad, la responsabilidad, la privacidad y la protección de datos. Por último, AIOTI apuesta explícitamente por un enfoque basado en microservicios, previendo una arquitectura descentralizada basada en la ejecución de microservicios combinados, aunque tiene en cuenta el “*riesgo de falta de estructura de la implementación resultante*”. Para solucionarlo, AIOTI hace una propuesta teórica que incluye los elementos más relevantes de la NGIoT. Se señalan aquellos que son relevantes para la arquitectura de la solución de la tesis.



**Figura 2.12:** Arquitectura AIOTI detallada. Fuente: [40]

Cabe destacar que el grado de abstracción varía mucho entre todas las arquitecturas de referencia analizadas en las páginas anteriores. Mientras que algunos no van más allá de las recomendaciones, otros especifican aspectos de implementación mucho más profundos, proporcionando incluso sus propios componentes de software para su realización. Las características típicas que se encuentran en las RAs de IoT (por ejemplo, la representación virtual de las entidades, la seguridad, la capacidad de gestión y los mecanismos de interoperabilidad) se consideran “características básicas”. Se puede observar que algunas características del NGIoT, como la virtualización de funciones, se incorporan comúnmente en muchas RA, mientras que otras características, como la comunicación de baja latencia o la SDN, no son tan comunes. Por último, algunos aspectos, como la aceleración por hardware o la DLT/Blockchain, sólo forman parte de unas pocas arquitecturas de referencia existentes.

En cualquier caso, hay que destacar que, a día de hoy, no existe una arquitectura única que aborde todas (o, al menos, la mayoría) de las características esperadas para despliegues NGIoT.

### Patrones de diseño

En esta parte de la subsección se repasan los diferentes patrones de diseño en arquitecturas de IoT. El objetivo es repasar sucintamente sus características y principales ventajas e inconvenientes de cara a seleccionar el más adecuado para el diseño de la arquitectura de esta tesis.

En las primeras etapas de despliegue, las aplicaciones de IoT eran sencillas y el número de elementos de IoT implicados era pequeño, por lo que los desarrolladores de IoT solían compartir la carga de contribuir y mantener el código base. Por lo tanto, enfoques básicos como el de *front-end* (interfaz y lado del cliente), *back-end* (donde ocurre la lógica) y base de datos (para almacenar los resultados) eran más que suficientes. A medida que el Internet de las Cosas de Nueva Generación (NGIoT) crecía, debían añadirse nuevas funcionalidades a las aplicaciones, lo que conllevaba (i) un aumento de la carga de trabajo operativa, y (ii) un necesario escalado horizontal y/o vertical, requiriendo que más servidores albergaran la aplicación, obligando por tanto a la creación de nuevos paradigmas que mejoraran los mencionados enfoques simplistas.

Los patrones arquitectónicos definen cómo organizar los componentes del sistema cuando se construye un sistema completo y se cumplen los requisitos establecidos por una actividad [41]. A continuación se enumeran las arquitecturas y los patrones de software más relevantes [42]:

- El patrón de la *arquitectura por capas se ajusta a* la comunicación tradicional, organizada en capas horizontales. Cada capa tiene una función específica dentro de la aplicación y puede comunicarse entre sí a través de interfaces definidas y diferentes topologías (p.ej., modelo OSI).
- *Microkernel* (patrón de arquitectura *plug-in*) es un patrón orientado a producto, que empaqueta el software como una pieza descargable, permitiendo añadir *plugins* para mejorar la funcionalidad del sistema (por ejemplo, Eclipse IDE).
- *Arquitecturas dirigidas por eventos (Event-Driven)* es un patrón asíncrono distribuido que (a través de un mediador) acepta datos y los envía (a través de un intermediario) a partes de código delegadas (por ejemplo, una página web de JavaScript).
- *Arquitecturas basadas en espacio*, diseñadas específicamente para abordar y resolver los problemas de escalabilidad y concurrencia, siendo un mejor enfoque arquitectónico cuando se piensa en un sistema muy variable en volumen requerido de BBDD.

- *La arquitectura sin servidor (serverless)* es un enfoque de *cloud computing* para construir y ejecutar aplicaciones y servicios sin necesidad de gestión ni mantenimiento de la infraestructura [43].
- El paradigma de la arquitectura *basada en servicios* se centra en la realización de funciones que descomponen los problemas complejos en una serie de servicios más simples [44], diseñados para ser desplegados por separado, lo que permite construir sistemas altamente escalables y resistentes. [45]. Existen tres enfoques principales del paradigma de arquitectura basado en servicios: (i) monolítico, (ii) orientado a servicios (SOA) y (iii) microservicios.

Como se puede observar, el candidato más apropiado para la solución en este doctorado promete ser el último, por lo que se analizan los tres enfoques:

**Las arquitecturas monolíticas** son modelos de software tradicionales que pretenden construir una aplicación de un solo nivel en la que los componentes se combinan en un único programa indivisible [46]. Su simplicidad permite evitar muchos de los problemas asociados a los sistemas distribuidos, lo que se traduce en una mayor simplificación de los flujos de trabajo de los desarrolladores, la supervisión y las pruebas. Normalmente, se trata de aplicaciones en las que todo el código se empaqueta en un único proceso, pero también puede darse el caso de aplicaciones que tienen varios módulos que deben desplegarse juntos como un sistema indivisible.

**Las arquitecturas orientadas a servicios (SOA)** son una evolución natural del patrón monolítico, desacoplando una aplicación en módulos más pequeños que se comunican a través de un bus de agregación [47] que se encarga de la mediación, la transformación de los mensajes y el enrutamiento entre los módulos. Los principios de la arquitectura orientada a servicios son independientes de cualquier producto, proveedor o tecnología, y permiten la interoperabilidad entre sistemas, servicios y aplicaciones [48], [49]. Como punto negativo, un bus de este tipo puede implicar ineficiencias informáticas y, además, todos los componentes deben desplegarse de forma dependiente.

El patrón de **arquitectura de microservicios** estructura una solución robusta como un conjunto de servicios ligeramente acoplados, pequeños y fáciles de entender que se aíslan en pequeñas unidades coherentes y autónomas [50] para resolver el problema de la arquitectura compleja y las redundancias de código, permitiendo el diseño y desarrollo independiente de los diferentes servicios [51]. A diferencia de SOA, en el patrón de microservicios no hay lugar para un bus de middleware, por lo que la comunicación entre los componentes se realiza a través de APIs o mediante la gestión posterior de APIs entre servicios y consumidores. [51] La comunicación entre los servicios y los consumidores se basa en mensajes ligeros y en APIs HTTP, REST o *thrift* [52]. Como aspectos negativos, la cadena de llamadas en los microservicios implica ciertas complejidades, obligando a los arquitectos a llegar a acuerdos y elaborar una sólida documentación de las APIs [53], [54]. Asimismo, la granularidad en el diseño plantea riesgos de desajustes,

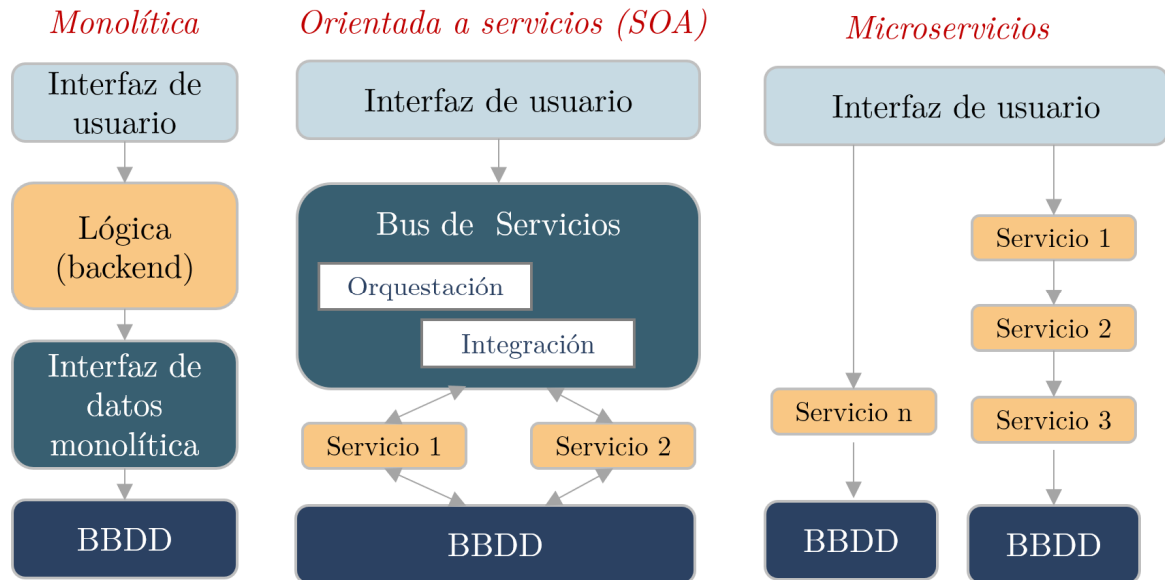
llamadas de red improductivas y puntos finales de seguridad adicionales [44]. Técnicamente, un microservicio tiene componentes internos similares al mencionado enfoque clásico, con un *front-end* que actúa como interfaz para enviar y recibir peticiones (punto final de la API), un *back-end* con la funcionalidad esperada y una base de datos, ya sea dentro de la memoria del microservicio o una base de datos externa.

En la siguiente imagen se aporta una comparación visual en términos de capas y componentes entre los enfoques descritos, mientras que en la siguiente tabla hay una breve referencia sobre cuál de ellos elegir en cada caso.

**Tabla 2.1:** Comparación entre arquitecturas monolíticas, SOA y microservicios

| Monolítico  | SOA   | Microservicios   |
|---|---|--|
| SÍ recomendado para entornos con las siguientes características   |   |  |
| Equipo pequeño o aplicaciones sencillas o cuando se requiere una inversión inicial baja.  | Servicios web, computación distribuida, <i>cloud computing</i> y SaaS.                            | Aplicaciones que puedan ser divisibles en pequeños servicios, despliegues que requieran cambios y escalabilidad y aquellos con equipos de desarrollo geográficamente dispersos.                                      |
| NO recomendado para entornos con las siguientes características   |   |  |
| Aplicaciones que requieren escalabilidad, desarrollo complejo previo a la ejecución o aquellos despliegues que prevén adaptaciones y cambios a lo largo del tiempo. | No es ideal para aplicaciones pequeñas, ya que no requieren componentes de mensajería middleware. | Los servicios deben ser en gran medida independientes o, de lo contrario, se deben llegar a demasiadas convenciones y existen riesgos.<br><br>Aplicaciones con componentes complejos que comprometen el rendimiento. |

Por definición, los despliegues de NGIoT deben estar preparados para adaptarse a los cambios, ya sea en cuanto a la incorporación de nuevos dispositivos a la instalación, la aplicación de nuevos servicios sobre los datos proporcionados por éstos o la modificación de la política de seguridad para su gestión, entre otros. Además, la escalabilidad también es clave en estos despliegues. Con una capacidad de computación creciente en dispositivos miniaturizados cada vez más pequeños, la estructura de un despliegue de IoT debe estar preparada para ajustar el procesamiento y las capacidades a lo largo del continuo, por lo que los microservicios parecen la opción más adecuada.



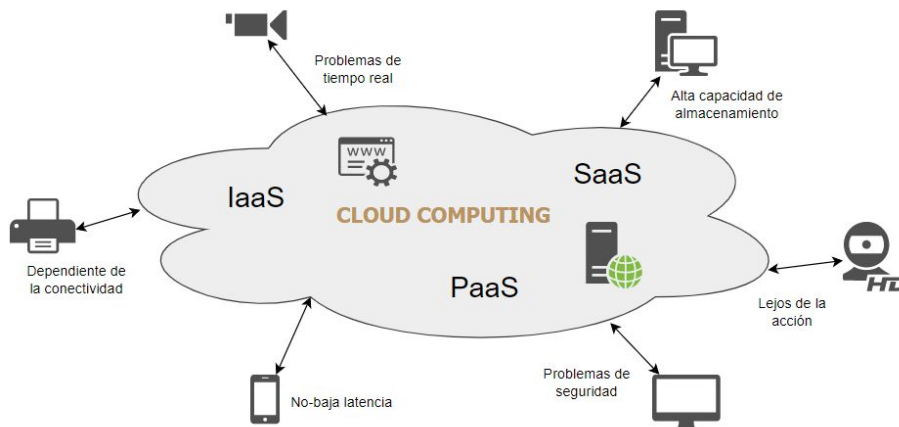
**Figura 2.13:** Comparación entre patrones de arquitectura

**Conclusión tras el análisis e influencia en la tesis doctoral:**

Se ha realizado una clara distinción entre plataformas IoT (destinadas a funciones como el descubrimiento y gestión de dispositivos, la conectividad, etc. y muy orientadas a despliegues específicos) y arquitecturas IoT, que tienen como objetivo proveer de una serie de recomendaciones de diseño e implementación para ser implementadas en diferentes entornos y características de servicio. La tendencia a la creación de arquitecturas es patente en los últimos años y coincide con la disminución de la aparición de plataformas, lo que responde a una mayor accesibilidad a tecnología *open source* y a una variedad de escenarios que no pueden responder con plataformas particulares. Dentro de éstas, se ha observado también un cambio entre las más empleadas en la primera fase del IoT (2014-2018), destacando FP7-IoT-A, IIRA y, sobretodo, RAMI4.0 y en las diseñadas durante la segunda fase, tras el advenimiento de las tecnologías de nueva generación (NGIoT). Desde entonces, arquitecturas de referencia como OpenFog, RAMEC (continuación natural de RAMI4.0), H2020 LSP IoT 3D o FAR EDGE incluyen de forma nativa consideraciones como el *edge computing*, las tecnologías DLT o nuevos protocolos de comunicación entre dispositivos. Las arquitecturas de referencia RAMI, FAR EDGE y AIOTI HLA son las más importantes de cara al diseño de la solución de la tesis, por su orientación a la aplicación, extensa documentación y objetivos de despliegue. Por otro lado, se han analizado los patrones de diseño de arquitecturas ICT (incluye IoT), donde se ha destacado aquellas basadas en servicios (descomposición de la aplicación en módulos simples e independientes que pueden ser instalados asincrónamente). Se ha expuesto una comparación entre ellas, destacando como la más apropiada, *a priori*, las arquitecturas de microservicios.

## 2.3. Edge computing en NGIoT

La computación en la nube (*Cloud Computing* - CC) ha dominado el ámbito del mundo TIC durante las últimas décadas. El CC propone un modelo de computación donde la mayor parte de las tareas son relegadas a una entidad central (normalmente propiedad de un proveedor de servicios - *cloud provider*) que se encarga del grueso del procesamiento. Dentro de esta filosofía, existen varias aproximaciones, utilizadas en función del objetivo de la utilización de los servicios *en la nube*: *Platform as a Service* (PaaS), *Software as a Service* (SaaS) o *Infrastructure as a Service* (IaaS) [55]. Apoyado en una alta fiabilidad gracias a la posibilidad de realizar copias de seguridad, una capacidad de almacenamiento prácticamente ilimitada, un ahorro de costes debido a la centralización, entre otros, junto con la escasa disponibilidad de verdaderas alternativas, CC sigue siendo hoy en día el paradigma para la computación de servicios distribuidos más extendido.



**Figura 2.14:** Perspectiva de cloud computing (CC)

Sin embargo, la llegada de la nueva generación de Internet (NGI), caracterizada por unas exigencias cada vez mayores de ancho de banda, latencia y potencia de cálculo, ha sacado a la luz algunas deficiencias de este paradigma. Analizando el defecto más evidente -la latencia-, el contexto es claro: en CC, el cálculo de los datos (todo lo que no sea un simple reenvío de información) se realiza normalmente a kilómetros del lugar real de la acción, pudiendo llegar a suponer retrasos comparados de 200 a 500 milisegundos, que pueden tener efectos críticos en determinadas aplicaciones [56]. Por otro lado, el defecto del ancho de banda parece ser una cuestión destacada a abordar de cara a las aplicaciones futuras. Los *wearables*, dispositivos de mano, drones, cámaras de vigilancia o sensores en general presentan flujos de datos cada vez más enriquecidos que ya ponen en cuestión los márgenes de transmisión habituales en el CC [57]. Otra de las principales preocupaciones en el CC es la privacidad. Con el modelo actual de CC, los datos deben ser enviados a servidores *cloud* situados, normalmente, a muchos kilómetros de distancia, circulando a través varias redes y elementos intermedios, donde la

seguridad de los datos puede estar comprometida. Por último, en CC existe una clara dependencia de la calidad de la conexión de la red y la disponibilidad del proveedor de servicios en la nube [58].

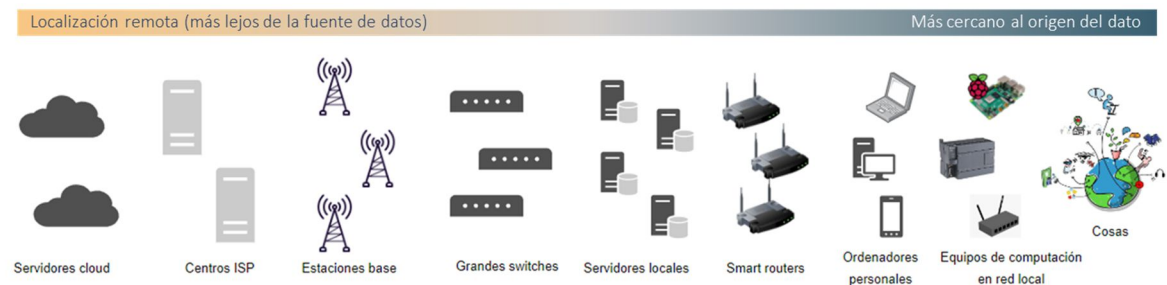
Además, es de esperar que estos (entre otros) se intensifiquen durante los próximos años como consecuencia de la nueva ola de tecnologías emergentes. Según el *Hype Cycle* de Gartner de 2021 [59], algunas de las tecnologías más populares en los próximos años serán la IA aumentada, la IA explicable (*explainable AI*), el *Data Fabric*, la realidad aumentada, el ML adaptativo y el 5G privado, que mapean casi exactamente con la lista de tendencias tecnológicas embarcadas en los sistemas del IoT de próxima generación (véase NGIoT en la sección anterior).

Además, tal y como se ha indicado en la sección anterior, se observa que tanto el paradigma CC como la aproximación tradicional (centralizada) de los despliegues IoT, carecen de las capacidades necesarias para manejar los nuevos requisitos de las aplicaciones centradas en el ser humano, la descentralización y el traslado de la inteligencia al extremo del despliegue (cerca del origen del dato).

En este sentido, es relevante observar el diluvio de datos esperado para los próximos años. Actualmente se estima que cerca del 45% de los datos del mundo se acercarán al borde de la red a finales de 2025 [60]. Este 45% se antoja como una cantidad enorme al observar las previsiones de generación de datos totales por parte de dispositivos, que se espera alcance alrededor de 80ZB procedentes de más de 40.000 millones de unidades de los mismos [61].

Teniendo en cuenta todo lo anterior, la conclusión fue la siguiente: muchos más dispositivos generarán datos cada vez más ricos y complejos que deben ser explotados lo antes posible (en términos de cálculo lógico) en aras de una mejor experiencia del usuario y del beneficio industrial en general.

En ese momento, se dedujo que la respuesta a todo lo anterior podría ser acercar la computación hacia el lugar donde se desarrolla la acción. Dentro de un esquema de computación distribuido pueden identificarse elementos generadores de datos, pasarelas o elementos intermedios, dispositivos de red (*routers, switches*), y unidades de procesamiento centralizadas (véase Figura 2.15: Continuo de computación (edge-fog-cloud computing)). En relación a este flujo bidireccional, el concepto de computación estudiado en esta tesis se refiere a este nuevo paradigma como “*edge-cloud computing continuum*”.



**Figura 2.15:** Continuo de computación (edge-fog-cloud computing)



Este término (que puede ser resumido directamente a “*edge computing*”) nace para definir el hecho de extender los recursos para ejecutar aplicaciones a través de diversas localizaciones que van desde entornos centralizados (lejos de la acción) hasta el borde o *edge* (más cerca de la fuente de datos), lo que en la práctica significa trasladar la carga computacional relevante elementos informáticos más cercanos al origen de la información. En otras palabras, más cerca de las “cosas” que producen y consumen datos (IoT).

Existe un gran debate sobre la terminología en el campo del *edge computing* (EC). En la búsqueda de una definición formal, se pueden encontrar algunos ejemplos en la literatura. El *Industrial Internet Consortium* (IIC) lo define de la siguiente manera “*El edge computing es aquella infraestructura informática descentralizada en la que los recursos informáticos y los servicios de aplicación pueden distribuirse a lo largo del continuo de computación desde la fuente de datos hasta el cloud*” [62]. Según M. Satyanarayanan, el *edge computing* es “*un enfoque para el análisis eficiente de datos contextuales en el que la computación se realiza en dispositivos de detección (sensores, actuadores, controladores, concentradores), conmutadores de red -switches- u otros dispositivos (pasarelas o gateways) en lugar de transmitir todos los datos a un entorno centralizado/cloud*” [63]. Esta divergencia terminológica es muy común en áreas en constante evolución como este [64]. En el contexto de este doctorado, el objetivo es permitir que el sistema IoT de la solución sea flexible para ser alojado y ejecutado en múltiples ubicaciones del continuo *edge-fog-cloud*, dependiendo de las necesidades particulares de cada escenario/despliegue.

En cuanto al “*edge computing*” como campo de investigación, se mencionó por primera vez en 2012 cuando F. Bonomi (CISCO) acuñó la “computación de niebla” (*fog computing*) como parte del paradigma. Un año después, se definió el *Mobile Edge Computing* (MEC) y otro año más tarde (2014), apareció el concepto de “*cloudlets*”. A pesar de ser un campo de investigación reciente (10 años), el *edge computing* ha ido ganando cada vez más atención tanto en el plano académico (más de 150 artículos en 2013-2022 en las revistas más destacadas) como por parte de las grandes compañías tecnológicas (Google, Cisco, Amazon, INTEL...) con muchos ejemplos diferentes y un sinnúmero de casos de uso potencialmente interesantes enmarcados en multitud de sectores [65]. No sólo se están anunciando nuevos proyectos innovadores (por ejemplo, a través de las convocatorias de H2020 ICT-12-2018, ICT-51-2020, HE-DATA-01-05 o HE-DATA-02-02), sino que también se han creado varias iniciativas, como el consorcio OpenFog, la iniciativa de computación OpenEdge [66] y la iniciativa *Multi-Access Edge Computing* [67] (MEC - promovida por el ETSI), cuyo objetivo es establecer arquitecturas y normas de referencia en este campo. Además, es un concepto que interesa a varios actores, desde los ISP o los proveedores de la nube hasta los diseñadores de sistemas IoT [68].

Todo lo anterior forma un entorno extremadamente complejo que hace técnicamente imposible la creación de una solución única para el *edge computing*.



Desde el principio, el EC se ha dividido en tres paradigmas principales de implementación: *Mobile Edge Computing* (MEC), *cloudlets* y *fog computing*. Todas ellas comparten la visión computación en los extremos de la red y se apoyan fuertemente en mecanismos como la virtualización, la containerización (véase sección 2.5.1), la gestión de recursos de seguridad y la monitorización continua. Sin embargo, las tres presentan claras diferencias de configuración, características y alcance.

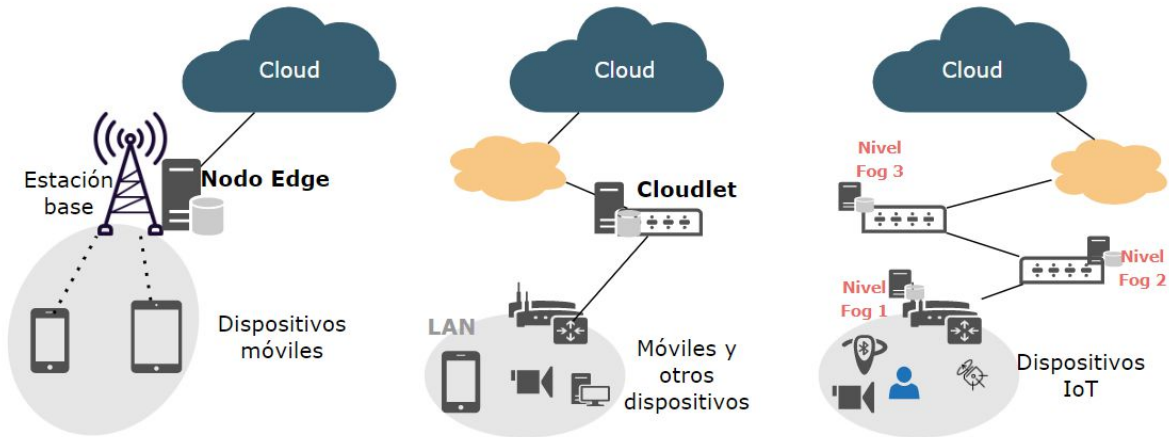


Figura 2.16: Esquemas EC: MEC, cloudlet y FC

El *Mobile Edge Computing* está asociado a las redes de acceso radio (RAN), donde el nodo *edge* siempre se encuentra en una estación base de comunicaciones móviles, cubriendo los dispositivos en su rango de celda. Esta implementación está orientada a proveedores de ISP y aprovecha técnicas de 4G-LTE y 5G como la virtualización de funciones para lograr beneficios de *edge computing*. Por otro lado, los *cloudlets* pueden entenderse como “réplicas” de las capacidades del *cloud* pero más cerca del borde de la red, reduciendo así la latencia, el tiempo de ida y vuelta y el consumo de ancho de banda. Los *cloudlets* se conciben como un pequeño *cloud* (“*cloud in a box*”) que se ejecuta sobre uno o un clúster de servidores con altas capacidades (recursos de computación) siguiendo la estructura canónica del *edge computing* de tres niveles. La computación de niebla (*fog computing* -FC), en cambio, pretende aprovechar la flexibilidad del IoT para realizar funciones de *edge computing*. Mediante el uso de “nodos *fog*”, que pueden extenderse a través del continuo del borde a la nube creando de 1 a N niveles “cercanos”, el *fog computing* contempla la localización de diferentes partes de la computación de un servicio en varias posiciones del continuo.

En [69] se puede encontrar una revisión de las diferencias entre estos paradigmas, incluyendo ventajas y desventajas. Como pequeña referencia, la siguiente tabla incluye un resumen comparativo conceptual. En el estudio de la referencia indicada, los autores definieron asimismo un árbol de decisión para seleccionar uno u otro en función de: la proximidad, los medios de acceso, el conocimiento del contexto, el consumo de energía y el tiempo de cálculo requerido.

**Tabla 2.2:** Diferencias en características entre MEC, cloudlets y FC

| Característica                    | MEC                                  | Cloudlets                         | FC                                |
|-----------------------------------|--------------------------------------|-----------------------------------|-----------------------------------|
| Ubicación del nodo de computación | Controlador RAN – estación base (EB) | En red local o en entorno próximo | Cualquier punto del continuo      |
| Proximidad a la fuente            | Un salto                             | Un salto                          | Uno o varios saltos               |
| Dispositivo HW nodo               | Servidor en EB                       | <i>Cloud in a box</i>             | Router, pasarela IoT, PLC, RPi... |
| Gestión de nodos                  | Orquestación móvil                   | Agentes <i>cloud</i>              | Orquestación, federación          |
| Mecanismos de acceso              | Red móvil (4G, 5G)                   | WiFi                              | BLE, WiFi, ZigBee, LoRa...        |
| Coste de adquisición              | Alto                                 | Alto                              | Bajo                              |
| Fiabilidad                        | Media                                | Media                             | Alta                              |

Aparte de estas tres divisiones principales, en los últimos años han surgido algunos conceptos nuevos para catalogar otros subparadigmas específicos de *edge computing*, como el *mist computing* (computación minimizada en la fuente de datos) [70] o el *dew computing* (que se apoya en servidores intermedios para catar como controladores de dispositivos IoT) [71].

El MEC existe desde 2013, cuando Nokia e IBM introdujeron servidor *cloud* de aplicaciones radio (RACS, del inglés *Radio Applications Cloud Server*), una plataforma de *edge computing* para ser ejecutada por los servidores en las estaciones base en las redes celulares 4G/LTE. El nuevo nombre (*Multi-Access Edge Computing*) se ha adoptado recientemente fruto de los esfuerzos de estandarización que está realizando el ETSI a través de un grupo de trabajo específico. La base de MEC es permitir que la computación en las redes celulares se realice más cerca del dispositivo móvil (en el *edge*), permitiendo así reducir la latencia en un entorno altamente dependiente de la ubicación. En las redes 3G, el MEC tiene lugar en el controlador de red de radio (RNC, *Radio Network Controller*), mientras que en las redes 4G se encapsula en las estaciones eNodeB y, en el panorama 5G [72] en los nodos gNodeB. Una de las principales ventajas de MEC es que el servidor *edge* (en el momento del *offloading*) dispondrá de información precisa y en tiempo real sobre todos los parámetros de la red, incluidos la carga y la capacidad, y al mismo tiempo conocerá las características de los dispositivos finales. No obstante, considerando los potenciales escenarios de aplicación de la solución de este trabajo (donde no se espera la introducción de elementos de radio), se ha descartado esta variante del *edge computing*.

Por otro lado, el término *cloudlet* fue acuñado por el Prof. M. Satyanarayanan de la Universidad Carnegie Mellon en 2014 [73] se refiere a aquellas implementaciones que crean pequeños centros de datos en el borde de la red (*cloud in a box*) [74]. Los *cloudlets* pretenden replicar exactamente el funcionamiento

clásico de proveedores *cloud* (por ejemplo, AWS, Azure, OpenStack), articulándose como un segundo nivel -más cercano a la fuente de datos- con caché declarada imitando las funciones *cloud* pero de una forma geográficamente distribuida, en lugar de centralizada (no limitado a estaciones base como en MEC). Algunos ejemplos comerciales relevantes son los servicios provistos por AWS Lambda o Akamai. Sin embargo, el obstáculo más preocupante que dificulta el despliegue masivo de *cloudlets* es la abrumadora variedad de implementaciones (la mayoría experimentales) sin una referencia clara. Este hecho está impidiendo que este paradigma alcance una gran adopción [75]. Además, este paradigma parece presentar algunas desventajas como para considerarse los casos de validación planteados en este doctorado. Supone el establecimiento de complejos mecanismos de computación y descarga de datos, lo que implica la necesidad de negociar con el proveedor de CC el alcance específico del servicio que se puede derivar a los *cloudlets*. Además, es muy difícil disponer de una infraestructura global sólida y consolidada de *cloudlets* en todas las ubicaciones de una red, lo que debería ser el objetivo para garantizar una "experiencia de latencia sin fisuras" en los despliegues de *IoT-edge computing*.

*Fog computing* (FC), en cambio, es el paradigma de EC que se relaciona de manera más directa con los escenarios y aplicaciones de IoT. Dicha arquitectura requiere que el procesamiento se incruste tanto directamente en los puntos finales – “cosas”- (por ejemplo, sensores), como en las pasarelas (*gateways*) u otros equipos en lugares de agregación cercanos. De ahí que el FC pueda concebirse como un escenario de varios niveles en el que el procesamiento de datos puede extenderse a través de múltiples ubicaciones a lo largo del continuo. Además, el FC es el único enfoque que acepta protocolos de acceso no basados en IP (como MQTT) y, de hecho, proporciona el enfoque más flexible para entornos dinámicos, escalables y en crecimiento. Las implementaciones de *fog computing* requieren un nivel de diseño customizado, necesitando mapear los requisitos particulares de la aplicación y el equipo disponible antes de proceder al despliegue, pero ofrecen muchas opciones y es el campo de investigación activo de mayor interés para este doctorado.

Por lo tanto, a lo largo de este manuscrito se utilizarán el término “*edge computing*” (EC) para referirse al enfoque de “*fog computing*” (FC).

En cuanto al estudio de este campo, el término *fog computing* fue introducido por primera vez por F. Bonomi -de Cisco- en 2012 [76]. La etimología de este paradigma responde a la implementación de “nodos de niebla” (*fog nodes*) que pueden estar distribuido en varios puntos del continuo para mejorar las deficiencias de la CC en las aplicaciones de IoT. Según el NIST, “*el FC es un modelo en capas para permitir el acceso ubicuo a un continuo compartido de recursos computacionales escalables*”. Otra definición interesante se esboza en [77]: “*concebimos el FC como un puente entre la cloud y el edge que pretende facilitar el despliegue de nuevas aplicaciones emergentes del IoT*”.

En definitiva, el objetivo del enfoque FC es desplegar la computación en ubicaciones (potencialmente más de una) más cercanas a la fuente, al tiempo que

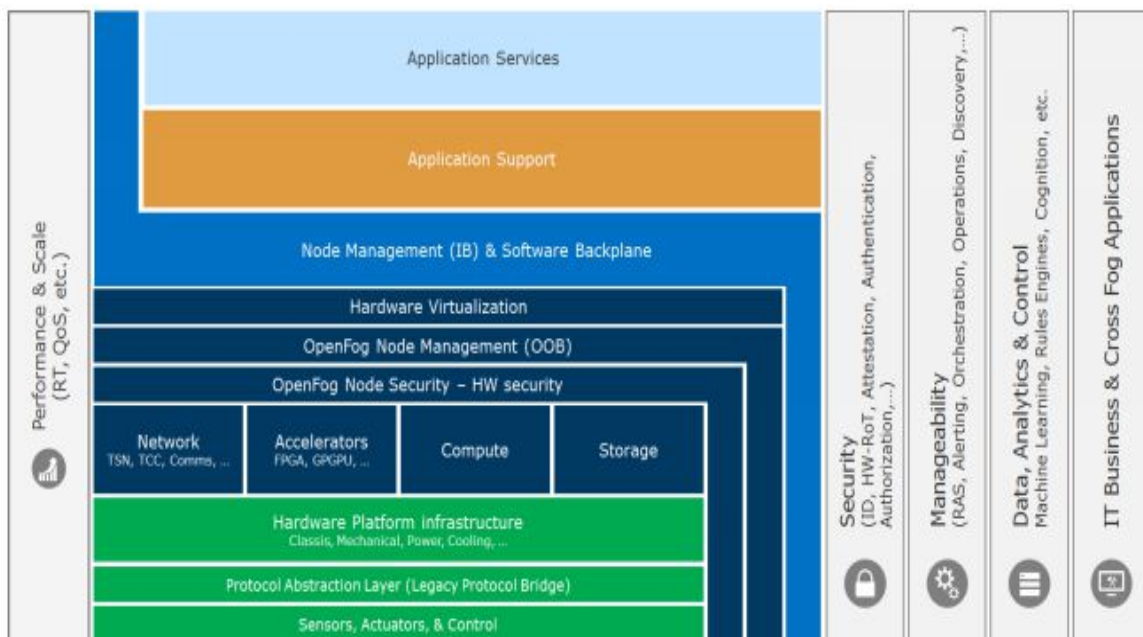
se aplican los conceptos de IoT a las capas superiores de la arquitectura de red, mejorando así la escalabilidad de la infraestructura de IoT. El *workload* cedido al entorno *edge* es ejecutado por nodos “fog” que se materializan en uno o más equipos (hardware) heterogéneos (por ejemplo, un ordenador portátil, un servidor local, una Raspberry Pi [78], etc.) que deben desempeñar un determinado papel en el servicio/despliegue en función de los recursos de dicho nodo, los objetivos de la aplicación, la capacidad de la topología global, etc. Cada nodo *fog* ejecuta piezas de software altamente virtualizadas de tal manera que pueden equivaler a un nodo IoT actuando tanto como de *gateway* como de elemento activo de procesamiento.

En otro orden de cosas, el FC presenta una diferencia clave con el resto de enfoques de EC. Aparte de la comunicación ascendente-descendente entre los niveles de dispositivo-*edge-cloud* (en FC, de *fog* a *cloud* y de *cloud* a *fog*), el FC permite que los nodos *fog* cooperen y se comuniquen entre sí formando una red [79]. Así, un despliegue de FC puede funcionar de forma semi-centralizada (incorporando elementos centrales) o totalmente descentralizada, diferenciadas en si los nodos son orquestados (el primer caso) o federados (el segundo caso), interactuando entre ellos y proporcionando así escalabilidad horizontal. En lo que respecta a este doctorado, este concepto posee especial relevancia. Uno de los objetivos de este trabajo es permitir un despliegue flexible de la solución (véase la sección 3), previendo una implementación completamente *edge*, la instalación parcial de los módulos de la arquitectura en la ubicación central mientras se mueve cierto procesamiento más cerca de la fuente de datos (agentes...), etc. Además, el FC parece el paradigma adecuado en el que basarse cuando se propone un escenario de aprendizaje federado. En el caso de este doctorado, el *federated learning* ha sido considerado un campo de investigación de interés, por lo que centrarse en *fog computing* parece la elección correcta de cara a futuras aplicaciones de aprendizaje federado, asegurando consistencia con los requerimientos de dichas técnicas.

El uso de FC frente a los servicios tradicionales de CC proporciona ventajas sustanciales. Entre las más relevantes, destaca la latencia, ya que el nodo de computación puede estar lo más cerca posible de la fuente y del usuario. Además, el hecho de no estar limitado a servidores específicos (como los *cloudlets* o el MEC en las estaciones base), hace que la FC pueda hacer frente a la actual heterogeneidad de equipos en la Industria y otros sectores (por ejemplo, *Smart Port-City*, principal sector de validación en este doctorado). En un despliegue de FC pueden existir datos procedentes de múltiples fuentes, en múltiples formatos y pasando por equipos variopintos (desde equipos *legacy* hasta equipamiento de procesamiento más moderno) haciendo uso de varios estándares de comunicación (por ejemplo, MQTT, ZigBee, BLE, TCP, HTTP, WiFi). Otros rasgos esenciales del FC son la escalabilidad, la modularidad y la flexibilidad. Según las referencias estudiadas, un nodo *fog* suele hacer uso de las tecnologías de virtualización y containerización (por ejemplo, Docker), lo que permite un despliegue fluido de componentes ad-hoc en tiempo de ejecución. Esta flexibilidad también se manifiesta en la capacidad de la FC para crear despliegues específicos en función del hardware,

el software y los protocolos de comunicación disponibles [80]. Por último, la seguridad se ve claramente reforzada con la FC, cuyo despliegue puede incluir reglas de seguridad específicas a aplicar en cada nodo, incluyendo autenticación, cifrado, filtrado y otras. Todos estos aspectos son de declarado interés para esta tesis. De hecho, la escalabilidad, la flexibilidad, la modularidad y la seguridad se incluyen dentro de la lista de requisitos técnicos imprescindibles de la arquitectura de la solución (sección 3.1.1.1). Por lo tanto, resultó claro que los conceptos (y, si es posible, las tecnologías) del paradigma de *fog computing* debían considerarse partes esenciales del diseño de la solución.

Revisando la literatura, se han encontrado varias especificaciones sobre arquitecturas para el *fog computing*, especialmente de 2015 en adelante. Sin embargo, una arquitectura de referencia prevalece sobre el resto: OpenFog [81]. Según su definición: “*La arquitectura de referencia OpenFog describe una plataforma de FC genérica que está diseñada para ser aplicable a cualquier mercado o aplicación vertical*”, entre los que se incluyen como ejemplos representativos las ciudades inteligentes, el transporte, la seguridad y vigilancia visual y los edificios Inteligentes. Esto fue un gran aliciente para considerar OpenFog como inspiración para la solución en esta tesis, teniendo en cuenta los escenarios de validación seleccionados (*Smart Port-Cities*).

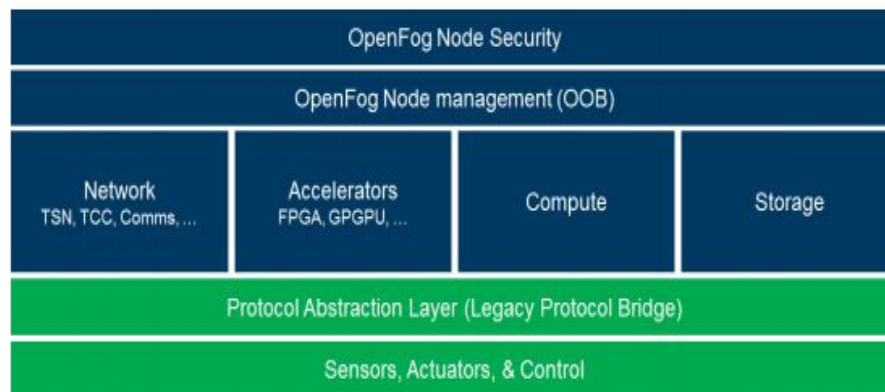


**Figura 2.17:** Arquitectura de referencia plataforma OpenFog. Fuente: [81]

La arquitectura de OpenFog está estructurada en pilares (seguridad, escalabilidad, apertura, autonomía, programabilidad, fiabilidad, agilidad y jerarquía), desplegados a través de N-niveles de nodos *fog* que interactúan entre sí para formar un sistema completo. El número de capas y niveles debe determinarse en función requisitos de los casos de uso y de los recursos disponibles. Según

OpenFog, un sistema de FC está compuesto por una plataforma (que incluye todas las capas, funcionalidades y módulos de la Figura 2.18: Esquema de un nodo *fog* según OpenFog. Fuente: [81]) que se instala (con flexibilidad modular) de forma diferente en varios nodos *fog*.

Mientras que una plataforma de FC incluye software transversal como la capacidad de gestión, las especificidades de las aplicaciones, el control de la seguridad y el análisis de datos, la implementación de un nodo *fog* básico sólo necesita incluir capacidades de conectividad y red, capacidad de computación y capacidad de almacenamiento, así como mecanismos de seguridad básicos y una interfaz de gestión del nodo para interactuar con el elemento central. Uno o varios nodos *fog* componen el sistema global de FC. La imagen siguiente representa una abstracción del software básico de un nodo *fog* tal y como lo proporciona la arquitectura de referencia OpenFog:



**Figura 2.18:** Esquema de un nodo *fog* según OpenFog. Fuente: [81]

El concepto de nodo *fog* no es original ni único de la arquitectura de OpenFog. Todos los enfoques y despliegues de FC se basan en la creación de nodos que llevan a cabo el procesamiento *fog*. En todas las implementaciones encontradas, estos nodos siempre cumplen el siguiente conjunto de requisitos (i) tienen la capacidad de ser autónomos, tomando decisiones locales a nivel de nodo, (ii) son modularmente heterogéneos, pudiendo implementar todos o algunos módulos de una plataforma de FC, (iii) siempre están estructurados de forma jerárquica, que puede ser vertical u horizontal, (iv) incluyen capacidades de gestión, en concreto orquestación de recursos y/o federación, (v) alto nivel de programabilidad, ya que pueden orientarse a resolver problemas específicos que pueden necesitar un desarrollo a medida a diferentes niveles.

Es conveniente detenerse aquí para señalar que este es el enfoque que se sigue en esta tesis. A pesar de la gran variedad de plataformas FC disponibles, no se consideró necesario depender de ninguna de ellas en concreto para los escenarios de validación para ejercer el procesamiento *edge computing* esperado. El diseñador/implementador/instalador de la solución diseñada puede elegir si utilizar o no una plataforma (de las que se señalan a continuación u otra), pero no existe



obligación de hacerlo. Siempre que el nodo *fog* cumpla con las características mencionadas, el cumplimiento y la aceptación están garantizados. De este modo, la arquitectura diseñada para la solución de esta tesis no incluye una plataforma de FC como elemento estructural, sino que contempla explícitamente el desplazamiento de cierto procesamiento a un posible nodo *fog*, cuya materialización debe escogerse en función de cada caso.

A continuación, se presenta una lista de las implementaciones de plataformas FC más relevantes y ejemplos de despliegues de FC que han sido analizados durante la actividad de estudio del estado del arte. Se acompañan de una imagen ilustrativa y una descripción sucinta:

- Cisco IOx [82]. La primera plataforma de FC puesta en marcha, con el objetivo de alojar la ejecución de servicios en equipos basados en Linux a lo largo de la red, utilizando un SO anfitrión (*host*) y máquinas virtuales para ejecutar funciones específicas.
- ParaDrop [83], procedente de la Universidad de Wisconsin-Madison, fue uno de los primeros despliegues *open source* de FC, diseñado inicialmente para ser ejecutado por puntos de acceso WiFi.
- La Fundación Linux se encuentra unificando una serie de proyectos de FC en el marco de la iniciativa *LF Edge*, cuyo objetivo es establecer un marco abierto e interoperable para que el EC sea independiente del hardware (eliminar *vendor-lock silos*). *EdgeX Foundry* [84] es el *framework* que está creando la Fundación Linux para el despliegue integral de los servicios de FC. En realidad, Foundry se compone de un conjunto de herramientas de código abierto que pretenden abordar funcionalidades específicas que, en conjunto, forman su arquitectura de plataforma de FC.

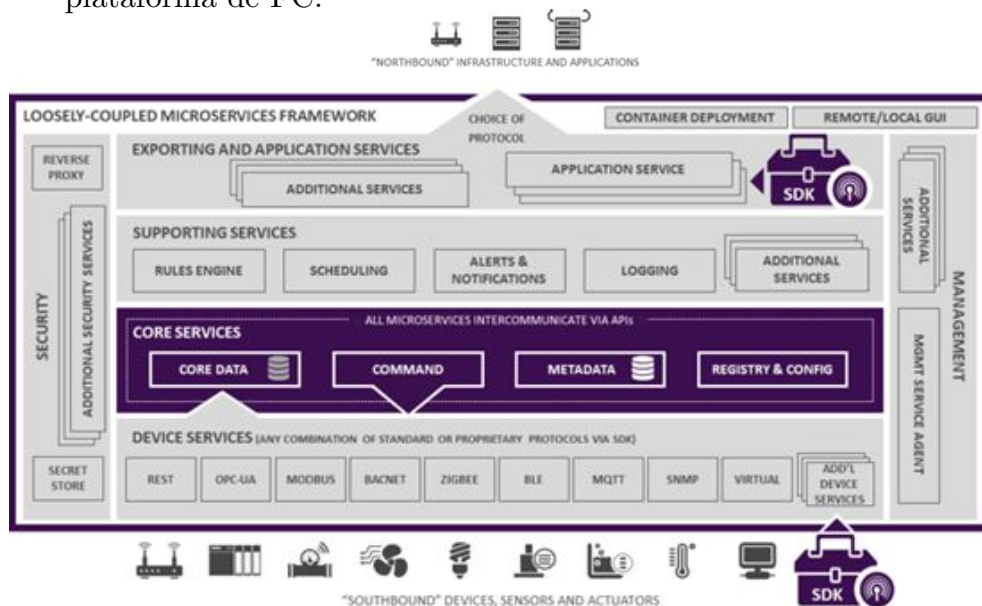


Figura 2.19: Arquitectura LF Edge – EdgeX-Foundry. Fuente: [84]

- *FogLAMP – Fledge* [85] actualmente llamado *Fledge* tras adherirse a la iniciativa de la Fundación recién mencionada. Fledge pretende ser la **implementación de nodo fog de referencia** de la plataforma EdgeX. Se basa en la gestión y utilización de nodos *fog* por las aplicaciones a través de una API y establece el aumento de funcionalidades de los mismos a través de *plug-ins* específicos de procesamiento norte, sur o interior.

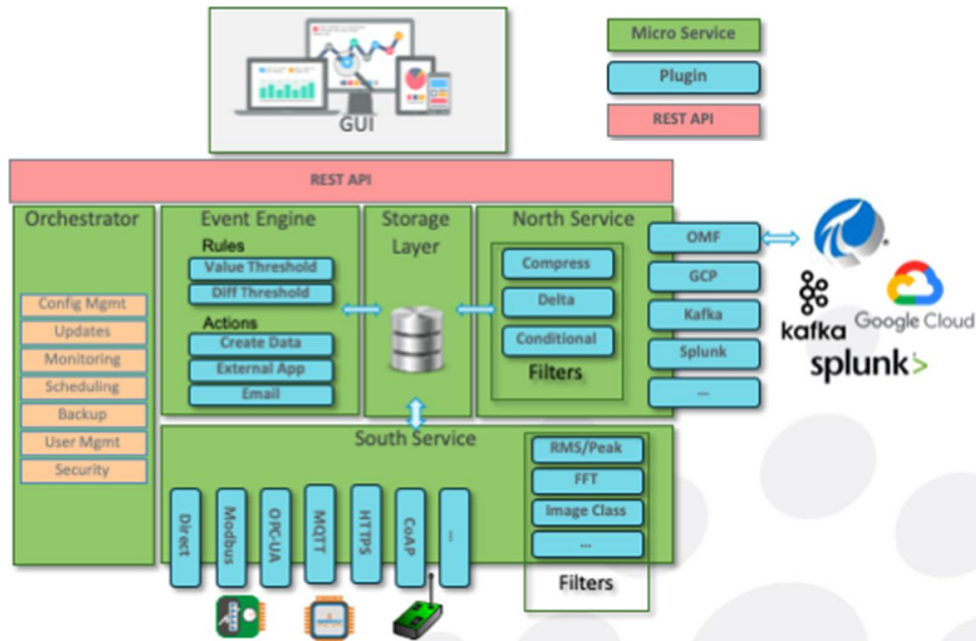


Figura 2.20: Nodo fog Fledge. Fuente: [85]

- *FogAtlas* [86] (procedente de la antigua *FoggyPlatform*) y contribuido por los proyectos H2020 ROMA y H2020 DECENTER es un *framework* para la implementación de la computación en la niebla orientado a proporcionar una provisión de servicios IaaS y PaaS “zero-touch” (*plug-and-play*) en *dispositivos edge*. Proporciona un software para orquestar nodos *fogh*, gestionándolos como recursos virtualizados tipo *cloud*. La especificación de FogAtlas se divide en módulos que se basan en tecnologías de código abierto como OpenStack, Docker, Kubernetes y Ansible, Prometheus y Grafana. Se observó durante toda la duración del doctorado, ya que se está posicionando como la herramienta de referencia de utilización en proyectos de investigación europeos.



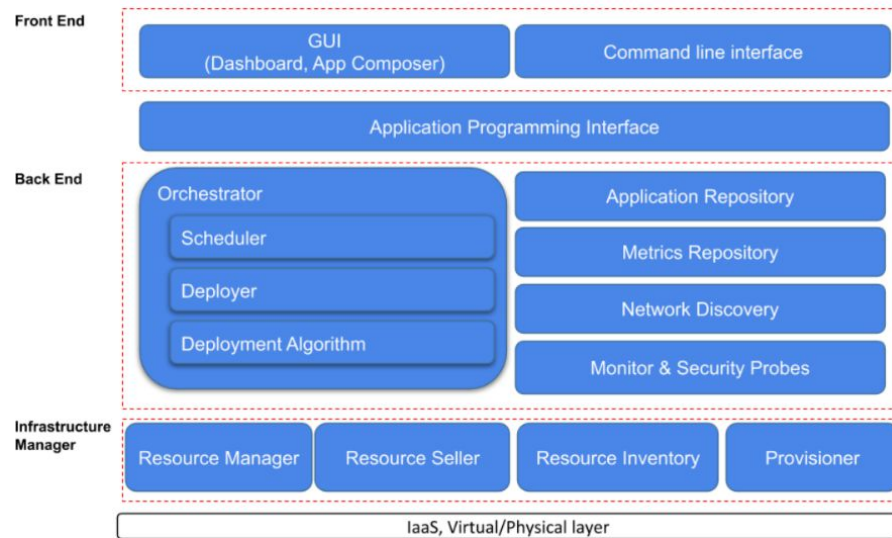


Figura 2.21: Nodo fog FogAtlas. Fuente: [86]

- Gateway* de H2020 INTER-IoT como nodo *fog*. Desarrollado por miembros del grupo de investigación donde realiza su investigación el candidato, el proyecto INTER-IoT [13] desarrolló una pasarela IoT capaz de realizar funciones cercanas a la fuente de datos IoT que a través de equipamiento *edge* de bajos recursos (por ejemplo, RPi). Este producto es especialmente relevante ya que introdujo un nuevo paradigma para las pasarelas IoT que se ajusta a este nuevo patrón de comunicación: la pasarela IoT dual física-virtual, comunicada a través de sockets web. Mientras que la parte virtual se sitúa en la capa *cloud* de una estructura de EC, las capacidades físicas se ajustan exactamente al papel de los nodos *fog*.

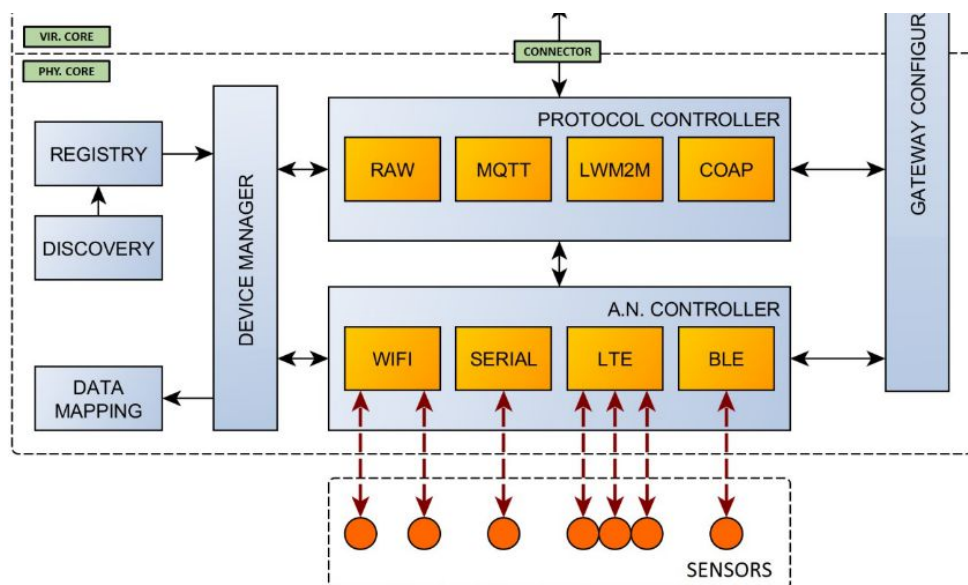


Figura 2.22: Nodo fog pasarela H2020 INTER-IoT. Fuente: [13]

- Eclipse lanzó en mayo de 2020 la primera versión de su solución “*fog05*” [87]. Al igual que Paradrop, se basa en contenedores LXD y en la integración con Kubernetes, pero siguiendo únicamente el modo de enfoque descentralizado.
- A principios de 2021, se anunció la plataforma de computación FORA [88], que se centró en la inserción de las nuevas tendencias de computación y comunicación como TSN, OPC UA- 5G y OpenStack dentro de una plataforma FC diseñada para aplicaciones de IoT industrial basada en un lenguaje de diseño de análisis de arquitectura. Tiene un conjunto de características prometedoras, pero resultó en un producto fuera del alcance de los objetivos de esta tesis.

En cuanto a las tendencias globales, las últimas plataformas observadas (o simplemente la implementación de nodos *fog*) se esfuerzan por construir sobre la pila de tecnología de Kubernetes. Conscientes de que la virtualización y la orquestación de contenedores es también el futuro en el sector EC, se ha identificado últimamente la necesidad de avanzar en este sentido para posibilitar la compartición de la computación, considerando que los nodos *fog* no deben limitarse a un vertical/servicio, pudiendo alojar simultáneamente varias aplicaciones de varios proveedores/consumidores. En este sentido, aunque se están publicando nuevas iniciativas en la literatura apoyándose en Kubernetes CRD [89], como StarlingX, KubeEdge, y Kubefed, es una línea abierta que habrá que observar durante los próximos meses/años.

El candidato realizó un estudio más exhaustivo del estado del arte en el marco del proyecto ASSIST-IoT [32].

#### **Conclusión tras el análisis e influencia en la tesis doctoral:**

Se ha realizado un estudio sobre cómo el *edge computing* cubre ciertas carencias del *cloud computing* como latencia, ancho de banda, seguridad y descarga de procesamiento en entidades centrales. Se han identificado los tres subparadigmas principales: MEC, *cloudlets* y *fog computing* (FC). Dentro de este último es donde se encuadra el interés en esta tesis doctoral, ya que es el que se relaciona directamente con los despliegues/aplicaciones IoT, identificando como potenciales elementos de computación aquellos nodos con capacidad de procesamiento situados en cualquier punto del continuo *edge-to-cloud*. El paradigma FC ofrece características de escalabilidad, la flexibilidad, la modularidad y la seguridad que se incluyen dentro de la lista de requisitos técnicos imprescindibles de la arquitectura de la solución. En este sentido, se ha tomado como referencia en concepto de nodo *fog* de la arquitectura OpenFog, donde se han definido ciertos requisitos para ser considerado como tal. La arquitectura diseñada para la solución de esta tesis no incluye una plataforma de FC como elemento estructural, sino que contempla explícitamente el desplazamiento de cierto procesamiento a un posible nodo *fog*, cuya materialización debe escogerse en función de cada caso.

## 2.4. Aprendizaje automático para indicadores compuestos

### 2.4.1. Predicción de series temporales en NGIoT

El aprendizaje automático (de ahora en adelante ML, del inglés *Machine Learning*) es un campo de estudio que busca extraer conocimientos o sintetizar conceptos de la experiencia de manera automatizada. El objetivo es, partiendo de una información histórica (que puede ser muy vasta y compleja), reducir la complejidad de interpretación, de descubrir patrones, detectar anomalías y de predecir valores futuros.

Una primera definición formal puede encontrarse en una cita de Arthur L. Samuel en 1959 [90]: “*El ML es el campo de estudio que da a los ordenadores la capacidad de aprender sin ser programados explícitamente*”. Esta frase (interpretada del texto original) ilustra la esencia del ML, que pretende “aprender” de forma automática conocimientos útiles sobre información original. Para efectuar este aprendizaje, el ML se basa en el uso de modelos, que son instrumentos matemáticos complejos aplicados sobre los datos para obtener dichos fines. En cuanto a su generación y aplicación, ha habido una evolución desde el comienzo de la investigación sobre dichos modelos, que data de hace más de 40 años [91], incluso más antiguamente en el campo de la estadística y econometría financiera.

Existen muchos modelos, aplicados a muchos tipos de datos (valores, textos, imágenes...). Dentro de aquellos que actúan sobre valores (contexto de esta tesis), pueden clasificarse en los siguientes grandes tipos:

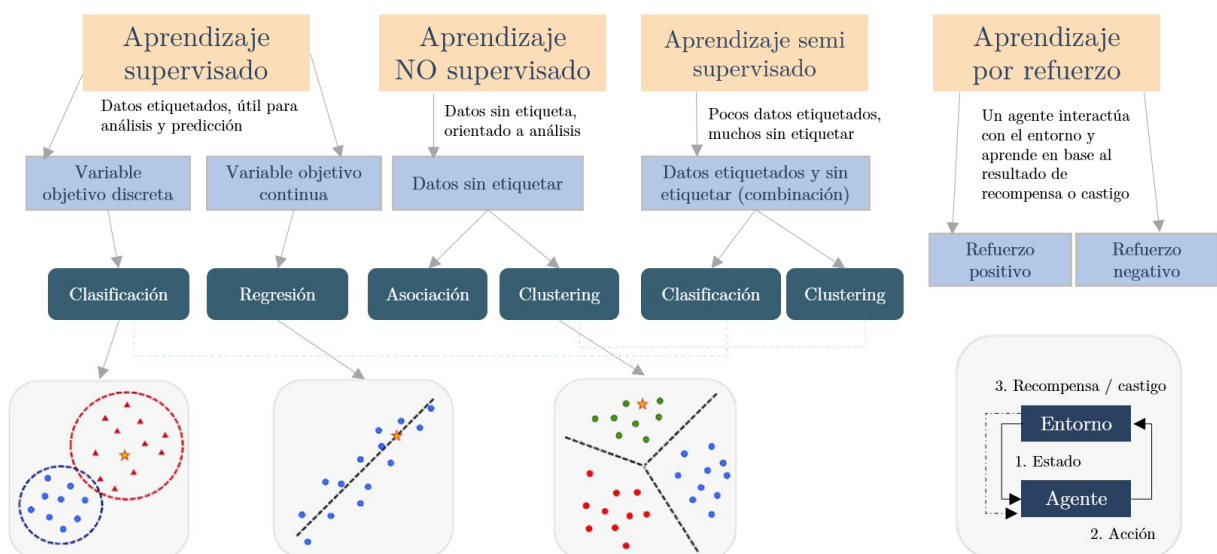


Figura 2.23: Clasificación de tipos de ML. Fuentes: [550][551]

El aprendizaje supervisado está diseñado para “aprender” a mapear entre una variable de entrada y una variable de destino a partir de ocurrencias anteriores etiquetadas. Se puede clasificar en problemas de regresión (lo que se quiere predecir es un número) y de clasificación (lo que se quiere predecir es una categoría o clase discreta). El aprendizaje no supervisado está orientado a descubrir patrones no detectados en un conjunto de datos sin etiquetas preexistentes [92]. El objetivo es identificar grupos de datos con características similares. Por otro lado, el aprendizaje por refuerzo parte de la base de optimizar un “premio” obtenido tras la aplicación del modelo, sin datos etiquetados ni descubrimiento de patrones.

Dentro de los entornos de IoT (alcance de esta tesis), la investigación sobre modelos de ML ha sido ampliamente explorada y aplicada durante los últimos 10-15 años. Existe una gran cantidad de casos de uso y ejemplos en muchos sectores [93]. Los problemas más frecuentes que se requiere resolver en estos entornos son de localización, clasificación, *clustering*, agregación, detección de anomalías en los datos de sensores o predicción de series temporales (para, por ejemplo, mantenimiento predictivo o congestión de tráfico). Por ello, los modelos más usados en IoT son SVM (*Support Vector Machine*), k-nearest neighbours (KNN) [94] o algoritmos de regresión lineal, aunque se está observando una tendencia al uso de redes neuronales [95]. No obstante, se considera un campo en continua evolución y donde se esperan innovaciones en los próximos tiempos, ya que todavía existen retos como el ajuste de modelos a entornos con baja capacidad computacional, un uso más eficiente de la energía de los dispositivos, el aumento de la precisión o la incorporación de modelos tradicionalmente reservados a otros entornos (por ejemplo, de imágenes o procesamiento de lenguaje natural), donde no es necesario encontrar un compromiso entre rendimiento y exactitud de los modelos [96].

Es muy importante realizar una reflexión sobre el problema donde se enmarca esta tesis doctoral con la intención de proveer información correcta sobre el estado del arte. Para comenzar, es conveniente entender que los datos provenientes de sensores (y el resto de información considerada) será constituida dentro de la arquitectura de esta tesis como indicadores (véase sección 4.2.1). En este sentido, estos datos pueden ser considerados series temporales (*time series*), ya que se trata de entradas con formato  $\langle \text{marca de tiempo}, \text{valor} \rangle$ , registrando cronológicamente los estados o valores numéricos de dichos indicadores [97], [98].

Dentro de la clasificación expuesta en la figura anterior, los modelos sobre series temporales constituyen una disciplina de ML encuadrada dentro del aprendizaje supervisado.

Estos modelos pueden cubrir diversos objetivos, que podrían ser categorizados de la siguiente forma: (1) análisis estadístico de las series temporales, (2) minería de datos en las series temporales, (3) búsqueda y exploración de las series temporales y (4) predicción (*forecasting*) de series temporales.

Las tres primeras categorías están orientadas a extraer información relevante de los datos, así como características que faciliten la interpretación del comportamiento de la serie temporal. Algunos ejemplos de estas categorías son la

función autocorrelación, correlación cruzada entre variables, autorregresión, etc. Existen grandes avances acerca de estos métodos en entornos IoT, como por ejemplo en el descubrimiento de tendencias, ciclicidad, descubrimiento de estacionariedad o para la detección de anomalías (que es el campo más explorado y aplicado de ML en series temporales de IoT) [98], [99], así como evidencias en la literatura científica en los últimos años [100]. Como nota relevante, los modelos empleados aquí (por ejemplo, SVM o Random Forest)) no son viables para el cuarto objetivo (predicción), que, por el contrario, buscan construir un modelo que pueda predecir ocurrencias futuras a partir de los datos anteriores observados [101].

Como anuncia el propio título de esta tesis doctoral, lo que se ha cubierto en este trabajo es cubrir la predicción de índices. En particular, tal y como se expone en la sección 4.2.3.1, se busca realizar la predicción a corto plazo (relevante para la selección del modelo) de indicadores (*cKPI values*). Estos indicadores son valores numéricos continuos, por tanto, se trata de un problema de regresión. En este sentido, los modelos de *forecasting* de series temporales más típicos plantean la construcción de un modelo expresado con una ecuación matemática que, a partir de dos o más variables independientes, predice el resultado de variables continuas [101].

Los modelos de regresión (un campo muy establecido en la literatura) pueden dividirse en tres grandes categorías:

- *Lineal*: Aquellos modelos cuya función matemática ( $y$ ) está compuesta por constantes ( $\beta_0$ ) y parámetros ( $x$ ) multiplicados por variables independientes ( $\beta$ ). Algunos ejemplos son: LASSO, Elastic Net, Folded Concave o regresión lineal de Naïve-Bayes [92].
- *No lineal*: Aquellos modelos cuya función no puede ser expresada de la forma anterior. Los ejemplos principales son SVR (*Support Vector Regression*), de ventanas de valores medios deslizantes (*moving averages* - MA), modelos de regresión adaptativa o de regresión logística.
- *Híbrida*: Modelos que combinan dos o más modelos lineales y no lineales. Se trata de un tipo utilizado con menor frecuencia (véase un ejemplo en el que se combinan modelos LSTM con Bayes [102]).

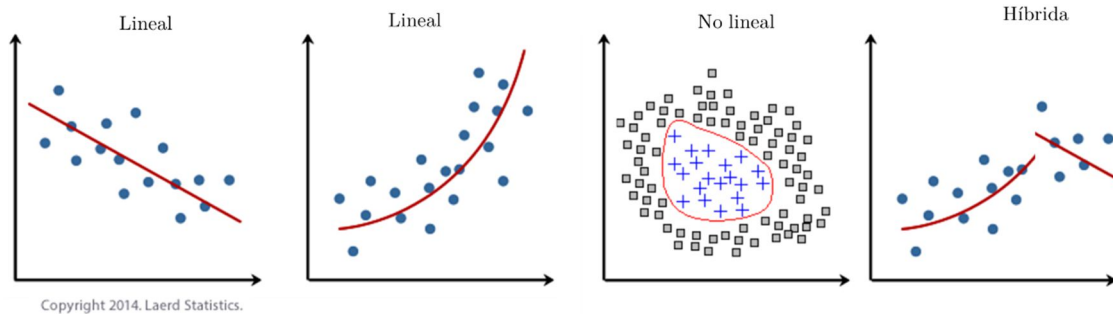


Figura 2.24: Tipos de modelos de regresión. Fuente: [552]

No obstante, existen modelos utilizados también en la literatura (y que gozan de un reciente incremento de atención) como los árboles de decisión (*Regression Trees*, *Random Forest* o *Boosting Regression Trees*) y, (especialmente) lo que se conoce como aprendizaje profundo (*Deep Learning*), en algunas de sus variantes como perceptrón multicapa (MLP – *Multi Layer Perceptron*), redes convolucionales (CNN – *Convolutional Neural Networks*) o redes recurrentes (RNN – *Recurrent Neural Networks*).

En recapitulación, de cara a su inclusión dentro del sistema, debe escogerse un modelo cuyo objetivo sea predecir el valor de una serie temporal de indicadores (continuos) de los que se conoce los valores anteriores (etiquetados). Para ello, el modelo a elegir debe ser capaz de estimar los valores futuros a partir de un entrenamiento y una validación [103].



**Figura 2.25:** Perspectiva de predicción en series temporales

La elección del modelo de ML más apropiado, así como su construcción y uso suele abordarse de una manera sistemática, en la que se exploran los datos, se analizan y se toman decisiones de elección de modelo en función del negocio y del problema particular planteado [104]. Por ejemplo, debe analizarse la estacionalidad de los datos (test de Dickey-Fuller [105], la autocorrelación, etc.). No obstante, para el objetivo de este trabajo (insertar un servicio de predicción de índices aplicable a entornos reales), estas actividades no pueden llevarse a cabo (modelo ejecutado como un servicio en tiempo real sobre una arquitectura de IoT). De este modo, se debe plantear un modelo de propósito general que pudiera ser entrenado y generar predicciones sin depender de análisis previos de la información de dichos contextos.

Es por ello que, previo a la realización de un estado del arte de los modelos más apropiados y utilizados, conviene contextualizar el entorno de los datos y de los resultados esperados por parte de la solución de esta tesis doctoral.

Para ello, se decidió emplear la taxonomía propuesta por J. Brownlee [106], como se ve reflejado en la siguiente tabla:

**Tabla 2.3:** Factores para el análisis de modelos de ML en el estado del arte

|   |                    |
|---|--------------------|
| <b>Entradas vs. salidas</b>   |                    |
| Los datos de entrada serán indicadores ( <i>cKPIs</i> ), que serán el resultado de una composición de datos de una o más fuentes a un modelo de datos concreto. La cantidad y el tipo de datos dependerá de cada caso de uso (diversos), no obstante se asume una serie de condiciones.   |                    |
| <b>Datos endógenos vs. exógenos</b>   | Exógenos           |
| Partiendo de que los fenómenos reales modelados por los CIs en los despliegues de la solución serán complejos, puede entenderse que, en casos generales, la correlación entre las fuentes de datos será baja. Esto se ha validado en uno de los escenarios, como se expone más adelante (sección 5). Además, se plantea realizar una predicción por indicador individual. La salida debe ser el valor de dicho indicador. |                    |
| <b>Datos estructurados vs. No estructurados</b>   | Semi estructurados |
| Se trata de datos de pueden ser representados (tras algo de procesamiento) en formato de tabla (serán datos JSON con campos determinados – véase sección 4.2.1) [93]. Además, se espera que las variables presenten altos grados de estacionalidad y tendencias (validado en la sección 5.3).   |                    |
| <b>Regresión o clasificación</b>  | Regresión          |
| Como se ha descrito previamente, se busca la predicción de valores continuos de indicadores, por tanto, se trata de un problema de regresión.   |                    |
| <b>Univariante o multivariante</b>  | Univariante        |
| Se analiza el valor del indicador, sin considerar otro tipo de variables como el tipo de dato, unidades de medida, etc. [107]. Otros trabajos como [108] se centran en el estado del arte de las series temporales multivariantes, lo que supone un mayor reto que el anterior.   |                    |
| <b>Un paso o multi-paso</b>   | Multi.paso         |
| Se busca dar la opción de que el horizonte de predicción de los indicadores se variable, por tanto debe considerarse multi-paso (granularidad mayor a la del periodo de datos de entrada).  |                    |
| <b>Modelo estático o dinámico</b>   | Mixto (estático)   |
| Un modelo estático es aquel que se entrena una única vez y se utiliza para el resto de entradas futuras, mientras que el dinámico debe actualizarse con cada nuevos valores. Como se expresa más adelante (véase sección 4.2.3), se desea permitir el reentrenamiento pero bajo petición del usuario, por tanto es un contexto mixto, con preferencia a modelos estáticos.  |                    |
| <b>Uniformidad de las observaciones</b>   | Contiguas          |
| Los datos tendrán una frecuencia constante (proporcionada por el agente).   |                    |



Por tanto, analizando las asunciones globales anteriores, se procede a describir aquellos modelos más utilizados y oportunos para un problema de regresión para la predicción de valores de series temporales univariantes, con datos semi estructurados y horizonte de predicción variable.

### **Regresión lineal**

En primer lugar, los modelos de regresión lineal se descartaron para su utilización debido a sus limitaciones. Únicamente contemplan relaciones lineales de los datos con las predicciones, sin poder incorporar aspectos como la estacionalidad, así como son sensibles a valores atípicos [109] [110].

### **SVR – Support Vector Regression**

Este modelo se basa en los principios de SVM (orientado a clasificación) y busca encontrar el *hiperplano* que más se ajuste a todos los puntos del *set* de datos, para utilizarlo como referencia en la predicción. Se trata de un modelo que, a pesar de aceptar bien la generalización, resultó descartado ya puede llegar a ser computacionalmente costoso (por ejemplo, el SVR Bayesiano [mariette]) y al no ser factible para grandes volúmenes de datos (lo que puede suceder en entornos de IoT de adquisición en tiempo real) [111].

### **Modelos de media móvil o ventana deslizante**

El siguiente grupo de modelos son los conocidos como de ventana deslizante (MA, del inglés *Moving Average*). Establecen ventanas de selección de la media de valores anteriores para predecir el valor en un momento del tiempo. Estos pueden ser una simple selección de la ventana del periodo de valores (SMA – *Simple MA*) o una ventana ponderada (por ejemplo, WMA, - *Weighted MA*, que da más peso a los valores más recientes con factor multiplicativo o EMA – *Exponential MA*, que hace lo propio con factor exponencial). No obstante, estos métodos no son utilizados ya que se trata de modelos altamente subjetivos (ponderación, selección de ventana, etc.).

### **ARIMA (*Auto Regression Integrated Moving Average*)**

También conocido como la metodología Box-Jenkins, es el primer modelo relevante para considerar fue ARIMA, que parte de términos de autorregresión, de diferencias estacionarias y de los errores de predicción incurridos en caso de utilizarse. Aunque puede encajarse dentro de la familia anterior, este modelo requirió de un análisis particularizado ya que es uno de los algoritmos más maduros y consolidados para la regresión y previsión de series temporales, que se encuentra implementado en una gran cantidad de lenguajes y embebido en muchas herramientas software [100]. De hecho, ARIMA es el modelo utilizado para benchmarking en la mayoría de los trabajos que proponen algún método de previsión de series temporales univariantes [112].

Se basa en la utilización de ventanas deslizantes y en la selección de parámetros de número de integraciones ( $d$ ), el orden de autorregresión ( $p$ ) y el orden de retardo o de media móvil ( $q$ ), así como en coeficientes de regresión. En su



utilización, adopta la notación  $ARIMA(p,d,q)$ . Es recomendado para series temporales tanto estacionarias como no estacionarias.

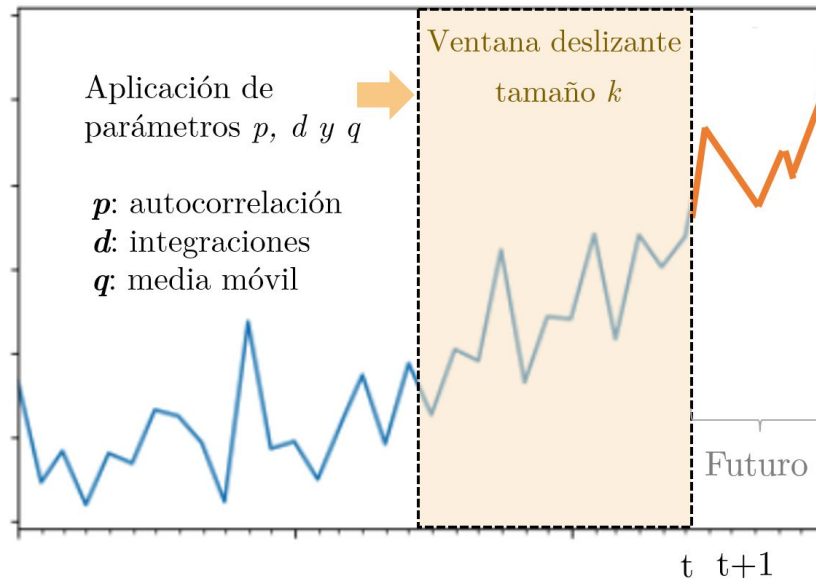


Figura 2.26: Modelo ARIMA

Cabe destacar que el modelo ARIMA es, hoy en día, referencia en el ámbito de la previsión de series temporales en entornos IoT [113]. El principal inconveniente de estos métodos es que son propensos a cometer errores cuando hay muchas variaciones en los patrones, lo que es especialmente común en las previsiones a corto plazo.

### **SARIMAX (Stationary Auto Regression Integrated Moving Average with exogenous data)**

Existe una serie de variaciones del modelo anterior (ARMA, SARMA, ARIMAX, SARIMA), pero para el análisis decidió escogerse el que más se ajustaba a las condiciones expuestas en la tabla Tabla 2.3. SARIMAX [114] es la extensión de ARIMA específica para series temporales estacionarias y con variables exógenas. Se trata de un método muy utilizado en aplicaciones IoT [115], aunque posee algunas desventajas como ser muy sensible a valores ausentes y que requieren una gran cantidad de datos para comenzar a ser utilizados (especialmente relacionado con la estacionalidad). Finalmente, se trata de un modelo computacionalmente costoso, sobre todo si los valores de  $p$  y  $q$  alcanzan valores grandes [116].

### **Holt-Winter Exponential Smoothing**

Ya fuera de la familia de modelos de ventana deslizante, el modelo *Holt-Winter Exponential Smoothing* [117] es un método para la predicción univariante especializado en series con tendencias marcadas y con componentes estacionales. Es un modelo de regresión no lineal que utiliza una fórmula matemática con parámetros de entrada: el tipo de tendencia y la longitud de la estacionalidad.

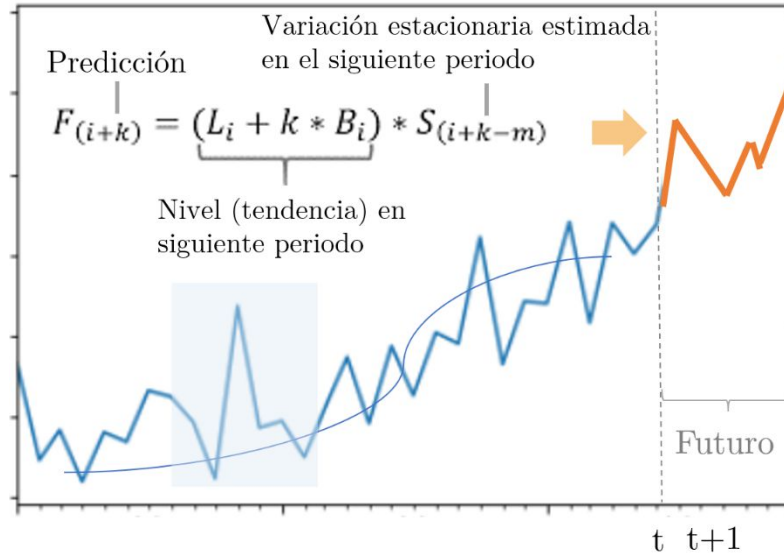


Figura 2.27: Modelo Holt-Winter Exponential Smoothing

### Facebook Prophet

Facebook Prophet es otro de los métodos más utilizados en entornos IoT [118]. Se trata de un *framework* open source desarrollado por Facebook, que utiliza un método de ajuste de curvas basado en el método bayesiano de predicción de series temporales. Puede clasificarse como un modelo de regresión aditivo en el que se ajustan las tendencias no lineales con estacionalidad anual, semanal y diaria, además de los efectos de las vacaciones. Funciona mejor con series temporales que tienen fuertes efectos estacionalidades (véase tabla Tabla 2.3) y con la capacidad de dotar de un buen histórico que vaya en aumento [119]. Además, supera los inconvenientes de otros modelos de este listado al tener mejor capacidad de absorción de datos ausentes y tratar proveer de predicciones multi-paso (de nuevo, véase tabla anterior) de una manera fiable.

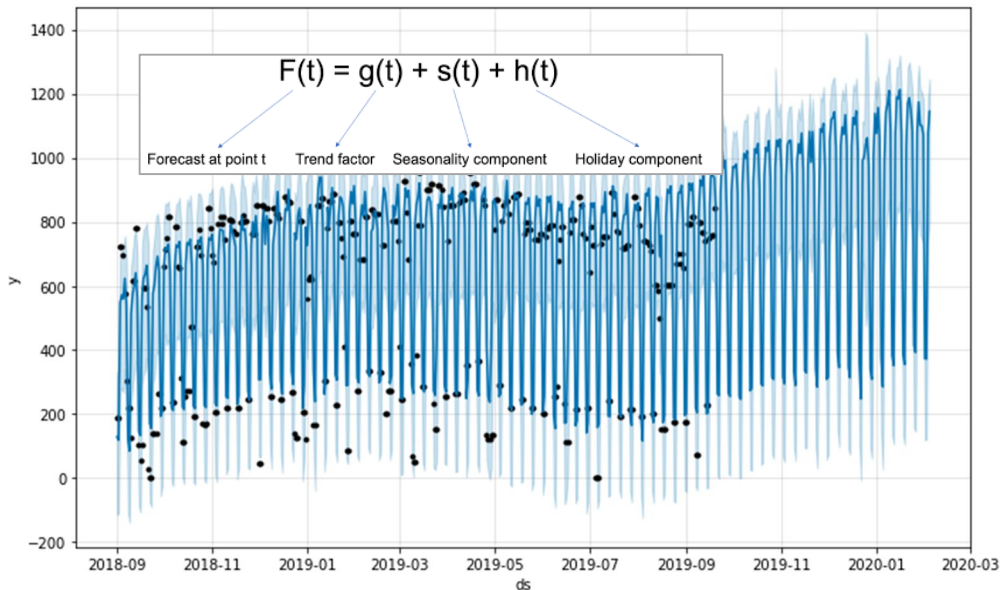
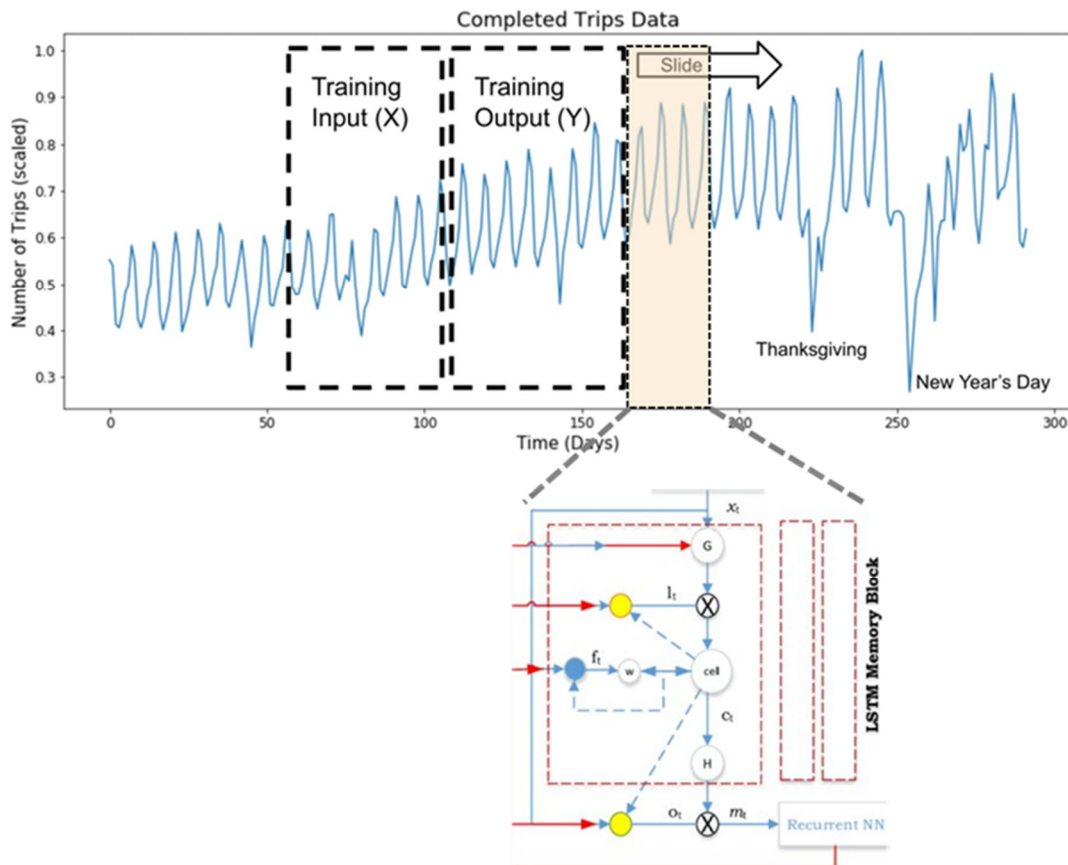


Figura 2.28: Modelo Facebook Prophet

### LSTM (Long Short-Term Memory)

Este es el método más destacado (y el único analizado en detalle en esta lista) que utiliza redes neuronales para la predicción de series temporales [120], que en la actualidad son un campo muy explorado ya que permiten aprender automáticamente la dependencia temporal en estructuras con estacionalidad y tendencias. LSTM se basa en construir red neuronal recurrente (RNN) que incluye el manejo explícito del orden entre las observaciones a la hora de mapear las entradas a las salidas. Este tipo de redes se benefician del algoritmo de conexiones *feedback* para ajustar la ponderación de las redes temporales, pudiendo “retener memoria” de predicciones anteriores y utilizándola para mejorar la precisión posterior, aprendiendo por tanto patrones más complejos. Este aspecto es muy relevante ya que supera otras redes como los perceptrones multicapa (MLP) o las convolucionales (CNN) [121].



**Figura 2.29:** Modelo LSTM. Fuentes: [553][554]

Operativamente, utiliza el concepto de Ventana deslizante para la determinación del *set* de entrenamiento y validación, que incluyen predicciones anteriores. Los datos de entrenamiento se normalizan entre 0 y 1, el método de optimización del error es Adam [122] y el número de neuronas de la RNN puede ser cambiado.

Existe un número considerable de ejemplos satisfactorios en la literatura que utilizan LSTM como modelo predictivo [112], [123]. No obstante, este tipo de modelos presentan varios inconvenientes destacables, como grandes requerimientos de entrenamiento (computacionales y de tiempo), incurrir en sobreajustes y son altamente sensible a cómo se introducen los parámetros para el comienzo del entrenamiento.

Se ha propuesto asimismo el uso de otros modelos, como SRM (*Sparse Representation Model*) en casos como Smart Cities [124], *Short Term Load Forecasting* (STLF) en energía [125], *Deep Belief Networks* (DBN) en predicción meteorológica [126] TBATS para predicción de enfermedades [127], entre muchos otros. Las técnicas más exploradas, hoy en día, para futuras aproximaciones prácticas son las Redes Neuronales y Bayesianas, las técnicas Difusas (*fuzzy*) y las Evolutivas, aplicadas a entornos de aplicación específicos con diferentes resultados de precisión [128].

### *Consideraciones particulares y desafíos*

Como se ha ido mencionado en páginas anteriores, se han encontrado muchos casos de aplicación de estos modelos en diferentes campos (tales como contaminación atmosférica, la manufactura de semiconductores, operadores de telecomunicaciones, Smart Cities, predicción de tráfico, energía, turismo entre muchos otros). No obstante, de cara a la materialización de un modelo dentro de la solución de la tesis se debió pensar en un caso general. La arquitectura diseñada está preparada para ser desplegada en cualquier dominio de aplicación, pudiendo capturar indicadores provenientes de diferentes fuentes muy dispares (desde temperatura y niveles de CO<sub>2</sub> hasta valores económicos, caudal de agua, energía consumida, ruido o densidad de gente en un espacio determinado). Por tanto, las asunciones y conclusiones de dichos trabajos podrían no resultar válidas para un caso general.

Además, la utilización de estos modelos en entornos de recogida de datos en tiempo real (y no sobre volúmenes de datos fijos o conocidos) supone una serie de desafíos adicionales. Los datos de series temporales a menudo tienen un volumen muy elevado, especialmente en escenarios de IoT. Almacenar, indexar, consultar, analizar y visualizar datos de series temporales de cara a una predicción univariante puede resultar una tarea compleja, que no puede abordarse caso por caso en los escenarios planteados (es decir, indicador por indicador). A nivel práctico, cada caso (indicador) puede variar y encontrar la combinación correcta entre capacidades técnicas y ajuste a un escenario, al tiempo que se minimiza el tiempo de despliegue e integración con una arquitectura IoT, es un aspecto que no se contempla en la literatura (centrada en soluciones particulares). Puede resultar complicado encontrar la combinación correcta entre almacenamiento de alta velocidad y operaciones de proceso eficaz para controlar el análisis en tiempo real, al tiempo

que se minimiza el tiempo necesario para efectuar la predicción y una calidad de resultado satisfactoria.

Por otro lado, uno de los principales problemas que se encuentran en las aplicaciones reales para realizar predicciones precisas puede dividirse en dos aspectos clave: (i) la falta de datos en tiempo real (la recogida de datos en puntos únicos y la disponibilidad de amplios conjuntos de datos en escenarios reales es compleja) necesarios para establecer buenos modelos y entornos de ejecución y (ii) la imprevisibilidad de los fenómenos que influyen en potenciales aspectos a modelar en el sistema propuesto (por ejemplo, impacto medioambiental, tráfico, densidad de gente) en un momento determinado (por ejemplo, vertidos no intencionados, choques u otros accidentes). Estos se suman a otros obstáculos que hacen que el campo esté abierto a la investigación, especialmente ahora (y en el futuro) con una tecnología de computación más potente y sofisticada que aúne las líneas actuales expuestas en este capítulo junto con las secciones 2.2 y 2.3.

**Conclusión tras el análisis e influencia en la tesis doctoral:**

El objetivo es plantear la automatización de modelos (y su entrenamiento) dentro de una arquitectura IoT. Por tanto, las expectativas de esta tesis residen en replicar con éxito algunos de los modelos y métodos ya validados de la literatura y aplicarlos dentro de un marco tecnológico a casos de uso reales de indicadores compuestos. El objetivo no es avanzar en el estado del arte en este campo, sino encontrar una forma de fusionar estas tecnologías con un marco de trabajo en tiempo real basado en los conceptos del IoT.

En este capítulo se ha introducido el uso de ML en entornos de IoT y se ha contextualizado para los problemas que se abordarán en los despliegues de esta tesis. En particular, se ha establecido que deben considerarse modelos orientados a resolver problemas de regresión para la predicción de valores de series temporales univariantes, con datos semi estructurados y horizonte de predicción variable.

Dentro de los mismos, se han analizado los principales modelos en la literatura actual, donde SARIMAX y Prophet destacan por su capacidad para ajustarse a los requerimientos (analizados en una etapa inicial). Los modelos de media móvil (ARIMA, etc.) requieren mayor ajuste manual de parámetros, aunque requieren en general menor capacidad de computación. Prophet aporta proporciona parámetros intuitivos y una configuración que no requiere demasiada intervención manual, ideal para entornos genéricos con datos que contienen estacionalidad (o no), tendencias y datos ausentes.

## 2.4.2. Aprendizaje federado (*Federated Learning*)

A medida que los métodos de aprendizaje automático fueron siendo explorados y mejorados, un mayor número de casos de uso con crecientes requerimientos de precisión y volumen de datos fueron apareciendo. Este hecho llevó a abordar el problema de que, aun con modelos ML avanzados, completar esa tarea en un solo equipo/procesador/nodo requeriría una cantidad de tiempo considerable (horas, días o incluso semanas). En este sentido, existen multitud de trabajos que han explorado maneras de “dividir” la carga computacional en diversos nodos (asumiendo que se dispone de un equipo de computación paralela, o varias estaciones o un *cluster* de equipos) y de cómo combinar sus resultados para completar la tarea original. Esto es lo que se conoce como aprendizaje distribuido (DL – del inglés *Distributed Learning*) [32]. La literatura, soluciones, algoritmos y herramientas en este campo es amplia y queda fuera del alcance de esta tesis doctoral.

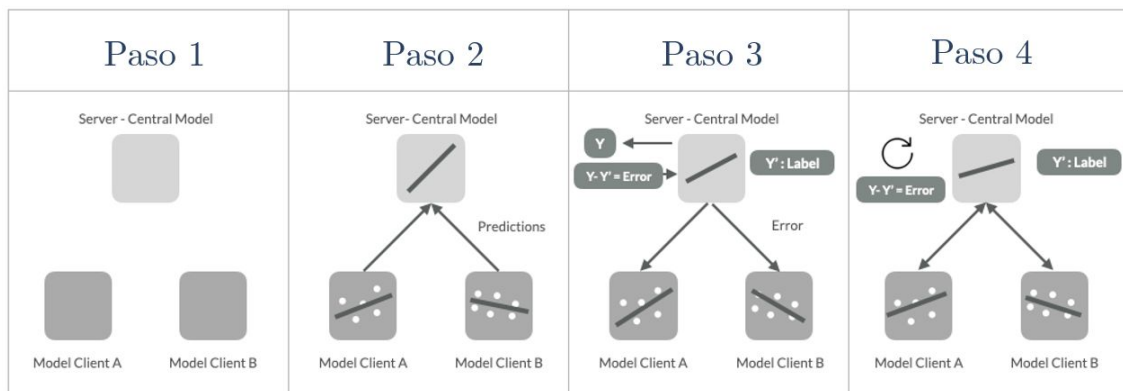
Sin embargo, una de sus ramas resulta de interés: aquella que trata de cubrir un aspecto olvidado por el DL, aquellos escenarios en los que no haya un único propietario de los datos y/o éstos se encuentren distribuidos en nodos geográfica y lógicamente dispersos.

Este tipo de escenarios son cada vez más habituales debido a la proliferación de redes inalámbricas con distintos anchos de banda y alcances (reales), al aumento de la capacidad de comunicación de elementos dispersos y en propiedad de múltiples entidades (un claro ejemplo son los *smartphones*) y a los avances en la investigación, desarrollo y despliegue de los ecosistemas IoT. Cabe aquí mencionar que una de las innovaciones más relevantes que hacen patente esta necesidad es la transición de centros de datos centralizados a entornos *edge* y *far edge*. Los datos generados en el *edge* (véase sección 2.3) están aumentando a una escala sin precedentes. La tendencia hacia *edge analytics* está cada vez más impulsada por la necesidad de tomar decisiones en tiempo real, la latencia constante y los requisitos de privacidad, frente a los objetivos iniciales de disminuir el coste de la red [129].

La gran afluencia de datos en el *edge* representa el ingrediente ideal para abordar dichos escenarios en los que elementos de diversos propietarios y en múltiples ubicaciones pueden participar en el entrenamiento e inferencia de modelos de IA/ML, aprovechando las capacidades y los datos locales. La intersección es natural: el *edge computing* ofrece las características y el caldo de cultivo para poder efectuar un aprendizaje, que solo tendrá lugar con los métodos, algoritmos y modelos ya explorados en el campo del ML. La unión práctica de ambos campos (combinación, división) constituye el repaso de esta subsección.

En este sentido, surgió un nuevo concepto: El **aprendizaje federado** (FL, del inglés *Federated Learning*), acuñado por Google en 2017 [130]. FL es un enfoque del aprendizaje automático, en el que el entrenamiento de un modelo implica

múltiples conjuntos de datos almacenados en “nodos locales”, mientras que el entrenamiento en sí procede sin intercambiar ningún dato entre ellos ni con el elemento central. En síntesis, el objetivo es entrenar un modelo único que pueda beneficiarse del aprendizaje de cada nodo y que, además, mejore la precisión de tal manera que pueda ser aprovechada por todos ellos. El flujo consiste en que cada nodo local utiliza únicamente los datos en su entorno (a los que tiene acceso) para entrenar su subversión local del modelo. A continuación, los parámetros del modelo se “combinan en el modelo”, orquestados por un elemento central. Una vez completada la actualización, el modelo actualizado se redistribuye a los nodos que han participado en el entrenamiento. En este caso, las inferencias se producen en los nodos. Finalmente, el bucle se cierra, y el proceso se repite, hasta que el modelo común se considera de “calidad suficiente” (utilizando criterios específicos acordados). En resumen, FL lleva el código desde la nube a los datos, en lugar de lo contrario, y garantiza que los datos reales permanezcan en los dispositivos y sólo se compartan globalmente las actualizaciones de los modelos, lo que garantiza un enfoque de aprendizaje cooperativo y descentralizado, al tiempo que mejora la privacidad y la seguridad [131].



*Paso 1* - Cada nodo *edge* diseña un modelo (o adquiere el enviado por la central), que aplica sobre sus datos locales.

*Paso 2* - El modelo es reentrenado con datos locales y se envía al servidor central.

*Paso 3* - El servidor central concatena los modelos recibidos como parámetros de entrada, calcula el error, combina y redistribuye.

*Paso 4* - El proceso continúa hasta que el error es minimizado para todos los modelos, por tanto el modelo central único será válido para los mismos.

**Figura 2.30:** Resumen del flujo de aprendizaje federado. Fuente: [555]

Las fuentes indican que, a pesar de ser muy reciente (apenas 5 años), este campo de investigación ya está despertando gran interés dentro de la comunidad científica e industrial. Según el publicado *roadmap* de AIOTI [132], la próxima versión (v6) de su HLA (véase sección 2.2) se centrará en la Inteligencia Artificial en entornos IoT descentralizados, donde el aprendizaje federado está llamado a

desempeñar un papel importante. Además, la potenciación de los dispositivos *edge* existentes y nuevos con IA/ML para extraer conocimientos valiosos representará una de las principales tendencias tecnológicas en el futuro próximo, según la misma fuente. Es interesante mencionar (al hilo de lo comentado sobre los *smartphones*) que uno de los sectores que más interés ha expresado por estudiar este nuevo paradigma es el de las operadoras de telecomunicaciones, entendiendo como nodos cada *smartphone* conectado a sus redes, y en el que podrían delegar parte del proceso de ML aprovechando las capacidades de computación y utilizando datos del usuario garantizándole que nunca serían analizados ni compartidos [133].

No obstante, un trabajo realizado en 2021 constata que se trata de un campo muy “inmaduro” en cuanto a validación en implementaciones a gran escala; los experimentos se han centrado, por ahora, principalmente, en simulaciones [32] [134].

Esto se debe en gran parte a que la integración de dispositivos *edge* con la creciente demanda de recursos y la complejidad de las aplicaciones de IA según el paradigma FL plantea numerosos desafíos técnicos, algunos de los cuales se exponen a continuación:

- Existe la limitación de que, al contrario que en el *cloud* o dispositivos de computación pueden no estar siempre preparados o disponer de capacidad de hardware para explotar los modelos ML distribuidos con la calidad de servicio esperada (en particular para modelos extensos, como las redes neuronales convolucionales). La coordinación y orquestación de recursos se ha estudiado ampliamente en entornos centralizados, homogéneos y con capacidades de computación suficientes, y más recientemente están surgiendo iniciativas para gestionar recursos en el continuo *IoT-edge-cloud*, considerando el acceso aleatorio a las redes, las limitaciones de energía y los requisitos (no) funcionales de las aplicaciones [135].
- Mayores costes de comunicación, ya que los datos y nodos se encuentran distribuidos y requieren de acciones de comunicación y un riesgo de desbalanceo de datos en términos de disponibilidad, completitud y precisión [136]. Para solventar estos problemas, se han propuesto técnicas como delegar mayor parte del entrenamiento en los nodos *edge*, distribuir modelos de mayor calidad o comprimir los mensajes con diferentes esquemas de modulación para disminuir la sobrecarga en la comunicación [137], [138].
- El carácter potencialmente sensible o confidencial de los datos plantea problemas de privacidad cuando se gestionan, almacenan y procesan en lugares centralizados. Al mismo tiempo, la capacidad de la infraestructura de la red corre el riesgo de verse saturada por esa recogida continua de datos. Aproximaciones como algoritmos más eficientes y la introducción de *blockchain* como elemento complementario al FL han sido estudiadas [139].



Para implementar este paradigma fue necesaria la definición de algoritmos de control del flujo mencionado y, sobre todo, de la combinación de los modelos recibidos por parte de los nodos para la construcción del modelo central único a distribuir. La principal referencia en este aspecto es FederatedSGD [140] (FedSGD), publicado en 2016 y que consiste en un algoritmo de Descenso Gradiente Estocástico (SGD – *Stochastic Gradient Descent*) que ha sido probado sobre diferentes arquitecturas ML como redes convolucionales, redes LSTM, perceptrón multicapa, árboles de decisión, modelos de regresión y otros. Aunque la mayoría de los trabajos publicados actualmente se centran en el uso de FL basados en redes neuronales, esto no es necesario. Algo más tarde, FedAVG [141] mejora FedSGD ya que propone un SGD en local y una combinación de modelos (media) en el elemento servidor central. Otros trabajos proponen cambiar el SGD local (muy costoso y poco preciso) por un algoritmo Primal-Dual (FedPD) [142].

Estos algoritmos constituyen el grueso de la parte teórica sobre como combinar los modelos para implementar soluciones de FL. En el aspecto práctico, los trabajos encontrados [143] hacen uso de herramientas de inteligencia distribuida (TensorFlow, MxNet o Caffe), Cloud ML (AWS, Azure, GCP), librerías de ML (pandas, Theano, scikit-learn) y herramientas específicamente diseñadas o con características específicas para el aprendizaje distribuido (Apache Fink, Horovod, MapReduce o Apache Spark).

Se han encontrado propuestas prácticas particulares para diversos sectores basadas en el uso de FL, como FedVision (procesamiento de imágenes) [144], WeBank (un banco digital que utiliza FL para la calificación crediticia) [145] o su utilización para la detección de drogas promovido por firmas farmacéuticas [146].

No obstante, se han encontrado muy pocas referencias de plataformas FL agnósticas de dominio. Y, de entre ellas, es difícil encontrar implementaciones y validaciones prácticas en escenarios similares a los planteados en esta tesis. Destacan [32] FATE, Google TF Federated, OpenMinded PySyft y Baidu PaddleFL.

No obstante, a pesar de la aparición de estas tecnologías y plataformas, existen todavía muchos aspectos del FL sin ser resueltos, lo que hace concluir que se trata de un campo en bajo grado de madurez, todavía en su infancia. En la siguiente lista se muestran algunas de las deficiencias a cubrir durante los próximos años, que constituyen las líneas de investigación actuales en FL:

- Muchos de los algoritmos de cálculo (como SplitNN [147]) presentan tiempos de entrenamiento imprácticos para aplicaciones en tiempo real y con altos requerimientos de comunicación.
- Muchos modelos asumen que los hiperparámetros del modelo central a distribuir ya han sido establecidos y desvían el foco en ajustarlos.
- Mientras que en entornos *cloud* es aceptable realizar muchas iteraciones hasta conseguir un modelo entrenado fiable, esto es impráctico en entornos distribuidos a gran escala, teniendo que llegar (hoy en día) a

compromisos que afectan a la calidad de la predicción.

- En las aplicaciones actuales, se asume que los elementos *edge* son homogéneos en sus características, requiriendo que realicen tareas sobre modelos sin considerar las características propias de cada uno (arquitecturas, dependencias, almacenamiento, RAM...).
- El planteamiento actual se basa en que un único modelo (distribuido por el elemento central) es usado para la inferencia (y es reentrenado) en cada nodo, sin considerar que puede que algún nodo tuviese un mejor rendimiento con versiones anteriores. No se contempla la convivencia de varias versiones de los modelos dentro del ecosistema.
- No existe una unidad estándar para el empaquetado de los modelos ni una referencia formal para el componente de registro de los modelos.
- La falta de estándares se hace patente también en cuanto a los modelos de interoperabilidad sobre los datos, el acceso a archivos del sistema, la identidad de los nodos, etc.
- El consumo de energía requerido para que elementos limitados en computación desempeñen las tareas de inferencia y entrenamiento requeridas es alto. Se están explorando estrategias para desplegar únicamente inferencias parciales de modelos ML, explotando teorías de computación paralela o aproximada. Relacionado con este aspecto, cómo integrar los mecanismos de aprendizaje para implementar la toma de decisiones adaptativa, es un gran reto de investigación, que se traduce directamente en ganancias de eficiencia energética y de rendimiento para los sistemas de borde distribuidos.

**Conclusión tras el análisis e influencia en la tesis doctoral:**

El aprendizaje federado parece ser una gran oportunidad para despliegues como el planteado en esta tesis ya que aborda las deficiencias del aprendizaje distribuido en el caso en que se disponga de entornos geográfica y lógicamente dispersos y con capacidad de computación singular. No obstante, la revisión en la literatura ha revelado que se trata de un campo en un estado de muy baja madurez, existiendo una gran cantidad de problemas sin resolver. Algunas soluciones y plataformas han aportado conclusiones interesantes, y se han propuesto algoritmos para la agregación de modelos entrenados por los diferentes nodos de tal manera que puedan combinarse en un entorno central y ser de nuevo distribuidos.

## 2.5. Otras tecnologías habilitadoras

### 2.5.1. Virtualización y escalabilidad

NGIoT incluirá cada vez más dispositivos y nodos de computación en todo el continuo (véase sección 2.2). Tal y como se ha puesto de manifiesto en la última publicación de AIOTI HLA [148], este hecho viene acompañado de importantes necesidades de escalabilidad. La escalabilidad en el IoT es una tendencia que pronto se convertirá en un habilitador clave para este tipo de despliegues. Esta escalabilidad se antoja esencial para adaptar las cargas de trabajo a los diferentes escenarios, al rendimiento, a los costes y a otras necesidades del negocio, preservando el rendimiento medio y la calidad de servicio a medida que aumenta el número de elementos (peticiones, nodos, servicios, etc.). Este concepto es de clara aplicación al ámbito de este doctorado, ya que deben poder añadirse dispositivos (y otras fuentes de datos) para alimentar el indicador compuesto. Además, como complemento a las reflexiones sobre el *edge computing* realizadas en la sección 2.3, la escalabilidad y flexibilidad son características de suma importancia.

Según otra entidad de referencia que ya se ha mencionado en secciones anteriores, OpenFog [81], la escalabilidad en IoT puede atacarse desde tres dimensiones: hardware, software y capacidades de comunicación:

**Tabla 2.4:** Diferentes dimensiones de escalabilidad

|   |  |
|---|--|
| <b>Hardware escalable:</b>  | Adición de GPU, múltiples interfaces...    |
| Capacidad de modificar la configuración de elementos físicos (ya sean sensores, actuadores, o nodos <i>fog</i> ) de un despliegue NGIoT, así como de añadir nuevo hardware que va desde PLCs de un solo núcleo de CPU hasta GPUs especializadas como nodos <i>fog</i> o incorporar nuevas interfaces. En cuanto a la gestión de estas características, la escalabilidad es un objetivo bastante complejo ya que los ecosistemas IoT están compuestos por nodos heterogéneos de múltiples proveedores [149].   |  |
| <b>Software escalable:</b>  | Virtualización y microservicios            |
| Capacidad de replicación de servicios software sobre una infraestructura así como de gestionar ambos aspectos sobre un continuo de computación escalable. Debe permitir la partición, el equilibrio y la asignación de software y recursos para que funcionen sobre dicho continuo. Para ello, la componibilidad y la modularidad son aspectos clave de la escalabilidad del software, donde los componentes individuales de hardware y software se ensamblan en un nodo NGIoT optimizado para ejecutar las aplicaciones específicas requeridas (por ejemplo, microservicios que se ejecutan sobre contenedores básicos). |  |
| <b>Comunicación escalable:</b>  | Configuración dinámica y virtual de la red |
| Implica la capacidad de modificar la configuración de los elementos de red, incluida la gestión de las redes IP (y no IP), así como la softwarización (SDN) y la virtualización (NFV) de la red en los despliegues modernos de IoT [150]  |  |

La primera dimensión (hardware) se encuentra fuera del alcance de la tesis, por tanto, no se profundiza en la revisión de sus posibilidades. En cuanto a la segunda y la tercera, ambas se antojaron interesantes en tanto en cuanto radican en la virtualización.

La virtualización es un mecanismo que puede aplicarse a todas las diferentes capas y funciones de un despliegue de IoT, desde la flexibilidad de la red o la planificación de la capacidad hasta el almacenamiento y el procesamiento de datos. Según la propia definición de AIOTI, la virtualización en el IoT se asienta en dos pilares, a los que se podría añadir un tercero: (1) los principios *cloud-native*, (2) la evolución de la red hacia la flexibilidad y (3) la virtualización de dispositivos IoT para la reutilización en aplicaciones concurrentes. Es conveniente señalar que en este doctorado sólo se aplica el primero (*cloud-native principles*). La segunda característica coincide con la segunda dimensión clasificada por OpenFog y también está fuera del alcance de este doctorado.

La aplicación de la virtualización a la arquitectura del IoT (relevante para este caso) debe entenderse como el cumplimiento de los siguientes principios *cloud-native*: microservicios, capacidad de crecimiento en datos, dispositivos y puntos de computación y el uso de DevOps. Mientras que este último se trata en otra subsección (2.5.3) y los del medio son subproductos del primero, esta subsección estudia la virtualización y la escalabilidad a través de los microservicios.

Aquí, se entiende como microservicio: un enfoque de arquitectura para desplegar aplicaciones como un conjunto de pequeños servicios, ejecutados como procesos separados, comunicándose a través de mecanismos simples.

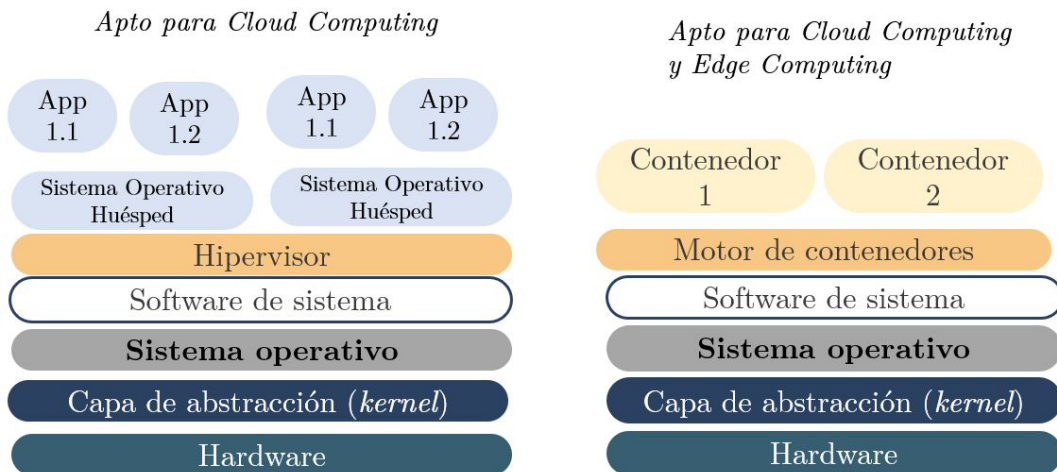
Dentro del entorno IoT, esto significa dividir las funcionalidades de una arquitectura (tal y como se concebían en las aproximaciones monolíticas) como distintas aplicaciones que pueden desplegarse e integrarse tras su instanciación.

Este modelo presenta muchas ventajas, como la capacidad de actualización y mantenimiento de cada aplicación (o módulo por separado), así como una mejor depuración y corrección de errores. Además, por diseño, este paradigma permite una mayor escalabilidad. Se otorga también una gran flexibilidad a la hora de seleccionar dónde (en términos de nodo de computación) debe ejecutarse cada aplicación. En este sentido, dentro de esta tesis doctoral, se ha creado una arquitectura modular que permita despliegues *edge computing* o híbridos, alejándose de la tradicional centralización de los sistemas de IoT clásicos. Esto facilitará, a su vez, la utilización de la misma arquitectura (véase sección 3) para escenarios modestos (apenas ejecución local) hasta grandes despliegues con equipamiento heterogéneo e incluso *cloud*. En relación a dicho despliegue, la utilización de microservicios, donde las operaciones son sencillas y estandarizadas, facilita la automatización de las mismas (DevOps).

Para el caso de esta tesis, los microservicios permitirán (más allá del despliegue independiente de servicios), proporcionar también una delimitación clara de cada componente, permitiendo que diferentes módulos de la arquitectura IoT

sean desarrollados en diferentes lenguajes de programación o haciendo uso de diversas herramientas de código abierto. Por tanto, los módulos y servicios a ejecutar serán microservicios, cada uno responsable de ejecutar una función.

A nivel práctico, en entornos de *cloud computing*, la virtualización se realiza a través de máquinas virtuales, dotando de un hardware de computación (IaaS) o de funcionalidades (servicios) empaquetadas dentro de los mismos. Sin embargo, este enfoque no es adecuado para el *continuo IoT-edge-cloud*, ya que los dispositivos limitados y los nodos *fog* no pueden permitirse la sobrecarga que añaden las máquinas virtuales. Por lo tanto, la respuesta es: contenedores. Los contenedores permiten que las aplicaciones se desplieguen de forma fiable y se migren rápidamente entre varios entornos (sistemas operativos) empaquetando el código, los ajustes de configuración y las dependencias en un único objeto. Varios contenedores pueden ejecutarse en el mismo *host* haciendo creer a los servicios subyacentes que tienen toda la capacidad de computación disponible (de forma similar a lo que hacen las máquinas virtuales). Los contenedores son, pues, ideales para dividir sistemas complejos en microservicios.



**Figura 2.31:** Comparación de opciones de virtualización

En cuanto a la implementación de contenedores, la tecnología más usada con diferencia es Docker. Docker ofrece varias características: multiplataforma, versionado, reutilización, bibliotecas compartidas y repositorio. Hay una gran comunidad sobre Docker, y existen muchas imágenes disponibles en DockerHub. Se ha seleccionado esta tecnología dentro de los principios de diseño de la arquitectura de esta tesis (véase sección 3.2.1). Docker puede considerarse como una evolución de LXC (Linux Containers), también conocido como microcontenedores y también vinculado a LXD (gestor de LXC). Aunque el enfoque de LXC es similar y su carga de recursos puede ser bastante reducida, sólo está diseñado para entornos Linux y no tiene una amplia comunidad detrás en comparación con Docker.

Reflexionando sobre la escalabilidad, existe en el estado del arte una serie de tecnologías que permiten trabajar sobre dichos contenedores en términos de automatización de despliegue, gestión de ciclo de vida, corrección de errores,

replicabilidad, instanciación en múltiples nodos formando redes o clústeres, etc. Kubernetes [151] [200], Docker Swarm [152] [201], Apache Mesos [153] [203], Amazon ECS o EPOS Fog son ejemplos de tecnologías que sirven para estos fines, siendo Kubernetes la más adoptada con diferencia.

Kubernetes (también denominado K8s) automatiza los despliegues y el versionado y supervisa el estado de los microservicios de software para evitar los malos despliegues antes de que existan caídas de servicio. También permite ejecutar continuamente comprobaciones de estado de los servicios desplegados, reiniciando los contenedores que fallan. Además, Kubernetes escala automáticamente los servicios en función de su uso, asegurando que sólo se ejecute lo necesario. Requiere unos requisitos de infraestructura relativamente altos (varios *hosts* potentes con mucha memoria), ya que se creó para poder gestionar cientos/miles de nodos. Esta es, principalmente, la razón por la que esta tecnología está muy consolidada en el entorno *cloud* en la nube, pero todavía tiene que adaptarse a la naturaleza heterogénea del continuo *IoT-edge-cloud* [154]. Sin embargo, iniciativas para superar estas limitaciones están ganando la atención de las alianzas NGIoT y las comunidades de código abierto, existiendo varias alternativas como Minikube, K3s [155], microk8s [156], o Akri [157] (que supuestamente permite registrar recursos nativos de k8s en dispositivos de hoja como cámaras IP y dispositivos USB en el borde). No son todavía soluciones maduras.

#### **Conclusión tras el análisis e influencia en la tesis doctoral:**

La virtualización se antoja un elemento clave para la nueva generación de despliegues de IoT al pensar en la consecución de escalabilidad en los mismos. En los entornos de *cloud computing* es común el empleo de este concepto, que se materializa en la provisión de máquinas virtuales como recursos de computación o para proveer una función determinada. No obstante, en entornos hacia donde se orienta la arquitectura de esta tesis (*edge-cloud computing continuum*) esta solución no es aceptable. En este trabajo, los módulos de la arquitectura y servicios a ejecutar se materializarán como contenedores, cada uno responsable de ejecutar una función (en contraposición a la forma monolítica de construir una aplicación que lo ejecute todo), alineando así la solución propuesta con las arquitecturas de microservicios. En un ejercicio adicional de escalabilidad, estos contenedores pueden orquestarse, gestionando su ciclo de vida o replicándolos en agrupaciones de nodos. Para ello, la tecnología principal es Kubernetes que, aunque ideada para la nube, contiene alternativas incipientes para entornos más restrictivos. A pesar de que este paradigma de orquestación parece relevante, se ha decidido no profundizar en su aplicación en la tesis doctoral ya que el foco del mismo se encuentra en el despliegue a gran escala, principalmente en entornos de orquestación de recursos. El objetivo de la tesis es el desarrollo de una arquitectura orientada a un servicio, por lo tanto la escalabilidad podrá ser satisfactoriamente mediante la utilización de contenedores y la inclusión de conceptos *cloud-native* en los principios de diseño.

## 2.5.2. Seguridad en IoT

Los despliegues de IoT son sistemas complejos potencialmente susceptibles ante una gran variedad de ciberamenazas. El uso de enlaces (en su mayoría) inalámbricos para conectar los dispositivos, y el hecho de tener diferentes capas (con muchas opciones tecnológicas), solo por nombrar algunos, son aspectos que hacen que los entornos IoT sean propensos a ataques de varios tipos.

La seguridad en el IoT implica la inserción de cualquier característica a un sistema (o plataforma) con el objetivo de salvaguardar la integridad y proteger los datos, servicios, dispositivos y cualquier otro elemento informacional y/o físico que forme parte de dicho sistema. Dentro del ámbito de la seguridad suelen incluirse provisiones como privacidad, confianza o gobernanza de datos. No obstante, los objetivos de estas técnicas son diferentes y es preciso distinguirlos. La seguridad incluye protección de la solución ante potenciales amenazas y ciberataques (tales como suplantación de identidad, *man-in-the-middle*, etc.), mientras que la privacidad busca proteger los datos (especialmente, aquellos de carácter personal) y la confianza se centra en corroborar la identidad de los actores intervinientes. Antes de describir el análisis del estado del arte sobre seguridad en IoT, es conveniente destacar que este aspecto no es el foco de la tesis doctoral. Aunque ha de estar presente en toda solución IoT, la solución propuesta no persigue incrementar la seguridad de las plataformas IoT ni explorar nuevas técnicas o algoritmos, sino integrar métodos relevantes existentes, apropiados para los casos de utilización expuestos.

Abordar la seguridad en la implementación del IoT parece crucial, ya que la lista de amenazas presentes en todas las capas es grande y creciente [158]. Varios trabajos recientes en la literatura han explorado esas amenazas por capa [159] [160]. En este párrafo sólo se mencionan las más relevantes. Debido al hecho de utilizar dispositivos IoT de hardware, el propio proceso de fabricación podría implicar amenazas a la seguridad (por ejemplo, dispositivos deshonestos o entidades de desarrollo poco fiables), pero ese tema está fuera del alcance de los límites de actuación de las capacidades IoT. Además, a nivel de hardware, los intrusos podrían introducir ataques de “privación de sueño” y de “agotamiento” de la batería. Por otra parte, confiar en redes inalámbricas (potencialmente inseguras) abre la puerta a los ataques físicos, la replicación de nodos, el reenvío selectivo, el ataque de agujero de gusano o los ataques sibilinos. En las capas inferiores de las soluciones IoT (dispositivo, conectividad), los sensores y los dispositivos también pueden ser objeto de ataques de inyección de código malicioso, *sniffing* y *eavesdropping* [161]. Además, como los enlaces de comunicación del IoT están protegidos con estándares de cifrado más débiles debido a las características de ligereza de los dispositivos del IoT o *far-edge*, esto facilita la incursión por parte de los atacantes, lo que redundaría en la fuga de información u otras vulnerabilidades. La capa de red también es propensa a los ciberataques que pueden comprometer la confidencialidad a través de amenazas persistentes avanzadas, inferencias del hombre en el medio o ataques

de inyección de bases de datos (estos dos últimos principalmente en los despliegues de IoT orientados a *la nube*). Además, la denegación de servicio (DoS) es un tipo de amenaza común presente en el entorno del IoT. A nivel de aplicación, los *brokers* MQTT también pueden ser un inconveniente en la seguridad, ya que no incluyen TLS por defecto (configuración manual) [162]. Además, las aplicaciones se desarrollan con diferentes lenguajes de programación, lo que puede derivar en ataques de inyección de malware o fallos específicos de la tecnología (por ejemplo, lenguaje C).

Por todo lo anterior (entre otros aspectos que son demasiado amplios y numerosos para ser recogidos en este estado del arte), y al observar la creciente cantidad de literatura en torno a este tema, se puede concluir que hoy en día la seguridad es una de las mayores preocupaciones en el mundo del IoT [158]. Por lo que respecta al interés de la investigación, hay un claro aumento de la atención a las cuestiones de seguridad en el IoT durante los últimos años. Algunos trabajos [163] indican que en 2015 había pocas (o ninguna) revisiones sistemáticas sobre la seguridad en IoT, mientras que a finales de 2020 se podían encontrar más de 50 revisiones en revistas de altos índices de impacto. Entre ellas, los temas más discutidos son: “*seguridad, preservación de la privacidad, autenticación y blockchain*”. En cuanto a la atención privada, según IoTAnalytics, [164] las empresas aumentaron en un 40,3% la inversión en seguridad del IoT respecto al enfoque tradicional en datos, dispositivos, servicios y aplicaciones. Para concluir, también es destacable que las arquitecturas IoT más relevantes (partiendo de IoT-A [20] y OpenFog [81]) incluyen la seguridad en todas sus capas como un componente crucial para el despliegue de plataformas/sistemas de IoT.

No es objetivo de esta tesis doctoral innovar en este aspecto, ni dedicar gran parte de las investigaciones a la inclusión de mecanismos avanzados de seguridad (merecería un trabajo aparte). No obstante, resultó conveniente realizar una pequeña observación del estado del arte de estas técnicas en IoT, con la intención de seleccionar el mejor modo de incorporar la seguridad suficiente en la arquitectura a proponer.

En los siguientes párrafos se hace un repaso de los aspectos básicos de seguridad que se contemplan en los despliegues IoT: (i) confidencialidad, (ii) integridad, (iii) disponibilidad y (iv) control de acceso [165]. Se aporta una definición para cada uno de ellos, así como los métodos más empleados tras una revisión de la literatura disponible.

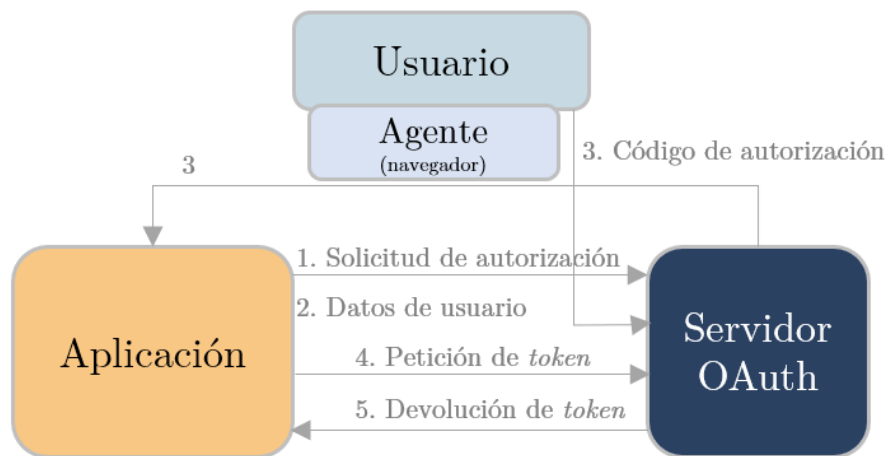
**Confidencialidad:** Consiste en asegurar que sólo los usuarios autenticados (autorizados) que han podido demostrar su identidad tienen acceso a los recursos o información del sistema IoT. En este sentido, las provisiones básicas que existen para garantizar la confidencialidad son: el uso de mecanismos de encriptación y mecanismos adecuados de autenticación.

En cuanto al cifrado, la literatura destaca que los más utilizados son esquemas de clave simétrica, AES128, AES256, RSA o SHA256 [161] pero se documenta que las plataformas IoT más modernas ya están proponiendo esquemas de cifrado más



avanzados (por ejemplo, encriptación homomórfica [166]). En este sentido, la principal limitación es la tendencia a la utilización de equipos con baja capacidad de computación (en los despliegues IoT) que impiden la utilización de estos esquemas. No obstante, con el constante incremento de capacidades en todo el mundo, esta tendencia se propagará hacia entornos *edge* y *far-edge* en los próximos años.

En lo concerniente a la autenticación, se trata de un aspecto en auge con respecto al interés por parte de la comunidad IoT [167]. La aproximación habitual consiste en la utilización de mecanismos de clave pública-privada. De hecho, la gran mayoría de plataformas basan esta autenticación en los protocolos SSL/TLS, asegurando que la información transmitida desde y hacia los servidores centrales que conforman el entorno IoT están protegidos [158]. Otras técnicas están siendo utilizadas y exploradas (como OpenID o PUF - *Physical Unclonable Function* [167]), siendo la más destacada OAuth (en su versión actual, OAuth2.0), que es capaz de garantizar autenticación bidireccional en los tres niveles de una topología IoT (sensor-nodo *fog-cloud*). OAuth basa su funcionamiento en una delegación de la autenticación cuyo objetivo es permitir el acceso al recurso a un usuario (por parte del propietario del recurso) sin necesidad de un intercambio de claves o contraseñas (que conllevaría un riesgo de seguridad). Técnicamente, utiliza *tokens* de autorización para permitir el paso de usuarios autenticados si se acepta por el propietario. La siguiente imagen resume el flujo de acciones que se llevan a cabo.



**Figura 2.32:** Ciclo del mecanismo OAuth

Otros mecanismos encontrados para mejorar la confidencialidad del despliegue son el uso de IPv6 sobre redes locales privadas inalámbricas (6LoWPAN) [158] o la aplicación global del principio de minimización de datos, significando que tanto los dispositivos, como el usuario, como todo el sistema tratan de reducir la comunicación de datos (personales y, en general) al mínimo posible para dotar de la funcionalidad esperada al despliegue.

**Integridad:** Los datos enviados por parte de los dispositivos y las aplicaciones tienen que ser fiables, estar inalterados y no haber sido falsificados.

Integridad es la capacidad de asegurar que los datos son completos, precisos y que no han sido manipulados sin autorización expresa.

En este sentido, existen herramientas externas para comprobar la integridad del sistema completo, especialmente si el despliegue es basado en *cloud*, utilizando regiones seguras o chequeo de máquinas virtuales. Otra opción considerada ha sido la limitación de la introducción de datos (o simple conexión/integración) de herramientas de terceros dentro del ecosistema, reduciendo la posibilidad de manipulación y creando un entorno más cerrado. Por otro lado, se están proponiendo herramientas tipo SAST y DAST para analizar dinámicamente los servicios, aplicaciones y código que forman parte del ecosistema [161]. En este sentido, una de las tendencias actuales para atacar los problemas de integridad es la incorporación de técnicas de DevSecOps (analizada en la sección 2.5.3).

**Disponibilidad:** Mecanismos para garantizar un servicio continuo y directo a un usuario autenticado. Idealmente, cualquier servicio provisto por un sistema IoT debería estar disponible tras petición y en cualquier momento (24/7). En muchas ocasiones, este aspecto no puede garantizarse debido a ataques sobre los dispositivos (p.ej., drenaje de batería), por problemas de la red o por desconfiguraciones. Las soluciones básicas para mitigar estos riesgos son: (i) selección de paradigmas de arquitectura que permitan un despliegue ágil y flexible y con configuraciones redundantes y (ii) el uso de APIs dinámicas para controlar el acceso a todos los servicios, añadiendo capacidades de conmutación por error y tolerancia a los fallos [160].

En cuanto a los paradigmas de arquitectura, véase sección 2.2. Con respecto a las APIs, la implementación más habitual de este tipo de garantías de disponibilidad es a través de las API Gateways, que exponen el resto de APIs subyacentes de los elementos de procesamiento de datos del entorno IoT. Suelen poder configurarse a través de una interfaz web y permiten el establecimiento de suscripciones, gestionadas por el administrador conforme (si se desea) a los sistemas de autenticación y autorización seleccionados. En este sentido, se han desarrollado también herramientas como el modelado de amenazas, el *framework* OWASP-TOP10 vulnerabilidades o MITRE ATT&CK que ejecutan funciones sobre dichas APIs para comprobar los niveles de seguridad y disponibilidad alcanzados [168].

**Control de acceso:** Los mecanismos de control de acceso son los principales mecanismos de seguridad para la aplicación de las características de seguridad que requieren la identificación con la debida autenticación y autorización. La autorización es un proceso de concesión, o de verificación automática, del permiso a una entidad para acceder a la información solicitada después de que la entidad haya sido autenticada.

Algunas técnicas utilizadas para implementar este parámetro de seguridad son el control de acceso basado en atributos (ABAC - *Attribute Based Access Control*), que básicamente utiliza atributos de los usuarios, del entorno, entre otros, para crear políticas de control de acceso; el control de acceso basado en roles (RBAC),

que perfila a los usuarios y asigna roles (que pueden ser dinámicos para conceder acceso - o no) o el uso de APIs. Las APIs, en este caso, pueden aprovechar mecanismos conocidos (de nuevo, OAuth2.0) y establecer un único punto final para canalizar el control de acceso de seguridad a través de verbos HTTP y URLs solicitadas [167].

Existen otros mecanismos adicionales para realizar el control de acceso, como la aplicación de políticas dentro de la red en forma de listas de control de acceso (ACL) específicas para cada puerto (o nombre o IP) [161].

En la siguiente tabla se exponen las estrategias, técnicas y metodologías seguidas por las plataformas IoT genéricas más utilizadas en cuanto a la cobertura de seguridad. Las indicaciones aportadas en las páginas previas, junto con esta perspectiva global, han servido como punto de partida para la selección de la aproximación a tomar en esta tesis doctoral.

**Tabla 2.5:** Mecanismos de seguridad de plataformas IoT

| Plataforma                     | Autenticación  | Control de acceso   | Otros  |
|--------------------------------|--|---|--|
| <i>OM2M</i>                    | TLS con certificados X.509. En COAP, <i>californio</i> . | Control de acceso por políticas (ACP) y reglas (ACR) tipo RBAC. | Protección de la identidad.                      |
| <i>FP7 OpenIoT</i>             | TLS y OAuth2.0   | RBAC con Apache Shiro y APIs REST                               | Configuración por UI con JBoss                   |
| <i>FP7 FIWARE</i>              | TLS cifrado RSA con certificados X.509<br>OAuth2.0       | IdM, PAP, PDP y reglas XACML (PBMS)                             | PEP adicional para gestión individual por UI     |
| <i>Sofía2 (from FP7 SOFIA)</i> | TLS cifrado RSA con certificados X.509<br>OAuth2.0       | RBAC con plug-ins propios de Spring.                            | Validación de mensajes JSON por ontología.       |
| <i>FP7 UniversAAL</i>          | TLS (externo) y RSA (componentes internos)               | Control por permisos del bus (Java) según usr/pwd.              | Documentos cifrados y gestión del consentimiento |
| <i>Azure IoT</i>               | TLS con X.509 y opción PSK para IoT                      | RBAC (roles IAM) y gestión de entidades                         | Etiquetas Azure Active Directory y               |
| <i>AWS IoT</i>                 | TLS cifrado AES con certificados X.509                   | RBAC (roles IAM) y políticas JSON                               | Identidad Cognito y AWS Signature                |
| <i>IBM Watson</i>              | TLS en HTTP y MQTT                                       | -   | Código de identificación IBM                     |
| <i>H2020 Symbiote</i>          | TLS y OAuth  | ABAC y JSON Web Tokens (JWT)                                    | -  |

|                              |  |  |  |
|------------------------------|--|--|--|
| <i>Google Cloud IoT Core</i> | TLS con X.509                                      | RBAC (roles IAM)                                       | -  |
| <i>ThingWorx</i>             | TLS con AES  | Acceso en fase de diseño y ejecución por perfil        | -  |
| <i>OracleIoT</i>             | TLS gestionado por OpenSSL                         | Tokens web JSON (JWT)                                  | Identidad digital única de dispositivo   |
| <i>Thingspeak</i>            | TLS  | Tokens de acceso a la API                              |  |
| <i>Kaa IoT</i>               | TLS cifrado RSA con certificados X.509<br>OAuth2.0 | Gestión de acceso e identidades con KeyCloak (SAML2.0) | Protección por API de los microservicios |

**Conclusión tras el análisis e influencia en la tesis doctoral:**

La seguridad es un aspecto muy relevante para los despliegues IoT. Al ser sistemas complejos compuestos de múltiples elementos que pueden escalar y crecer según una estructura de capas, son muchas y muy variadas las ciberamenazas que se ciernen sobre estos sistemas. Cualquier arquitectura IoT debe considerar incorporar seguridad; al menos, provisiones para asegurar un nivel de confidencialidad, integridad, disponibilidad y control de acceso suficiente, que son los pilares que deben cubrirse según el estudio de la literatura realizado. Existen muchas herramientas y técnicas para abordar esta protección, siendo las más comunes el cifrado y la autenticación, el establecimiento de control y autorización según identidades, atributos o roles y la creación de gestores de APIs que faciliten la disponibilidad y acceso a los recursos por parte de usuarios/elementos autorizados. Se ha realizado un repaso sobre las técnicas usadas por algunas de las plataformas IoT más utilizadas, que confirman las afirmaciones anteriores.

A pesar de que se trata de un aspecto muy relevante, esta tesis no se centra en el incremento de la seguridad en IoT sino de la incorporación de mecanismos que garantice un uso apropiado de los datos y la plataforma.

### 2.5.3. DevSecOps

Todo este capítulo relata cómo el espectro de características del IoT está evolucionando rápidamente. Esta evolución se basa en la introducción de conceptos y tecnologías más modernas, que sin duda plantean nuevos retos a diferentes niveles. La *Alliance for Internet of Things Innovation* (AIOTI) [169] también prevé un cambio en los modelos de gestión y control del flujo y la transmisión de datos.

Estos rápidos avances en los ecosistemas del IoT requieren enfoques más seguros y privados en términos de integración y despliegue continuos. En particular, para los escenarios de la próxima generación de IoT (NGIoT) [170] existe la demanda de ecosistemas altamente descentralizados que necesitarán ser apoyados por habilitadores de seguridad, privacidad y confianza para garantizar una adecuada protección humana (véase sección 2.2). Además, el enfoque centrado en el ser humano que caracterizará a los sistemas NG-IoT requerirá nuevas formas de interacción usuario-sistema de IoT, lo que plantea una serie de nuevos retos relacionados con la seguridad y la privacidad. Se espera que incluir estas características por diseño sea un aspecto crucial para mantener la confianza en las tecnologías NGIoT. En este sentido, dichas prácticas deberán extenderse a lo largo del ciclo de vida de los sistemas implicados, haciendo uso de metodologías de desarrollo de software seguras y nuevos habilitadores.

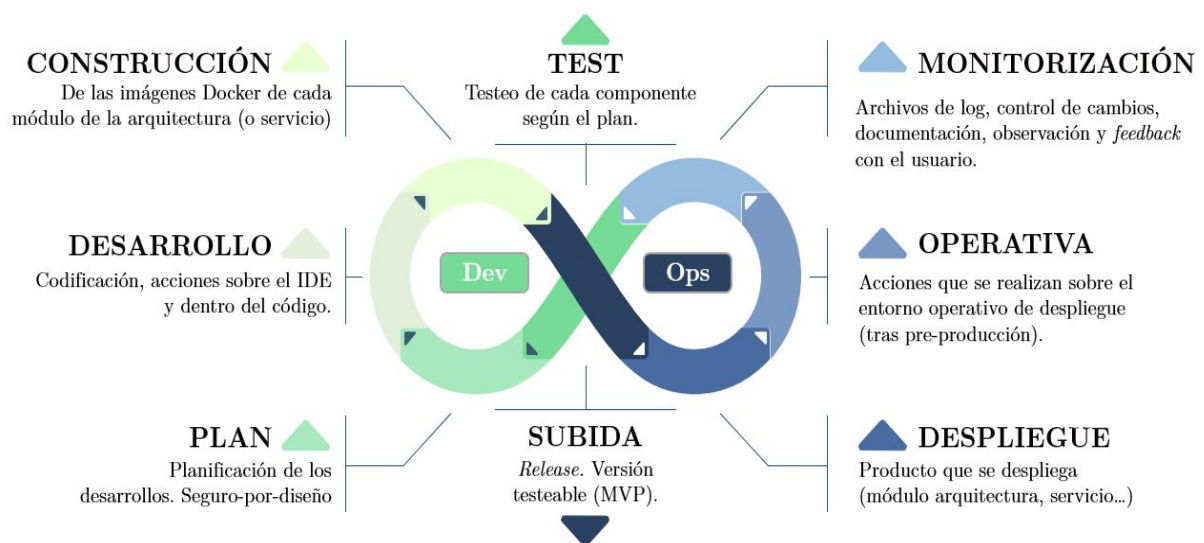
Hasta ahora, existen varias metodologías en la literatura (como señala GitLab [171]), siendo las más destacadas DevOps y Agile. Otras metodologías como Waterfall (ciclo secuencial de desarrollo), Kanban (cantidades más cortas de trabajo y foco en la comunicación del equipo) o Lean (centrada en la limpieza y la entrega rápida) son menos utilizadas hoy en día debido a sus inconvenientes, que van desde el alto coste y el esfuerzo o la integración y las pruebas tardías hasta la baja retroalimentación del cliente final durante el proceso [172].

El manifiesto de la metodología Agile [173] expone los principios en torno a los cuales orbita: software de trabajo temprano, interacción con el cliente y capacidad de respuesta a los cambios. Se materializa a través de *frameworks* como Scrum, el más popular con diferencia. Al implicar políticas, comunicación entre personas, papel de los individuos, etc., Agile se considera una estrategia de gestión de proyectos que lo abarca todo.

DevOps, por el contrario, establece un *pipeline* de acciones centradas en la producción, empaquetado y entrega del código. Por lo tanto, es mucho más adecuada para aquellos proyectos que necesitan maximizar el proceso de generación ágil de software de una manera muy directa. DevOps, como concepto, existe desde hace aproximadamente una década, pero aún no se ha concretado una definición consensuada del mismo. Según el candidato, una descripción válida sería la que se propone en [174]: “*un esfuerzo colaborativo y multidisciplinar dentro de una organización para automatizar la entrega continua de nuevas versiones de software, garantizando su corrección y fiabilidad*”. Básicamente, DevOps es una metodología que ayuda a las

organizaciones a construir software y a sus equipos de producción de forma que permita un despliegue rápido y continuo.

Reflexionando sobre su alcance en el ámbito de IoT, DevOps se está utilizando cada vez más en organizaciones e iniciativas que despliegan estos sistemas [175]. Su potencial se hace patente al observar que entornos de IoT suelen estar contruidos sobre Infraestructura como Código (*Infrastructure as Code - IaC*), y con fuertes dependencias de software en el despliegue. Las actualizaciones del firmware que se ejecutan cerca de los dispositivos IoT (agentes, en el caso de este doctorado) deben ser atendidas adecuadamente, planteando también amenazas de seguridad y potenciales rupturas de servicio. Paralelamente, los despliegues en *edge* y *cloud* necesitan herramientas que apoyen la automatización para implementar un flujo de trabajo continuo, reproducible y repetible.



**Figura 2.33:** Pasos en metodología DevOps

El campo de acción de DevOps se compone de varios bloques:

- *Planificación continua:* Implica la planificación, ejecución y seguimiento de las diferentes actividades de las fases de desarrollo, pruebas, subida de MVP, despliegue y operaciones. En el contexto de este trabajo, ha supuesto la planificación del desarrollo y la elaboración de una metodología para el despliegue de la solución (véase 5.2), junto con la configuración del entorno, la integración y la disponibilidad de la infraestructura, la comunicación y la documentación (finalizada con este manuscrito).
- *Integración continua:* Implica actividades relacionadas con la colaboración en equipo para la correcta producción de código y productos compuestos. Suele materializarse (siempre que haya equipos) en herramientas de software colaborativo, siendo Git la más destacada. En el caso de este doctorado, algunos trabajos se han realizado de forma

colaborativa (dentro de proyectos H2020), y en esos casos se han utilizado estas técnicas. En los trabajos individuales también se ha seguido una estrategia de versionado y actualización del software.

- *Entrega continua*: En el contexto del doctorado, se trata de un bloque bastante similar al anterior, abarcando esta vez las fases de liberación y prueba. Como se describe en la metodología, cada agente de un despliegue debe pasar por algunas pruebas.
- *Despliegue continuo*: En la especificación DevOps, esta parte refleja la automatización de todos los pasos en el despliegue, incluyendo la construcción automatizada o el aprovisionamiento de la infraestructura (fuera del alcance de este trabajo).
- *Control de operativa continuo*: El objetivo del control de operativa continuo es garantizar la continuidad del servicio, lo que se traduce en un registro y escaneo continuos. En la arquitectura propuesta implicaría el registro de los errores de ejecución de las diferentes partes del sistema. Si bien esto está contemplado, no se ha identificado la necesidad de una automatización de los informes y de la resolución de problemas, por lo que no se sigue completamente.
- *Feedback continuo*: Herramientas para permitir la retroalimentación continua de los usuarios con respecto a la analítica, el rendimiento, los requisitos, etc. Considerado para el trabajo en esta tesis como parte de la participación del candidato en varios proyectos de investigación con contacto continuo con potenciales “clientes de la solución”.

En este sentido, como se ha comentado en los puntos de la lista anterior, en el trabajo de la tesis se ha seguido en cierta medida la metodología DevOps. Si bien es cierto que no existe la necesidad de gestionar la comunicación/colaboración de un equipo, sí que hay algunas características como la automatización, la monitorización y la sistematicidad, incrustadas dentro de la metodología, que podrían ayudar en gran medida a la evolución del trabajo en la tesis. Esto es especialmente relevante teniendo en cuenta que se ha planteado la generación de productos válidos intermedios (*Minimum Value Product* – MVP) para que puedan tener lugar los escenarios de validación (que aparecen en momentos no correlativos en el tiempo ya que son dependientes de la ejecución de los proyectos de investigación en los que ha participado el doctorando).

En los esfuerzos por mejorar continuamente la metodología DevOps, surgió una extensión para cubrir cuestiones relacionadas con la seguridad, dando lugar a la llamada metodología DevSecOps. El principio de seguridad por diseño es uno de los enfoques más relevantes para tener en cuenta para el desarrollo y la explotación del software en el IoT [176] (dentro de esta tesis, ha merecido su propia sección de estudio (véase sección 2.5.1) y un módulo específico en la arquitectura de la solución -3.6). Sin embargo, la seguridad en el proceso de DevOps tradicional únicamente se mencionaba en las etapas finales. Para dar respuesta a esta limitación, DevSecOps

surge como un esfuerzo consciente por desplazar las consideraciones de seguridad hacia la “izquierda” en el *pipeline* de desarrollo, para abordar adecuada y oportunamente cualquier consideración de seguridad [177].

A nivel práctico, DevSecOps aborda la seguridad por diseño y las pruebas continuas mediante la construcción de controles de seguridad automatizados en el flujo de trabajo de DevOps. Los siguientes párrafos (crédito de imagen base: [Slidesgo](#) y [Freepik](#)) destacan las principales extensiones de DevSecOps sobre los principios de DevOps, indicando al mismo tiempo cuáles de ellas pueden ser relevantes para ser consideradas en el trabajo de esta tesis.

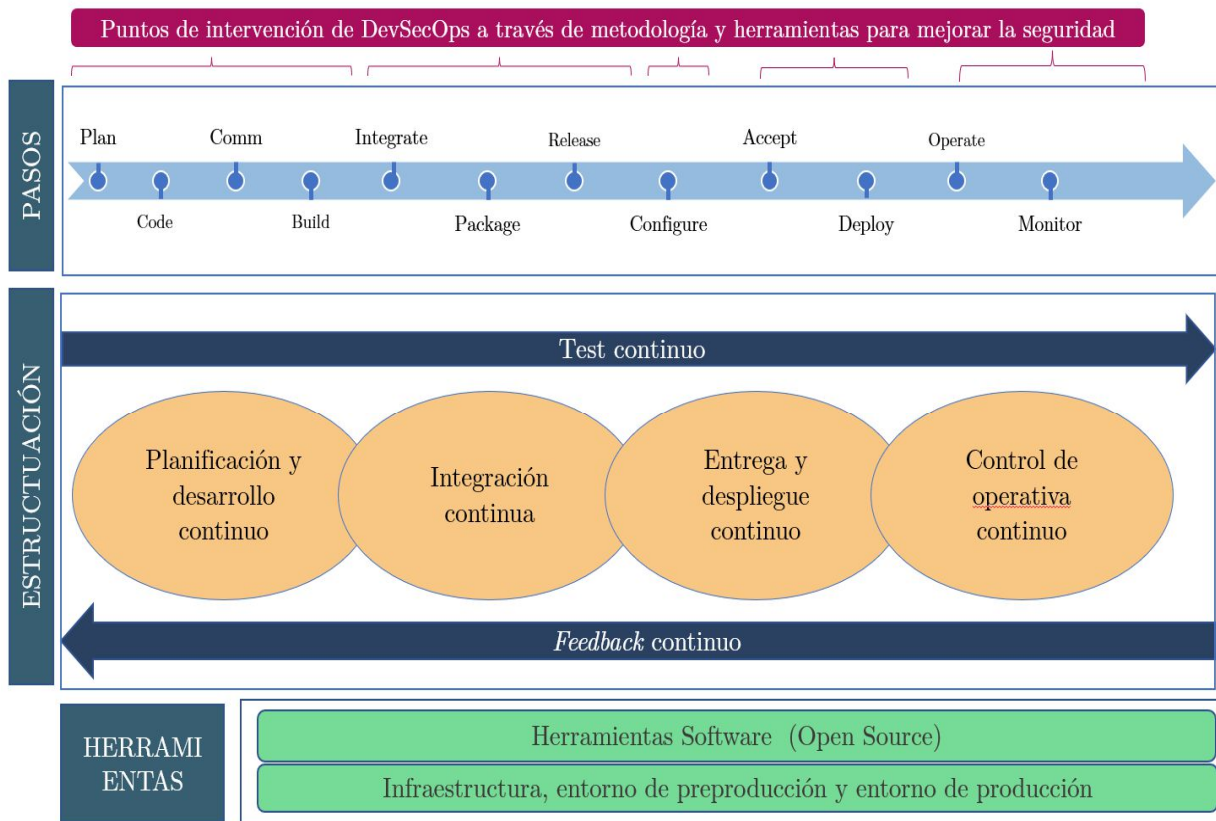
**Pruebas continuas:** Los controles de pruebas se ejecutan en paralelo con otros controles de seguridad en diferentes etapas del flujo de trabajo. Mientras que las pruebas tradicionales de DevOps incluían técnicas de pruebas funcionales y no funcionales, DevSecOps introduce técnicas adicionales de pruebas de seguridad, como el análisis de composición de software (SCA), las pruebas estáticas de seguridad de aplicaciones (SAST), las pruebas dinámicas de seguridad de aplicaciones (DAST), las pruebas interactivas de seguridad de aplicaciones (IAST) y la protección de seguridad de aplicaciones en tiempo de ejecución (RASP).

**Feedback continuo:** Adicional a DevOps, DevSecOps también extiende el *feedback* continuo a otras etapas del flujo de trabajo, para obtener retroalimentación de la fase de Construcción y Prueba, con el fin de incluir informes de seguridad relevantes.

Partiendo de estas premisas, se han hecho suposiciones para comprender el papel potencial de DevSecOps en los despliegues de NGIoT y en esta tesis [178]:

- Se supone que los sensores IoT de las máquinas que proporcionan datos son fiables.
- Se supone que las estaciones de trabajo utilizadas para acceder a los datos recogidos de las máquinas son fiables y operativas.
- Se supone que los programas informáticos de los servidores desplegados en el marco de los servicios del IoT son fiables y se actualizan continuamente.
- Se supone que los elementos de computación dentro del espectro del despliegue IoT (nodos *edge*, servidores, *cloud*...) son fiables en lo que respecta a la protección de datos y la privacidad, y son fiables aplicando los mecanismos de seguridad adecuados a nivel de red.
- Los usuarios del software están autenticados y autorizados, y el propietario del servicio prestado se considera de confianza.





**Figura 2.34:** Evolución hacia DevSecOps

Las actualizaciones seguras de software y firmware son una medida técnica para el IoT, asociada a la autorización, tal y como se describe en las buenas prácticas de ENISA para IoT [176].

Existen trabajos que han desarrollado cuáles deben ser las acciones particulares (y cuáles son las herramientas *open source* recomendadas) en cada uno de estos pasos del ciclo de DevSecOps particularizado para despliegues NGIoT [172].

En la siguiente tabla se muestra una lista con herramientas interesantes para cada uno de dichos pasos que podrían ser considerados para soluciones (NGIoT) similares a la planteada en esta tesis. Cabe destacar que no existe ningún requisito específico en este trabajo que fuerce la utilización de ninguna de ellas en la cobertura de DevSecOps:

**Tabla 2.6:** Pasos, herramientas y métodos DevSecOps y su encaje en la tesis

| Paso                          | Herramienta / comentario   |
|-------------------------------|--|
| Planificación ( <i>plan</i> ) | La solución ha sido ideada segura-por-diseño, por tanto, se ha tenido en cuenta este paso. Se han realizado diagramas de flujo de datos e interacciones entre módulos. OWASP TOP-10 o Mitre Attack son algunas de las herramientas recomendadas. |

|  |  |
|--|--|
| Desarrollo<br>( <i>código</i> )          | Los plugins en los entornos de desarrollo son una buena herramienta para garantizar la seguridad en el código. Se recomienda la documentación y el control de versiones del código. Todo lo anterior parece ser relevante para este trabajo.   |
| <i>Construcción</i>                      | Análisis de composición estática (SCA) o pruebas de seguridad estática de aplicaciones (SAST).   |
| Prueba                                   | En el caso de entornos <i>edge</i> , que es un aspecto clave en los escenarios de NGIoT, la integración debe probarse cuidadosamente (véase 5.2). Como herramientas, se recomienda el análisis de escaneo de contenedores, pruebas dinámicas de seguridad de las aplicaciones (DAST) o Pruebas de Seguridad de Aplicaciones Interactivas (IAST).       |
| Emisión de versión<br>( <i>release</i> ) | Se trata de herramientas a utilizar/ejercer sobre el repositorio de artefactos (imágenes) en caso de disponer de uno local (o en un proveedor <i>cloud</i> , por ejemplo, GitLab). Se recomienda el aseguramiento de seguridad perimetral, autenticación para la subida de una versión y Autoprotección de aplicaciones en tiempo de ejecución (RASP). |
| <i>Aceptación</i>                        | Realización de estas actividades en un entorno de preproducción; donde se debe obtener validación por parte de todos los actores previo paso a un entorno productivo.  |
| <i>Despliegue</i>                        |  |
| Operativa<br>( <i>operate</i> )          | En este paso se recomienda realizar de nuevo una ronda de test DAST, SAST, IAST y RAST sobre el entorno operativo donde se encuentra ejecutándose la solución.   |
| <i>Monitorización</i>                    | Monitorización de logs y tests de seguridad periódicos contra vulnerabilidades. Potencial ejercicio de equipo rojo externo vs. equipo azul interno [179].  |

**Conclusión tras el análisis e influencia en la tesis doctoral:**

La metodología DevOps establece un *pipeline* de acciones centradas en la producción, empaquetado y entrega del código que facilita la automatización de las tareas, el trabajo en equipo y, en general, el proceso de producción de software. DevSecOps es una evolución natural que se centra en incluir mecanismos de seguridad desde fases tempranas de dicha producción. A través de herramientas y metodologías, DevSecOps es aporta valor a la cadena de generación de soluciones TIC. En el contexto de esta tesis, ha resultado interesante realizar este estudio del ya que algunos de los conceptos (tanto de DevOps como de DevSecOps) son de aplicación, tales como seguridad por diseño, utilización de repositorios de código automatizados (Git), la utilización de entornos de preproducción, o el testeo en diferentes fases incluyendo *feedback* por parte del usuario.

## 2.5.4. Tactile Internet

*Tactile Internet* es una de las tendencias tecnológicas más recientes. Según el *Telecommunication Standardization Sector* de la *International Telecommunication Union* (ITU-T), el *Tactile Internet* (TI) se caracteriza por una latencia extremadamente baja, combinada con una alta disponibilidad, fiabilidad y seguridad [180]. Se denominan así aquellos sistemas/despliegues que tienen por objetivo aprovechar las capacidades de las redes ultrarrápidas para permitir una interacción háptica entre hombre y máquina.

Las aplicaciones del TI son múltiples y diversas, y su relevancia, observada en la literatura, está creciendo rápidamente (por ejemplo, desde la introducción del término en 2014, en el primer trabajo [181], se han publicado más de 150 artículos a mitad del año 2021 y la tendencia sigue creciendo). Además, el auge de tecnologías relacionadas, como los robots sociales, las redes de detección inalámbricas y el Internet de las Cosas de Nueva Generación (donde se encuadra la arquitectura de esta tesis), contribuyen al del TI. Los siguientes párrafos resume los puntos más importantes de la *Tactile Internet* de hoy y esboza los aspectos clave de las arquitecturas de NGIoT para considerarse listas para cubrir los requerimientos del *Tactile Internet*.

Se ha documentado un aumento de las entidades oficiales que incluyen el TI entre su ámbito de actuación. En este sentido, el grupo de trabajo de estandarización de AIOTI definió el TI (en 2018) como: “*un facilitador de tecnología para aplicaciones de ultra-alta fiabilidad y ultra-baja latencia, donde la red es parte de un bucle de retroalimentación háptica* [182]. La iniciativa NG-IoT presentó después al TI como una de las tecnologías habilitadoras de la próxima generación de IoT.

La aplicación del *Tactile Internet* promete ayudar a la Industria a atacar aquellos problemas que van más allá de las interacciones visuales y/o auditivas con las capacidades actuales. Un ejemplo es el uso de robots en la fabricación. La gestión de este tipo de maquinaria de forma remota, replicando los movimientos de los operarios, sin poner en riesgo su salud, ha sido relatada como una clara aplicación beneficiosa de esta tecnología [183]. Además, las aplicaciones críticas, en las que intervienen seres humanos, pueden beneficiarse del TI. El número de sectores que podrían beneficiarse del TI en un futuro es amplio. Se exponen a continuación unos pocos ejemplos que pueden relacionarse con los objetivos de esta tesis:

- Facilitar la intervención a distancia sobre el terreno por parte de expertos en alguna materia, sin necesidad de una presencia física. En este caso, es necesario disponer de movimientos hápticos, accionables a través de una máquina (por ejemplo, un robot).
- Facilitar operaciones de cualquier tipo a distancia en lugares inaccesibles o para la aplicación de políticas de una forma rápida (por ejemplo, drones, acciones en los fondos marinos, en incendios, misiones

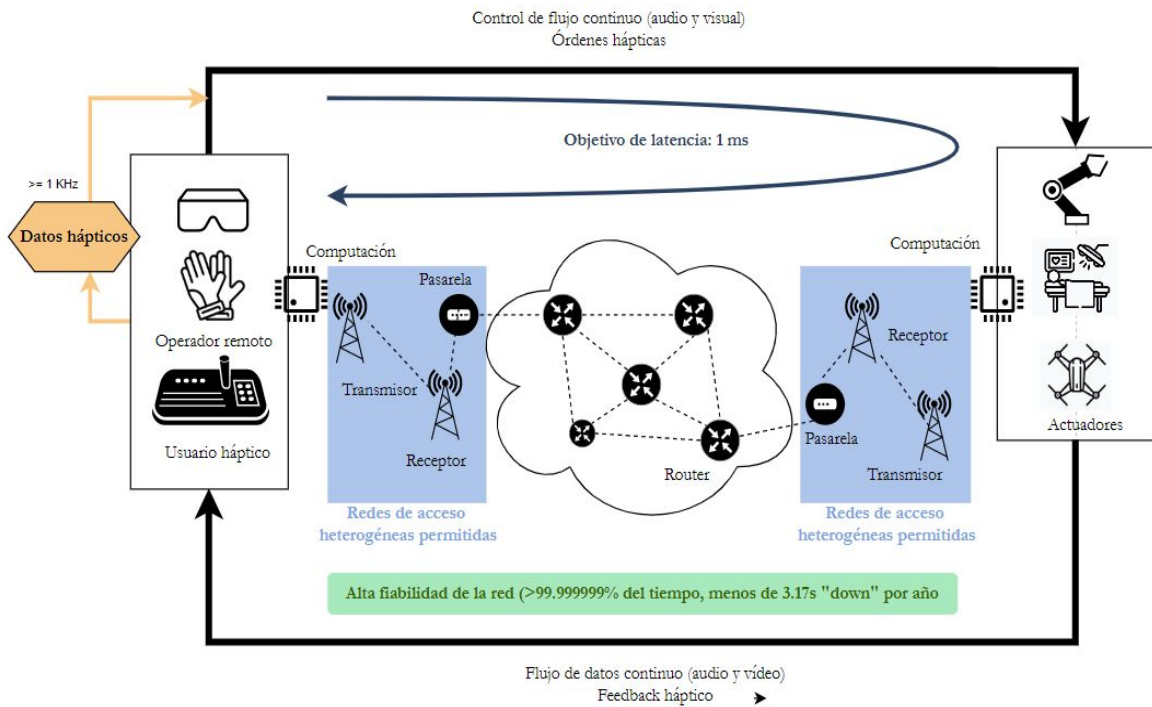
de rescate, etc.).

- Crear experiencias remotas para la interpretación e interacción con sistemas complejos, que permitan visitar sitios sin estar físicamente presentes. En este caso, el despliegue a gran escala podría suponer enormes beneficios en diversos contextos (por ejemplo, reducción del tráfico, traslado de personas, etc.) relacionados con la monitorización y comparación del impacto de fenómenos reales en/por parte de las entidades.

Existe debate sobre si el término TI debe utilizarse para describir aplicaciones (sistemas de control inalámbricos en red, sistemas de operación remota, sistemas de entretenimiento inmersivo y *edutainment*, sistemas de movilidad inteligente, sistemas de energía inteligente), o como un conjunto global compuesto por tecnologías habilitadoras (inalámbricas de alto rendimiento, requisitos de conectividad y visión) [184].

El candidato, en base a la experiencia adquirida en el proyecto ASSIST-IoT, razona que el TI debe entenderse como cualquier sistema que (i) transporta con éxito datos hápticos mediante (ii) una comunicación que cumple los requisitos de (a) baja latencia, (b) alta fiabilidad y (c) alto rendimiento de datos.

En la siguiente figura se muestra un esquema general de un despliegue de comunicación de *Tactile Internet*, mientras que en los siguientes párrafos se describen brevemente los puntos mencionados (requisitos de comunicación, intercambio de datos y requisitos de infraestructura).



**Figura 2.35:** Estructura de comunicación en Tactile Internet. Imágenes: Flaticon

En cuanto a los requisitos de comunicación, y debido a la participación humana, un despliegue de TI debe responder a la exigencia impuesta por la sensación del tacto, estimada en una resolución temporal de 1000 Hz [185]. Para que la experiencia sea realista, la TI debe superar una latencia de comunicación de 1 ms, desde el momento en que se inicia la acción (por ejemplo, el usuario mueve un dedo, o el operador remoto presiona el borde de un guante), hasta que se recibe la respuesta y no superar en exceso este *margen de feedback*.

Para conseguirlo, el TI impone unos requisitos tecnológicos estrictos para la comunicación de extremo a extremo en tiempo real. Para conseguirlo, las evoluciones tecnológicas actuales en el campo de las redes (por ejemplo, SDN, NFV, antenas de ancho de haz estrecho, MEC en 4G/5G) se antojan claves para cumplir dichos criterios de comunicación.

En términos de protocolo de comunicación ultrarrápida, se basa en el intercambio de datos hápticos que son codificados extremo a extremo mediante códecs hápticos (véase Figura 2.35). Estos esquemas de comunicación se basan en mecanismos *maestro-esclavo*. En el contexto de la comunicación de TI, el maestro sería, por ejemplo, una entidad controladora remota (humana o máquina) que, a través de una interfaz, codifica la entrada háptica en señales de comando y las envía al esclavo. El esclavo podría ser un objeto/robot, guiado directamente por el controlador remoto (maestro), capaz de proporcionar retroalimentación y cerrar el bucle [186]. Pueden responder a dos tipos de estructura: (i) de retroalimentación de bucle cerrado (es decir, la percepción), que requiere estar lo más cerca posible del tiempo real (intolerante al retraso), y (ii) de retroalimentación de bucle abierto, que es menos restrictiva y acepta la comunicación con retraso.

En general, los esquemas de codificación y los códecs asociados que aplican algoritmos de transformación de datos son una línea de investigación abierta destinada a reducir la cantidad de datos hápticos transmitidos, para alcanzar la validez de la percepción humana.

Aunque son varias las iniciativas detectadas en el estudio de literatura realizado, el estado de la técnica no señala actualmente un producto/solución de referencia claro que reúna todas las características e innovaciones anteriores en una única especificación. Análogamente, hasta el momento no existe un estándar claro al que atenerse en cuanto a la implantación del *Tactile Internet*. Se ha llegado a estas conclusiones tras realizar una revisión inicial de las propuestas de formalización del TI, cuyas conclusiones más relevantes son las siguientes:

- A). En 2014, el UIT-T [180] propuso enmarcar la comunicación dentro de los sistemas de TI utilizando un esquema maestro-esclavo, razonando que es lo más natural, y basándose en los sistemas de telecomunicaciones heredados, anteriores a la era TCP/IP.
- Unos años más tarde, el grupo IEEE 1918.1 propuso una posible arquitectura basada en el mismo concepto de maestro-esclavo, pero enfocada a una conectividad y actuación en el borde de la red (*edge*

*computing*), estableciendo un nodo pasarela para conectar los esclavos a la red. La arquitectura incluía el nodo pasarela en el *edge*, cerca de los actuadores maestro y esclavo, o en la red local de actuación.

- Las contribuciones del UIT-T y el IEEE, junto con otras arquitecturas que se encuentran en la literatura, son principalmente teóricas y genéricas, y no incluyen aspectos reales/prácticos (por ejemplo, incluyendo una lista de tecnologías y cómo integrarlas para garantizar un funcionamiento acorde a los requisitos expuestos) para implementar una aplicación real de *Tactile Internet*. Además, carecen de un vínculo con el papel que debe desempeñar el IoT (sensores, pasarelas, actuadores, intercambio de datos, redes de acceso) en las especificaciones. Este punto es el más relevante para esta tesis doctoral, de tal manera que puede concluirse que la integración de IoT dentro de los planteamientos TI es, a día de hoy, un campo de investigación poco explorado y con mucho recorrido.

Al margen de estas propuestas oficiales, se han documentado trabajos que sí buscan el despliegue de prototipos en escenarios reales. Se mencionan algunos ámbitos y ejemplos que resultan de interés para esta tesis doctoral:

- Aplicaciones en el ámbito de realidad virtual (VR – *Virtual Reality*): en general, las aplicaciones de transmisión de vídeo no implican la inclusión de datos hápticos, ya sea porque el maestro (receptor de los datos) sólo precisa de imagen y sonido y no hay interacción que controlar, o bien porque se trata de una aplicación de interacción con órdenes y metadatos, sin la inclusión de vídeo. Sin embargo, las imágenes de alta resolución y el audio estéreo en 3D de algunas aplicaciones de realidad virtual y realidad aumentada conllevan un gran volumen (flujo masivo) de información y, por lo tanto, plantean diferentes retos que pueden beneficiarse de las características del TI. Por ejemplo, en relación al rendimiento de la red (*throughput*) o la latencia, que tiene que ser inferior a 10 ms, para evitar el mareo cibernético (*cybersickness*) [185].
- Algunas aplicaciones encontradas en el ámbito de ciudades inteligentes (*Smart Cities*) también utilizan TI y comunicaciones M2M relevantes. En [187], se propone una arquitectura de cinco niveles orientada a la maximización de la calidad de experiencia (QoE – *Quality of Experience*). De este modo, ciudadanos y entidades podrían monitorizar e interactuar de manera táctil con elementos de la ciudad que estuvieran accesibles de manera remota con una gran experiencia de utilización. Técnicamente, plantea una agregación del espectro entre bandas durante la transmisión y computación osmótica [188] para gestionar la carga en la integración entre los entornos *edge* y *cloud*. La optimización de la red, adaptando los recursos a las demandas, se consigue mediante la toma de decisiones inteligentes en la capa de

aplicación. Este caso de uso es extremadamente interesante para la tesis (y tal vez justifica la inclusión de TI como una futura línea de trabajo en la solución) por dos motivos:

- (1) Varios de los escenarios de validación de la solución se han realizado en entornos de ciudad-puerto inteligente (*Smart Port-City*) que plantea desafíos y características similares a las *Smart Cities*.
- (2) Podría vislumbrarse un potencial uso de la solución de esta tesis como una herramienta para la interacción en tiempo real sobre una infraestructura de TI. Así, tanto entidades públicas (por ejemplo, ayuntamientos, organizaciones con capacidad de toma de decisiones, etc.) como ciudadanos podrían monitorizar fenómenos complejos a partir de índices compuestos seleccionados y actuar sobre dichos fenómenos (por ejemplo, elementos de las ciudades como aceras, parques, etc.) con una baja latencia y gran calidad de experiencia (QoE).

Cabe señalar que buena parte de los artículos estudiados expresan implícitamente la necesidad urgente de una arquitectura de referencia válida, robusta y escalable, centrada en la aplicación práctica de la *Tactile Internet*. Los ejemplos proporcionados ilustran la variación de la importancia de los requisitos de TI, de una aplicación a otra. A raíz de lo observado, puede decirse que el TI parece requerir características particulares en función de (i) el vertical donde vaya a desplegarse y de la aplicación concreta (datos hápticos a intercambiar, comandos, flujos de información) y de (ii) los requisitos de la infraestructura (red, latencia, protocolos de acceso), que pueden variar entre los distintos ámbitos de aplicación. Por tanto, es crucial incluir las estrictas limitaciones del TI en las decisiones de diseño de cualquier arquitectura de sistemas distribuidos o de Internet de las Cosas donde quisieran desplegarse este tipo de aplicaciones. En este contexto, se observa que, aunque la arquitectura diseñada en esta tesis doctoral será agnóstica a la aplicación, se ha validado en varios escenarios de aplicación (sección 5). Aunque no se ha expresado ningún requisito específico sobre las necesidades de latencia o de *Tactile Internet* (véase sección 3.1.1), es muy interesante reflexionar sobre cómo éste puede afectar al diseño de la arquitectura. El servicio de indicadores compuestos (por su naturaleza de cálculo retrospectivo) no promete requerir *Tactile Internet* en un futuro próximo, pero si la evolución del IoT y otras técnicas lo permiten, servicios como el propuesto en este manuscrito pueden derivar en decisiones automatizadas para corregir el impacto de las entidades en un fenómeno real con fuertes componentes de tiempo real (véase el ejemplo expuesto en la página anterior). En ese caso, sería interesante que arquitecturas como esta comenzaran a tener en consideración las preocupaciones del TI para permitir crecer sobre ellas cuando estos requerimientos alcancen de pleno el ámbito de aplicación del IoT.

Con la intención de cubrir estos aspectos de TI, que responden a las

necesidades expresadas a lo largo de la subsección, la arquitectura de la tesis ha sido diseñada con los siguientes principios en mente:

- La arquitectura debe ser flexible ante diferentes situaciones, dependiendo de los requisitos (nótese que todos los escenarios vislumbrados de TI requieren una red de alta capacidad). En este sentido, debería ser lo suficientemente robusta y genérica para aceptar protocolos y dispositivos de red orientados a proporcionar: (a) alta disponibilidad, (b) gran ancho de banda, o (c) baja latencia. En este contexto, la arquitectura diseñada debe poder permitir en un futuro la instalación de funciones virtualizadas de red y la modificación dinámica de la red, para escenarios y requisitos cambiantes dentro del mismo despliegue, aprovechando las técnicas SDN y NFV.
- Debe otorgarse un papel clave a los nodos de computación en borde (*edge computing*), ya que en escenarios TI serán los encargados principales de facilitar un entorno de baja latencia, y el uso de sistemas avanzados de información visual para la toma de decisiones.
- Se debe establecer un claro rol para la codificación de los datos provenientes de las diferentes fuentes. En estos puntos (agentes dentro de la arquitectura propuesta) deberá poder ejecutarse el software de los códecs hápticos en caso de tratarse de un escenario de TI..
- El componente de *edge computing* planteado para la arquitectura, en caso de elegir/recomendar un elemento hardware en particular (pasarela IoT encargada de desplegar dichos agentes), éste debe ser capaz de soportar múltiples redes de acceso (5G, 4G, WiFi, Ethernet, Zigbee, LoRa, Bluetooth).

#### **Conclusión tras el análisis e influencia en la tesis doctoral:**

Las consideraciones anteriores se han tenido en cuenta a la hora de establecer los requisitos técnicos y los principios de diseño de la arquitectura planteada, agnóstica al dominio e implementación, soportando varios tipos de aplicaciones. Por otro lado, el elemento de preprocesamiento de datos (*data curation*), aunque en la arquitectura diseñada se recomienda *pyngsi* para la implementación de estos agentes, se trata de un componente clave en caso de existir requerimientos de TI. Además, el planteamiento indicado prevé la posible inclusión de habilitadores adicionales, para la agregación y anotación semántica de datos hápticos, necesarios para codificar la información en el mismo punto. Por último, la participación en el proyecto H2020 ASSIST-IoT, con aplicaciones de estas premisas TI en entornos marítimo portuario, de construcción y de automoción han influido a la hora de seleccionar tecnologías para la implementación de la arquitectura propuesta.



# Capítulo 3

## Definición e implementación de la arquitectura

### 3.1.Introducción

El objetivo principal de la tesis doctoral ha sido el desarrollo de una arquitectura de IoT de nueva generación para el cálculo y predicción de índices compuestos.

Este capítulo se centra en cómo dicha arquitectura ha sido ideada, diseñada e implementada a nivel laboratorio (aplicaciones reales pueden consultarse en la sección 5). Se trata, por tanto, de uno de los capítulos más importantes de la tesis doctoral, reflejando la mayor parte del bloque tecnológico de trabajo en la tesis, sobre dónde se cimientan el resto de las investigaciones y demostraciones.

La construcción de esta arquitectura parte de los conocimientos obtenidos por el candidato tras el análisis del estado del arte del capítulo anterior y a partir de la participación en proyectos de investigación como INTER-IoT, PIXEL y ASSIST-IoT.

Aprovechando análisis realizados por el candidato con anterioridad [189], y tomando como punto de partida las arquitecturas RAMI4.0, IIRA, AIOTI y la plataforma PIXEL [19], se ha diseñado un sistema modular con una base sólida. Se parte de un enfoque consolidado de capas para la implementación orientada a servicios de IoT para despliegues prácticos factible para una gran cantidad de entornos, aunque con una validación centrada principalmente en escenarios de ciudad-puerto. La solución persigue también validar la hipótesis inicial haciendo uso de componentes disponibles en abierto en el estado del arte, junto con algunos módulos personalizados y extensiones cuidadosamente seleccionadas.

El objetivo técnico de la arquitectura es componer un sistema basado en los conceptos y técnicas del Internet de las cosas para procesar datos heterogéneos y convertirlos, después de la ejecución de cierta inteligencia, en indicadores compuestos interpretables para la aplicación de políticas de sostenibilidad y mejora organizativa sobre ellos. La intención es plantear una arquitectura simple y práctica, pero a la vez integral, para cubrir las necesidades de cualquier tipo de despliegue (*edge*, *cloud*, a baja escala, a gran escala, rápido, concienzudo...) y cualquier tipo de índice compuesto basado en una gran cantidad de indicadores iniciales.

A su vez, el objetivo es completar el diseño teniendo en cuenta las capacidades de Internet de la Cosas de nueva generación (véase sección 2.2) de tal manera que la arquitectura se encuentre preparada desde su creación para las tecnologías de interés creciente como *blockchain*, *Tactile Internet*, interfaces de realidad virtual, mixta y aumentada y para nuevos conceptos de inteligencia artificial como aprendizaje federado. Asimismo, ha sido concebida de una forma holística para servir tanto como un sistema IoT por sí solo (en caso de iniciar un despliegue desde cero) como para integrarse fácilmente con otras arquitecturas IoT (la interoperabilidad no ha sido un objetivo principal, pero se han incluido las disposiciones pertinentes) y otros sistemas de información.

Conviene recordar que, aun tratándose de una arquitectura IoT perfectamente válida para una multitud de escenarios, ha sido diseñada con la intención de estar preparada (nativamente) para el servicio de cálculo y predicción de índices compuestos. Por tanto, a la hora de capturar requisitos y de diseñar ciertos elementos (por ejemplo, el entorno y las capacidades de visualización), se ha tenido en cuenta el servicio de CI que se ejecutará sobre esta arquitectura.

Estas consideraciones debieron ser tenidas en cuenta desde el principio del diseño arquitectural para maximizar la eficiencia, ajustar los componentes a las necesidades posteriores y para evitar rediseños que postergaran el proceso de creación de la solución.

### 3.1.1. Requisitos

Según buenas prácticas en la ingeniería del software, el éxito de una arquitectura comienza con la captura de requisitos [190]. El objetivo de esta acción fue el de reunir y analizar las virtudes esperadas y las necesidades a cubrir, que su vez debían impulsar el desarrollo del software, el encaje del servicio de CI y la definición de los escenarios de validación. La obtención y validación de los requisitos ha seguido enfoques metodológicos para garantizar la coherencia general y maximizar la utilidad para todas las fases técnicas del proyecto.

De este modo, en el momento de plantear la arquitectura completa, se tuvieron en cuenta tres tipos de requisitos:

- **Técnicos (no funcionales):** las propiedades generales de la arquitectura, bebiendo directamente de las características de los sistemas IoT.
- **Funcionales:** capturando el comportamiento del sistema, enlazando de una forma más directa con los módulos que la componen (interfaz de usuario intuitiva, funcionalidades de visualización y configuración, etc.), relacionados con el objetivo de la arquitectura (cálculo de índices compuestos) pero manteniendo un perfil general para cubrir aplicaciones más genéricas.
- **De servicio:** podría considerarse una subclasificación de los elementos funcionales, ya que se tratan de aquellas características o disposiciones específicas asociadas al servicio de CIs. Han decidido mantenerse separadas (e incluirse en este punto en vez de en la sección 4) ya que se consideraron desde fases iniciales de la tesis doctoral.

Como se observa en la siguiente figura, esta estructuración de requisitos está fuertemente inspirada en varias proposiciones modernas [191] adaptadas del modelo jerárquico de pirámide de requisitos publicado inicialmente por Liffingwell y Widrig en 2003 [192].



**Figura 3.1:** Estructuración de requisitos para el diseño de la arquitectura

El enfoque escogido, abanderado por la metodología VOLERE [193], consistió en asegurarse que cada uno de los requisitos definidos contaba con las siguientes características:

- **Necesario:** en caso de no incluirse, la arquitectura contaría con características o necesidades insatisfechas.
- **Apropiado:** la intención del requisito es clara, acota su espectro de actuación garantizando así su independencia y evitando solapes.
- **No ambiguo:** el grado de detalle es acorde a las expectativas de desarrollo y está expresado de forma concisa con una sola interpretación posible.
- **Completo:** el requisito es autocontenido y entendible sin necesidad de consultar otra información u otros requisitos.
- **Singular:** debe representar una única característica y cualidad, no abarcando una serie de ellas.
- **Factible:** debe poder ser realizado dentro de las limitaciones de la solución y en consideración del sistema y de la tesis en general.
- **Verificable:** debe ser posible de analizar si se ha cumplido o no una vez finalizada la tesis doctoral. Esto puede ser tanto a nivel integración en entorno de desarrollo (véase 3.7) como durante los escenarios validación (sección 5).
- **Correcto:** debe representar directamente la intención original, sin ser el resultado de una interpretación parcial de la misma.
- **Coherente:** no se encuentra en contradicción con el resto de requisitos.
- **Comprensible:** debe ser entendible por todos los agentes involucrados en diseño de la arquitectura. En este caso, solo el candidato tuvo acceso a estos requisitos, siendo también el autor de los mismos.

Un análisis del cumplimiento de los requisitos (véase sección 6 – Conclusiones) ha garantizado que estos han sido claros, coherentes e inequívocos, así como alineados con el resto de características enumeradas. Los requisitos se perfeccionaron durante la ejecución de las tareas técnicas y durante la integración de los diferentes módulos de la arquitectura.

### 3.1.1.1. Requisitos técnicos

En primer lugar, dado que se ha querido diseñar un sistema IoT, primero y, un servicio sobre el mismo, después, se comenzó con la recopilación de requisitos técnicos (no funcionales) para la arquitectura IoT.

En la siguiente tabla se muestran los requisitos técnicos capturados junto con una descripción de los mismos. Esta tabla es el resultado final de una serie de iteraciones (véase metodología en la sección 3.1.2).

**Tabla 3.1:** Requisitos técnicos de la arquitectura IoT

| Requisito                   | Descripción  |
|-----------------------------|--|
| Flexibilidad                | Debe tratarse de una solución modular, con la capacidad de adición de nuevos módulos y funcionalidades que pueden ir creciendo a lo largo del tiempo. Igualmente, debe ser flexible desde el punto de vista de despliegue en diferentes equipos a lo largo del continuo de computación <i>edge-to-cloud</i> .  |
| Portabilidad                | Cada uno de los componentes de la arquitectura debe ser lo suficientemente genérico como para ser fácilmente desplegados (transferidos) a diferentes entornos, casos de uso o aplicaciones. Esto debe ser conseguido a través de mecanismos de extensión bien documentados (como <i>plug-ins</i> , interfaces genéricas, clases abstractas y similares).                                 |
| Escalabilidad               | La arquitectura debe diseñarse de tal manera que los despliegues sean escalables, es decir que puedan crecer (en recursos, espacio de almacenamiento...) en función de las necesidades o de la gran cantidad de datos con los que pueda trabajar en el futuro.   |
| Conectividad                | Debe soportar diferentes protocolos de conexión desde el punto de vista IoT. Es decir, varias tecnologías de acceso deben estar cubiertas y validadas durante la tesis doctoral para considerar este requisito cumplido.   |
| Agilidad                    | Debe tratarse de una solución IoT agnóstica de sector y enfocada en un despliegue rápido, ágil y de bajo coste.  |
| Interoperabilidad           | La solución deber ser interoperable con el resto de las herramientas tecnológicas convivientes en el entorno de despliegue. Dada las infinitas posibles combinaciones, este requisito solo puede cumplirse a través de interfaces estandarizadas y bien documentadas, del mismo modo que con la utilización de protocolos de comunicación y modelos de datos conocidos e interoperables. |
| Multi instanciación         | Esta solución no debe diseñarse como un entorno único donde diferentes usuarios de diferentes entidades, con acceso a sus propios datos, convivirán. No se trata de un sistema <i>multi-tenancy</i> , si no que cada despliegue será único en entidades lógicas diferenciadas.   |
| Seguridad entre componentes | Los intercambios de información entre los diferentes componentes deben garantizar seguridad, de tal manera que se disponga de una conexión segura extremo a extremo. Esto es un factor clave para una arquitectura IoT de nueva generación. Los protocolos de comunicación deben garantizar  |

|                                  |  |
|----------------------------------|--|
|                                  | la integridad y la privacidad de los datos (esto puede resolverse, por ejemplo, utilizando intercambios de información que soporten TLS/SSL).  |
| Seguridad en el acceso           | El acceso a la solución debe restringirse únicamente a usuarios autorizados, en función de sus roles y funciones. Se establece la necesidad de perímetro y de acceso a la arquitectura desde todos los puntos de vista (uso, configuración, solicitud de datos, etc.).   |
| Usabilidad                       | El sistema debe ser usable por todos los perfiles a identificar, desde el usuario final al profesional de despliegue, mantenimiento o configuración. Al poder albergar una gran cantidad de datos de muchos dispositivos, una buena estructuración de la información y de los flujos de intercambio forma también parte de este requisito. |
| Configurabilidad                 | La solución debe poder ser configurable tanto a nivel de adquisición de datos, como de almacenamiento, de ejecución de servicios y de interfaz.  |
| Disponibilidad                   | Se trata de garantizar que la solución estará disponible (en funcionamiento y respondiendo acorde al comportamiento esperado) durante los tiempos habituales de operación. Este requisito debe cumplirse también para minimizar el tiempo de inactividad de la solución.   |
| Compatibilidad                   | La variedad de dispositivos, clientes de conexión, tipos de datos, librerías utilizables, etc. en el entorno IoT, garantizar compatibilidad (con varios sistemas operativos, navegadores, etc.) se antoja clave para asegurar una correcta adopción de la solución.  |
| Integridad                       | La confianza en que los datos obtenidos de diferentes fuentes IoT son auténticos (no inmutados), precisos y correctos. La integridad de los datos debe ser considerada en el diseño de todos los elementos que conformen la arquitectura.  |
| Cumplimiento de las regulaciones | La solución debe respetar las normas y leyes de instalación de prototipos en los escenarios de validación y debe asegurarse del cumplimiento de la normativa LOPD.   |

### 3.1.1.2. Requisitos funcionales

Los requisitos funcionales son aquellos que especifican el comportamiento de la solución ante un posible caso de uso. Modelan las funcionalidades durables esperadas por parte de la arquitectura en cuanto a almacenamiento, procesamiento

de la información, cualidades *middleware* [194] de la solución y otras que asumen los requisitos técnicos como una base de donde partir.

**Tabla 3.2:** Requisitos funcionales de la arquitectura IoT

| Requisito  | Descripción   |
|--|---|
| Fuentes de datos en tiempo real                        | Capacidad de adquisición de información desde diferentes fuentes en tiempo real a partir de técnicas de IoT.  |
| Observación de datos en tiempo real                    | El usuario debe poder consultar información siendo adquirida en tiempo real desde las diferentes fuentes.   |
| Capacidad de importación de información histórica      | Además de la obtención de datos periódicamente (o asíncronamente) de sensores u otras fuentes, la arquitectura debe permitir la carga de una gran cantidad de información ( <i>batch</i> ) para contribuir al histórico de algún índice de datos. Esta información puede ser cargada como datos agregados o primitivos, con la intención de ser usado por servicios de capas superiores y la fuente original puede provenir de archivos de todo tipo: Excel, CSV, txt, etc. siempre y cuando existan entidades (agentes) encargados de realizar esta importación. |
| Introducción manual de datos                           | La solución debe ser capaz de recoger los datos proporcionados manualmente, con el fin de alimentar la base de datos y, posteriormente, los servicios ya que en determinadas ocasiones no existirá la posibilidad de adquirir información de una manera automatizada.   |
| Almacenamiento a corto y largo plazo de la información | El sistema debe contar con los elementos y mecanismos necesarios para permitir el almacenamiento de los datos obtenidos por los requisitos funcionales anteriores de cara a dar un servicio al usuario final de la solución.  |
| Análisis de datos históricos                           | Se debe permitir una visión gráfica (en forma de evolución temporal) de la información relevante que proviene tanto de los datos originales como de los resultados de ejecución de los servicios.   |
| Instanciación de servicios                             | La solución debe estar orientada a una instalación ágil de servicios basada en una metodología clara.   |
| Ejecución automática periódica de servicios            | Los servicios instanciados deben ser capaces de: (1) obtener información de persistencia del sistema, (2) invocar funciones específicas para proveer una funcionalidad en particular, (3) proveer resultados y persistirlos para la utilización por parte del usuario o de otros servicios.   |

|   |  |
|---|--|
| Interoperabilidad sintáctica de los datos   | Deben acordarse mecanismos para garantizar la interoperabilidad sintáctica de la información para el correcto funcionamiento de los servicios como el uso de modelos de datos estandarizados   |
| Capacidad de entrenamiento de modelos ML y de ejecución de inferencia dentro de la arquitectura | Se trata de un requisito de la arquitectura el que, en caso de que se desee implementar servicios de aprendizaje automático (ML), éstos puedan incluirse de forma nativa en la arquitectura. La arquitectura debe estar preparada para este procesamiento y debe disponer de una metodología clara para la creación y uso de modelos ya que la naturaleza de éstos difiere de la de otros servicios.   |
| Visualización web de los resultados de servicios sobre los datos IoT                            | La solución debe incluir una interfaz de visualización por cada uno de los servicios que se desplieguen sobre la misma. Debe poseer, por tanto, de una metodología para la incorporación de estas interfaces de usuario. Además, cada una de ellas tiene que poder ser accesible por los navegadores más utilizados, basarse en estándares web (HTML, CSS, Javascript), debe ser claramente interpretable y legible desde varios tipos de dispositivos ( <i>smartphone, tablet...</i> ) y fácil de usar. |
| Administración centralizada de la solución  | La gestión de los usuarios que van a interactuar con la solución debe estar centralizada para facilitar la gestión y utilización de los servicios.   |

### 3.1.1.3. Requisitos de servicio

Por último, en la siguiente tabla se recogen los requisitos capturados relacionados con el servicio de cálculo y predicción de índices compuestos. En este sentido, esta actividad se encontró a caballo entre los objetivos de este capítulo y la sección 4. Muchos de los aspectos incluidos en la tabla fueron descubiertos tras realizar en análisis de la literatura de este tipo de cálculos (sección 4.1.2.2) y tras diseccionar las posibilidades de configuración y resultados de los métodos matemáticos necesarios. No obstante, es relevante incluirlos en este punto ya que algunas de estas observaciones tuvieron efecto directo a la hora del diseño de la arquitectura (por ejemplo, la necesidad de contar con un servidor externo para la normalización en el caso del despliegue en entornos comparativos – sección 4).

**Tabla 3.3:** Requisitos del servicio a ejecutar sobre la arquitectura IoT

| Requisito                  | Descripción  |
|----------------------------|--|
| Valor del índice compuesto | Se establece como requisito que el valor final del índice compuesto sea entre 0 y 1. |



|                                       |   |
|---------------------------------------|---|
| Diferentes orígenes de la información | El servicio de CI debe ser capaz de aceptar datos (indicadores) capturados de diferentes formas (sensores, APIs, desde ficheros, datos insertados a mano, etc.).  |
| Flexibilidad en la estructura         | Un índice compuesto puede originarse desde una cantidad muy diversa de indicadores iniciales (mayor de 2). El servicio a desarrollar debería poder aceptar cualquier cantidad de estos indicadores iniciales (dependiendo de la aplicación y el despliegue). Además, esta estructura debería poder ser cambiante dinámicamente y configurable por el usuario. Este mismo razonamiento aplica a las sucesivas divisiones de subíndices.  |
| Selección de métodos de cálculo       | Los diferentes métodos matemáticos a utilizar tienen que ser configurables igualmente por el usuario. Esto incluye la indicación de la importancia relativa de cada uno de los indicadores iniciales.   |
| Establecimiento de reglas             | La arquitectura debe permitir que el servicio pueda aplicar reglas sobre los resultados de la ejecución.  |
| Recomendaciones                       | La arquitectura debe permitir la introducción dinámica de recomendaciones. El objetivo es poder insertar una serie de sugerencias de actuación para la mejora de los resultados del índice compuesto.   |
| Informe                               | Debe existir la posibilidad de exportar los resultados de una ejecución del índice compuesto en un formato de informe.  |
| Opciones de visualización             | La visualización del servicio particular de cálculo de indicadores compuestos debe incluir, como mínimo, los siguientes elementos: <ul style="list-style-type: none"> <li>● Gráfico tipo velocímetro expresando el último valor calculado del CI.</li> <li>● Evolución temporal de los valores de los índices y los subíndices.</li> <li>● Configuración de métodos matemáticos y de estructura de cálculo.</li> <li>● Comparación histórica de los valores.</li> <li>● Observación del último valor de cada uno de los indicadores.</li> <li>● Comparación entre entidades.</li> <li>● Observación de los datos predichos.</li> <li>● Carga de valores por defecto.</li> </ul> |

### 3.1.2. Metodología

En este subcapítulo se expone la metodología que se ha seguido para el diseño de la arquitectura IoT que compone el corazón de la tesis doctoral. Se ha excluido de esta explicación la metodología del servicio de índices compuestos y el proceso de orientación hacia explotación de producto ya que éstos son objeto de estudio en otras secciones del manuscrito.

Los requerimientos anteriores fueron obtenidos a través de un proceso iterativo de estudio del estado del arte (qué existe, qué características tienen las arquitecturas IoT de nueva generación, qué se necesita para ofrecer un servicio de cálculo de CIs a partir de técnicas de IoT...) y análisis de las necesidades globales de la tecnología en la tesis doctoral. El diseño de la arquitectura siguió un percurso equivalente, aunque partiendo de la base establecida por dichos requisitos. A nivel metodológico, ambas actividades pueden explicarse de forma análoga.

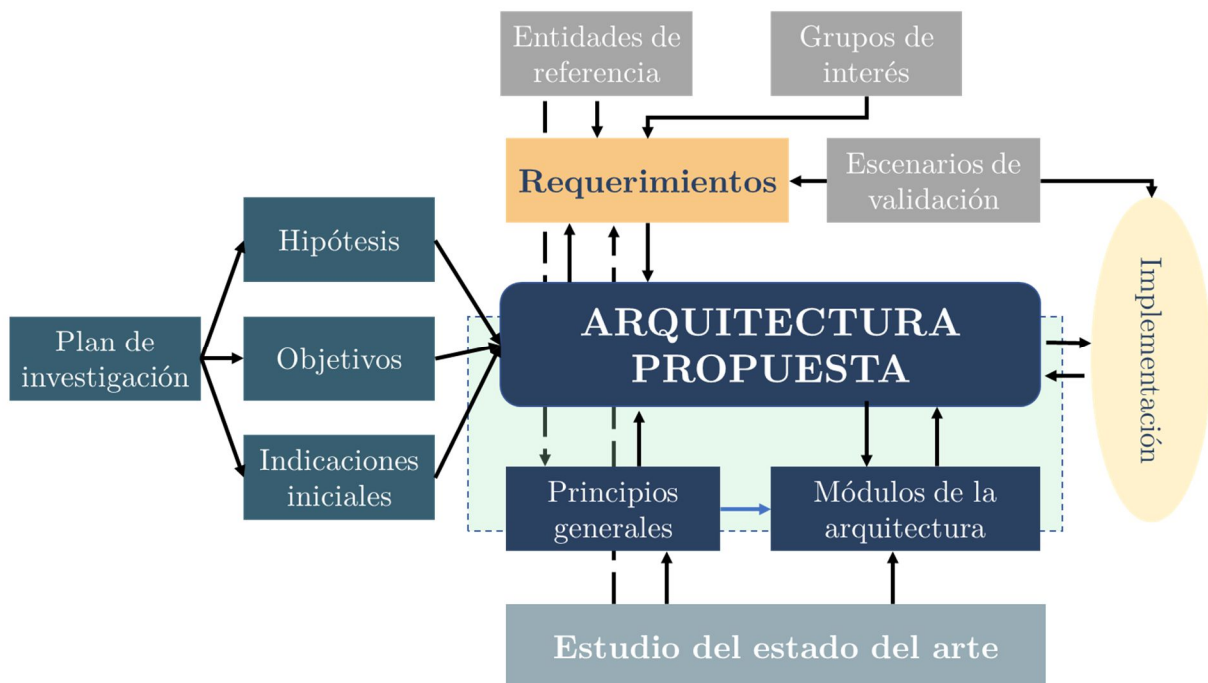
Para comenzar, el diseño de la arquitectura se basó en la propia definición de la hipótesis de la tesis doctoral y en los objetivos de la misma. Desde la generación del plan de investigación, una serie de líneas maestras fueron esbozadas y sirvieron como esqueleto para el diseño de dicha arquitectura. Por otro lado, las consideraciones de las arquitecturas del IoT de nueva generación han alimentado continuamente este diseño. Por ejemplo, entidades como AIOTI [195] o NGIoT [196] publican periódicamente estudios y *whitepapers* sobre dichas consideraciones, proponiendo opciones de diseño, tecnologías predominantes, innovaciones, etc. Por otro lado, entendiéndolo que esta arquitectura está claramente orientada al cálculo de índices compuestos, se aprovechó la participación del doctorando en proyectos de investigación (con acceso a diferentes perfiles de entidades socias de los consorcios) para capturar las necesidades de varios grupos de interés que podrían utilizar este tipo de soluciones.

Segundo, se realizó un concienzudo análisis de las arquitecturas IoT clásicas e IoT de nueva generación (véase sección 2.2), lo cual contribuyó en gran medida a la determinación, primero, de los principios fundamentales de la arquitectura y, después, a las recomendaciones de implementación tecnológica. A través de este estudio, se perfiló la arquitectura modular, asimilándola a un diseño por “capas” o por niveles, alineado con las referencias más importantes.

En tercer lugar, dichos bloques tecnológicos de la solución de la tesis han tenido una relación de dependencia mutua con la propia definición de la arquitectura, ya que uno ha servido para dar forma al otro. Los diferentes módulos se han guiado tanto por los principios de la arquitectura como por los objetivos y particularidades de cada tecnología escogida. En las actividades posteriores de la tesis (diseño y desarrollo del servicio y validación del sistema en escenarios reales) ha sido necesario alinearse con la arquitectura, por lo que se establecieron y cumplieron ciertas reglas de interacción.

Cuarto, teniendo en cuenta que la solución debe ser fácil de usar, eficaz y sencilla al mismo tiempo, gran parte de su definición se basó en permitir el despliegue exitoso de aplicaciones (escenarios de validación en el mundo real). Esto se logró mapeando las necesidades de los diferentes perfiles usuarios con los conceptos abstractos y los módulos concretos de la arquitectura. Estos usuarios poseían perspectivas diferentes y van desde arquitectos software, desarrolladores, ingenieros de producción, instaladores o usuarios intermedios hasta usuarios finales y agentes implicados en la toma de decisiones.

No obstante, estos cuatro pasos no siguieron una distribución lineal y en cascada. Debido, por una parte, a la naturaleza asíncrona de la participación en proyectos de investigación (avanzando a diferentes velocidades) y, por otra, a las dependencias mutuas de las diferentes actividades implicadas (captura de requisitos, diseño, selección de tecnologías), el flujo real de la metodología resultó en la siguiente figura:



**Figura 3.2:** Metodología de diseño de la arquitectura

Para mantener una trazabilidad en todo este proceso, el candidato hizo uso de la versión gratuita del software JIRA [197]. Los diferentes requisitos fueron modificados durante gran parte de la duración de la fase de investigación.

Todas las consideraciones anteriores llevaron al doctorando a proponer la versión final de la arquitectura resumida en la subsección 3.2. Además, el resto de la sección 3 tiene como objetivo describir sucintamente la función principal de cada módulo, las tecnologías seleccionadas y los detalles de desarrollo de este trabajo.

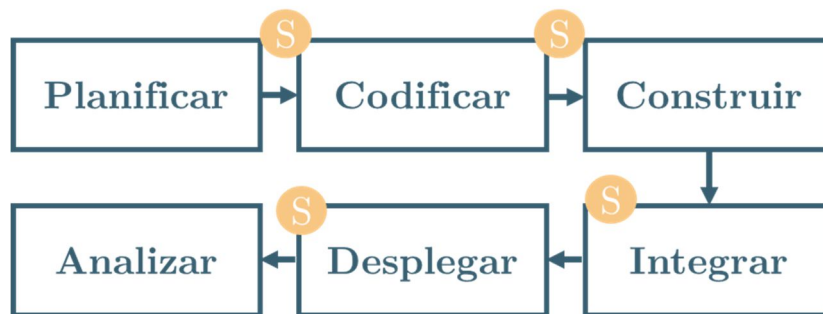
No obstante, previamente a entrar en materia tecnológica, conviene destacar el aspecto metodológico del desarrollo de software. La tesis doctoral propuesta por

el candidato ha tenido un mayor componente de integración de tecnologías open source; seguido a cierta distancia por la realización de desarrollos a medida, mejoras de los módulos existente y la definición de decisiones de interconexión entre todos los elementos anteriores.

Para la confección de todo lo anterior, se ha seguido una metodología denominada DevSecOps [168], catalogada dentro de la familia CI/CD [198], orientada a la inclusión de consideraciones y pruebas de seguridad dentro de un modelo de desarrollo e integración continua. Se eligió seguir esta metodología ya que, precisamente por la naturaleza asíncrona iterativa del diseño, permitía la creación de prototipos intermedios (MVPs del inglés *Minimum Valuable Products*) que podían comenzar a validarse en los escenarios expuestos en la Sección 5.

Otro de los motivos para escoger esta metodología fue su recomendación en escenarios avanzados de IoT por parte de entidades como ENISA [199], alegando que en estos entornos aspectos como la actualización del software, la privacidad de los datos de sensores y otro tipo de amenazas requieren una atención especial a la seguridad desde fases tempranas del diseño.

En la siguiente figura se muestra un resumen de esta metodología, que ha sido aplicada durante la tesis para el diseño y desarrollo de cada módulo que compone la arquitectura:



Durante la fase de planificación (selección de tecnología, protocolos de comunicación y composición global del módulo), se aseguró de comprobar las amenazas más habituales y su cobertura [200]. Durante la codificación, se utilizaron extensiones en los entornos de desarrollo (IDE) para realizar ciertas comprobaciones de seguridad. Durante la construcción de las imágenes y compilación de los diferentes desarrollos (*build*), se estudiaron las dependencias con software de terceros y se aplicaron medidas si fue necesario. A la hora de la integración entre módulos, pruebas de carga con tests de seguridad se exploraron para todos los módulos.

Otro componente importante dentro de esta metodología fue el mantenimiento de un versionado de cada versión de los diferentes artefactos. Esto se realizó tanto dentro de los proyectos de investigación como durante los desarrollos independientes.

## 3.2. Visión general

En esta subsección se ofrece una visión general de la arquitectura IoT diseñada en la tesis doctoral. En particular se ofrece información sobre los principios de diseño que han regido el funcionamiento global del sistema y la estructura modular de la misma. Las indicaciones aportadas a continuación provienen directamente de los capítulos anteriores (estudio del estado del arte y requisitos).

### 3.2.1. Principios de diseño

Los principios de diseño de la arquitectura han sido aquellos en los que se ha basado la implementación de todos los módulos, así como su despliegue. Este tipo de consideraciones son clave a la hora de proceder a la concepción del sistema global, y suelen conformar las primeras acciones en esta tarea. Se trata de decisiones tecnológicas tomadas que guiarán el espíritu de toda la arquitectura.

Estos principios provienen directamente de la interpretación de los requisitos técnicos fundamentales que, en combinación con lo analizado en el estado del arte de arquitecturas IoT, hicieron al candidato plantearse una serie de preguntas sobre las que tomar decisiones de diseño global. Para resolver dichas cuestiones, también se debió analizar los requisitos funcionales y de servicio, sirviendo para despejar incógnitas sobre la dependencia mutua entre módulos, como se explica posteriormente.

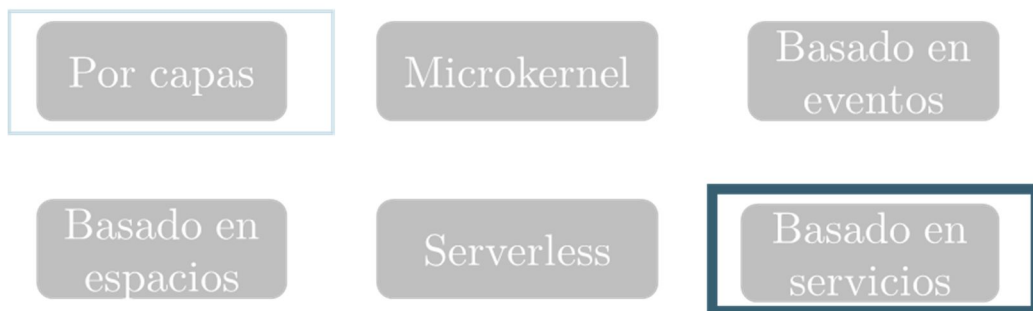
La subsección se estructura siguiendo precisamente dichos requisitos fundamentales (véase sección 3.1.1.1), que necesitaron una reflexión, consideraciones y decisiones justificadas, como las que se proveen en las siguientes páginas:

#### ***Selección de paradigma:***

Hoy en día, las arquitecturas del IoT son cada vez más complejas: se incluyen continuamente nuevas funciones, se amplían las cargas de trabajo y aumenta el número de componentes. El análisis de los paradigmas de arquitectura realizado en la sección 2.2 es un buen punto de partida para hacer frente a esta complejidad, apoyando la organización de los componentes del sistema [201]. No obstante, como se ha observado en ese mismo capítulo, no todos los paradigmas y patrones trasladan bien a implementaciones reales, siendo demasiado abstractas, planteando problemas que redundan en su descarte en favor de soluciones comerciales ya existentes.

Comenzando con los patrones de diseño de arquitectura más globales, los más relevantes son los siguientes [202] (que en algunos casos pueden combinarse): por capas, *microkernel*, basado en eventos, basados en espacios, sistemas *serverless* (sin necesidad de servidor) y basados en servicios. Dado que los requisitos de las soluciones IoT evolucionan hacia la descentralización [203], las arquitecturas de *microkernel* que se basan en la existencia de un elemento central sobre el que se

adhieren diferentes funcionalidades (en forma de *plugins*) parece no ser la opción más recomendada para la arquitectura de esta tesis. Por otro lado, los enfoques basados en espacios tienen como (casi) único objetivo crear entornos orientados a la escalabilidad de las bases de datos, descartando otros factores relevantes en el IoT como la modularidad, la separación entre servicios o la reutilización. La tecnología sin servidor (*serverless*) ha ganado cada vez más adeptos durante los últimos años en el ámbito *cloud computing* (computación en la nube), ya que permite desacoplar con éxito el mantenimiento de la infraestructura del servidor del desarrollo de aplicaciones sobre la misma. De hecho, su principal valor se basa en conseguir esta disociación, olvidando elementos relevantes en el caso de esta tesis como agilidad, portabilidad o multi instanciación. Por tanto, no parece la mejor decisión a explorar. Los despliegues de IoT como los enunciados como objetivo para esta tesis (lejos del IoT entendido por las grandes operadoras de telecomunicaciones [204]) tienden a mantener una “esencia de infraestructura controlada”, que sigue requiriendo mantener al ser humano en la mayoría de las fases del despliegue (incluido el diseño de la estructura del servidor, el mantenimiento, el acceso, adición de nuevas fuentes de datos, etc.). Por todo lo anterior, el último paradigma global de arquitectura (basado en servicios), promete ser el candidato más adecuado a explorar. Este paradigma parece encajar perfectamente, ya que la alta modularidad permite crear una gran variedad de cadenas de aplicaciones, reutilizando instancias de los mismos “servicios”, un aspecto realmente interesante a considerar en el diseño de esta arquitectura pensando en la combinación de servicios de cálculo, entrenamiento e inferencia (véase 3.4).



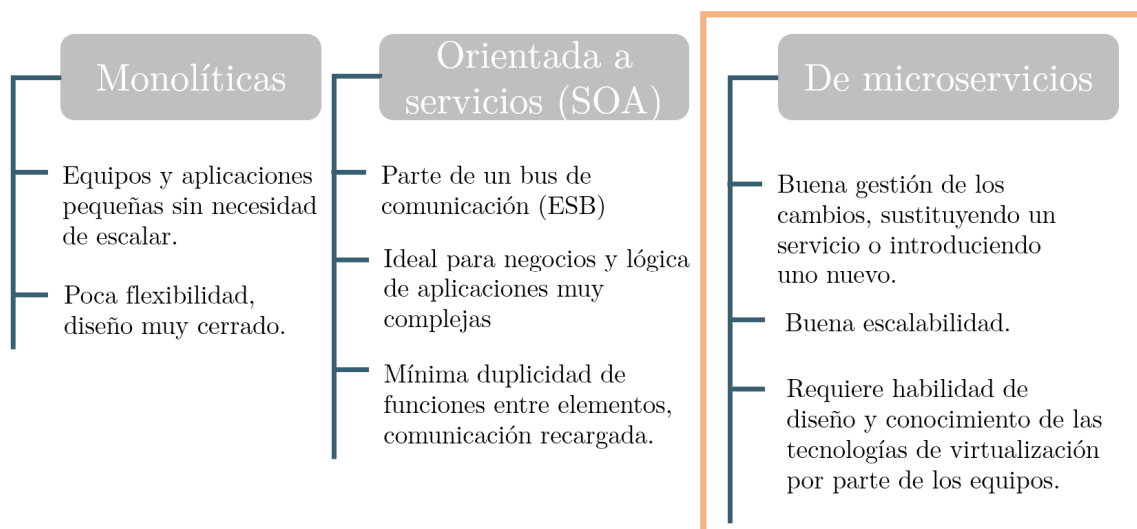
**Figura 3.4:** Patrones de diseño de arquitecturas (remarcado el elegido en la tesis)

Las arquitecturas basadas en servicios pueden, a su vez, dividirse en: (i) arquitecturas monolíticas, en la que sus componentes forman un programa o plataforma unificada e indivisible; (ii) arquitectura orientada a servicios (SOA del inglés *Service Oriented Architecture*), que la disgrega en un conjunto de módulos, que se mantienen de forma independiente y trabajan juntos mediante una capa de agregación (*middleware* de mensajería también conocido como bus empresarial); y (iii) arquitectura de microservicios, compuesta por servicios independientes ligeramente acoplados, que se comunican entre sí normalmente a través de API o de mecanismos claramente definidos.

Para elegir entre las tres diferentes perspectivas, se tuvieron en cuenta los siguientes factores, inferidos a partir de los requisitos técnicos de flexibilidad, escalabilidad y portabilidad y los requisitos funcionales de instanciación de servicios y administración centralizada de la solución:

- Dado el número de tecnologías y funcionalidades que podrían utilizarse (instanciación de servicios de diferente índole), mantener el software en módulos independientes que puedan interconectarse posteriormente facilita su mantenimiento y su uso sólo cuando sea necesario.
- La solución debe estar preparada para entornos de computación de borde (*edge computing*), donde las características de los equipos de computación son diversas. Por tanto, es importante desacoplar estos servicios, al máximo posible, del entorno donde van a ser ejecutados.
- La orquestación de servicios ocupa un lugar necesario pero no destacado dentro de la arquitectura, primando la facilidad de despliegue y rapidez de configuración orientados a un servicio de cálculo y predicción de índices compuestos.

Reflexionando sobre lo anterior, se decidió optar por una estructura híbrida, a caballo entre el subparadigma de microservicios y esencia de diseño por capas (planteamiento global), ya que, como en cualquier arquitectura IoT, debe existir una serie de funcionalidades correspondientes asimilables a un modelo por capas OSI simplificado [205]. Por tanto, no existe un desacoplamiento absoluto entre todos los módulos, ya que se debe garantizar un flujo bidireccional vertical (*upstream* y *downstream*) característico en estos despliegues. Por ejemplo, para transmitir los datos de sensores a un sistema remoto persistente, deben intervenir varios elementos verticales y planos (gestión de datos, red, aplicación y otros). Además, cualquier proceso en el que intervengan varios elementos requiere que se incluyan también mecanismos de gestión y configuración. Otros motivos considerados se aportan en la siguiente imagen:



**Figura 3.5:** Subparadigmas basados en servicios (remarcado el elegido en la tesis)

No convino la adopción de un enfoque monolítico, ya que se desea dar la opción de introducir nuevos servicios y crecer sobre la arquitectura básica y la opción SOA se descartó ya que un bus de mensajería se antojó un inconveniente innecesario (funcional y computacionalmente) que dificultaría una descentralización de los despliegues en caso de ser aplicable. Además, aunque estos dos enfoques permiten reducir la cantidad de gestiones (sobre diferentes módulos y servicios) al proponer una sola entidad, la decisión tomada compensa esta desventaja por los motivos expuestos anteriormente.

### **Flexibilidad y escalabilidad:**

Es un requisito fundamental de la arquitectura a proponer el proveer flexibilidad (adición y sustracción de elementos) y escalabilidad (capacidad de crecimiento). Se trata de aspectos de suma importancia, ya que conseguirlos no solo afecta a los servicios a desplegar, sino también la infraestructura para ejecutarlas, así como la gestión óptima de esos recursos.

Se partió de los principios:

- El sistema debe ser modular, con diferentes elementos que puedan añadirse en tiempo de ejecución y sin la necesidad de rediseñar la arquitectura.
- La solución deberá poder desplegarse en varios entornos de computación (*edge*, *cloud*, dispositivos con bajos recursos).
- Diferentes módulos del sistema deberían poder residir en diferentes puntos del continuo de computación (*computing continuum*).
- Se busca un despliegue ágil y rápido.
- Como se indica en la sección 2.5.3, la metodología DevSecOps requiere realizar pruebas en diferentes momentos del desarrollo y la producción ágil de MVPs testeables en diferentes entornos.
- La seguridad entre componentes es uno de los requisitos fundamentales expresados para la arquitectura.
- Pensando en la escalabilidad de la solución propuesta, para cubrir el caso donde el *edge computing* y el aprendizaje federado jugaran un papel importante, la arquitectura debería contemplar la existencia de decenas/centenares de nodos de computación/procesamiento participando en el sistema y la ejecución del servicio.
- El aislamiento entre sistemas operativos, librerías y otros software necesarios para ejecutar cada módulo es un requisito relevante, ya que la arquitectura debería permitir la implementación de servicios desarrollados con diferentes tecnologías.

Reflexionando sobre lo anterior, resultó claro que debería optarse en la arquitectura por una aproximación de virtualización de todos los componentes. Para esto, existen varias alternativas. La opción de basar la arquitectura en



máquinas virtuales se contempló, pero se descartó inmediatamente, ya que la escalabilidad estaba comprometida y el tiempo y facilidad de despliegue no podían estar a la altura de lo requerido. Por tanto, debía elegirse la utilización de **contenedores**. Además de las ventajas estudiadas expuestas en la sección 2.5.1, existen referencias apuntando a que, hoy en día, la mayoría de propuestas para despliegues IoT modernos adoptan esta aproximación [206]. Al ejecutarse en contenedores dedicados, los componentes están separados del sistema anfitrión (*host*) subyacente. Este paradigma de diseño permite la flexibilidad en la comunicación interna de los componentes y su implementación, sin comprometer la estandarización, la documentación y la seguridad de los canales de comunicación orientados al consumidor (por ejemplo, las API REST, los flujos, etc.).

A la hora de escoger la herramienta de containerización, las alternativas mencionadas en la sección 2.5.1 (RKT, LXC/LXD, Apache Mesos) fueron descartadas debido a su baja adopción, limitado soporte y reducida existencia de imágenes preconfiguradas disponibles en repositorios públicos. La opción a escoger, por tanto, fue **Docker** (según [207], más de 3.5 millones de aplicaciones son ejecutadas hoy en día utilizando esta herramienta).

#### ***Orientación de despliegue: cumplimiento de requisito de agilidad:***

A pesar de que el diseño de la arquitectura contempla diferentes perspectivas de despliegue (en nodos más potentes, en entornos remotos, etc.) gracias a la opción de containerización recién expuesta, resultó conveniente analizar cuál va a ser la opción prioritaria y por qué, ya que este hecho tiene repercusiones a varios niveles de la arquitectura.

De hecho, la mayoría de las soluciones actuales son demasiado especializadas, ya que se ejecutan sólo en un conjunto concreto de hardware y/o firmware, o demasiado genéricas, como los típicos servicios puramente *cloud* que no admiten ningún tipo de interacción con el hardware. La arquitectura propuesta para esta solución no se encuadra en ninguno de los dos extremos, alineado con los requisitos de agilidad y portabilidad.

Para ello, es interesante repasar los conceptos de despliegue *cloud* y en servidores propios (*on-premise*) para entender la mejor aproximación a tomar [208]:

- La opción *on-premise*, en la cual se opera sobre equipamiento propio, el control es la principal ventaja. Se puede escoger exactamente qué sistema (operativo, capacidades, etc.) y dónde se sitúa, teniendo la flexibilidad (requerimiento técnico) para crear despliegues ad-hoc y así diseñar topologías, redes internas y otras características de interés. Esta opción cobra mayor relevancia al plantear una arquitectura compatible con el *edge computing* y la descentralización. En cuanto a las desventajas, la principal es el coste. Un equipamiento ha de comprarse (si es nuevo), ha de mantenerse, se le ha de dotar de la suficiente conectividad y seguridad, ha de actualizarse su *firmware*; y por otro lado deben considerarse los costes de electricidad, espacio

físico, ventilación, etc. Si el sistema escala muy rápidamente, escoger la opción *on-premise* podría llevar a un incremento de costes inasumible.

- La alternativa *cloud*, por su parte, abstrae la mayoría de los costes recién indicados, convirtiéndolos en tasas mensuales o anuales de alquiler de recursos computacionales a un proveedor de *cloud* (por ejemplo Amazon, Google, Microsoft), normalmente en función del uso del equipamiento. Este equipamiento (virtualizado) no es controlable y, en caso de necesitar un acceso con características razonables de administración de sistemas, el precio se dispara [209]. Como se ha comentado, los despliegues IoT, por su naturaleza, requieren gran parte de configuración y acceso al sistema, especialmente considerando la conexión de diferentes fuentes de datos, sensores y la introducción del *edge computing*. Además, requiere un conocimiento muy concreto sobre la cantidad de datos y la habilidad de los usuarios de desarrollo de ajustarse a unos patrones de comunicación, ya que un servicio con peticiones excesivas o una recepción de datos inusualmente grande podría contribuir a incrementar este coste. Por otro lado, ofrece claras ventajas de crecimiento y escalabilidad, igual que flexibilidad desde el punto de vista de “acceso -virtual- a los recursos” de manera ubicua.

Como se ve, ambas opciones ofrecen ventajas y desventajas considerando los requisitos planteados para la arquitectura.

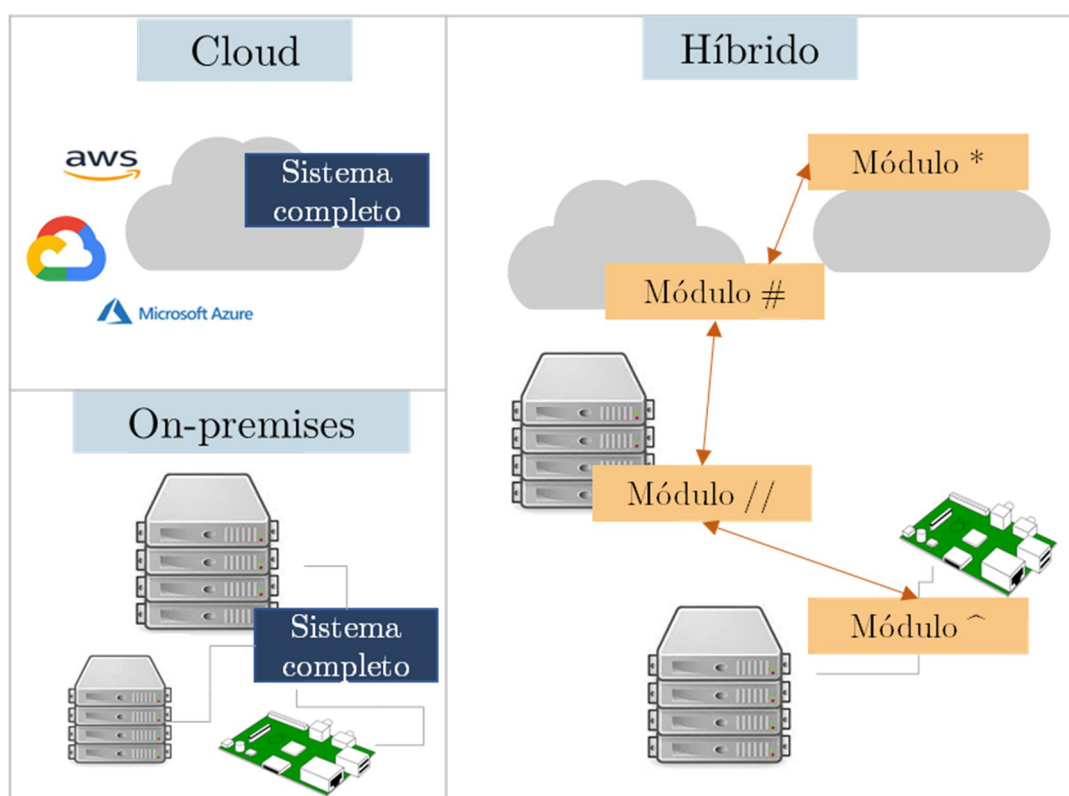
La idea es diseñar una arquitectura lo suficientemente flexible como para ser válida en cualquier escenario de crecimiento, volumen de fuentes de datos, requerimientos de computación, predicción... dentro de los límites del servicio principal deseado (cálculo de CIs). Siguiendo este razonamiento, existe la posibilidad de una tercera vía: el **modelo híbrido**. No es necesario decantarse al cien por cien por una sola de las opciones anteriores. La solución híbrida sería aquella en la que una organización utiliza una combinación de recursos locales y en la nube. Puede ofrecer flexibilidad tanto al inicio (diseño del despliegue y topología) como durante la vida activa del sistema (podría moverse parte de la carga de trabajo de un entorno a otro de forma dinámica), ajustando a las necesidades y coste. Otra de las ventajas de una solución híbrida sería poder ajustar los despliegues a las consideraciones de privacidad y seguridad requeridas, más fácilmente controlables en entornos de computación local.

Por tanto, se trabajó en encontrar un **modelo híbrido** partiendo de las siguientes premisas.

1. En primer lugar, los servicios sobre despliegues IoT tienen una marcada naturaleza de control, por tanto debe considerarse la opción *on-premise* como la perspectiva por defecto.

2. Se ha indicado la intención de convertir la solución como un sistema orientado al *edge computing*, lo cual refuerza la idea expuesta en el punto anterior.
3. Algunas de las funciones que aportan ciertos módulos de la arquitectura han de ser ejecutadas en entornos locales. Por ejemplo, el acceso a datos de sensores, en particular el despliegue de (la gran mayoría) de conectores de datos, siempre que la fuente sea un sensor (o asimilable), requerirá de equipamiento local para poder ejecutarse.
4. Como se verá más adelante (esta vuelta atrás fue fruto del proceso iterativo mencionado en la sección 3.1.2), la solución propuesta hace un intensivo uso de herramientas del sistema y de la configuración a través de consola, por tanto, se ha de garantizar unos altos niveles de acceso de permisos de administración. En estas circunstancias, la opción *on-premise* suele ser preferible.

Al margen de estas consideraciones, lo que se propuso para la arquitectura, aprovechando el requisito de diseño modular y basado en contenedores, fue el ceder la decisión del uso de *cloud* u *on-premise* glandularmente por módulo. Es decir, en lugar de seleccionar una u otra opción, el elegir alojar unos módulos u otros:



**Figura 3.6:** Perspectiva híbrida de despliegue de la solución

Atacar el desarrollo de la arquitectura bajo esta premisa implica conocer que la mayoría de los módulos deberían poder desplegarse tanto en la nube como en las

instalaciones sin afectar a las características de la plataforma (de nuevo, relevante la decisión de utilizar containerización mediante Docker).

En cuanto a la decisión de si adoptar, para cada despliegue, un entorno híbrido y, en caso de hacerlo, qué módulos, se debería realizar en función de varias condiciones.

- Capacidad financiera: la posibilidad de invertir en nuevos recursos propios o en alquiler *cloud*, considerando infraestructura, licencias software, soporte...)
- Tasa de crecimiento (volumen de datos, número de nuevas fuentes de datos añadidas, número de nuevos servicios a ejecutar...).
- Conocimiento técnico por parte de la entidad adoptante, tanto de la gestión de sistemas locales como de *cloud computing*.
- Necesidades de customización y de administración del sistema con permisos suficientes.
- Integraciones esperadas con otros sistemas externos.

### ***Interoperabilidad:***

Debido a que los diferentes elementos de la arquitectura tendrán una naturaleza aislada y modular, y visto que pueden residir incluso en diferentes localizaciones físicas y lógicas, se debe llegar a una serie de acuerdos para asegurar el entendimiento entre ellos y con otros sistemas externos.

Esta serie de acuerdos son lo que se denomina mecanismos de interoperabilidad [210] Para la solución propuesta se ha determinado que los dos mecanismos de interoperabilidad a emplear deben ser:

- El seguimiento del estándar de modelos de datos NGSII [211]. Propuesto por FIWARE, este estándar permitirá que la solución trate de forma agnóstica los datos independientemente de su modelo de llegada inicial. Además, se ha escogido por su capacidad de extensibilidad, permitiendo definir modelos nuevos personalizados, contando con la comunidad para ser validados y consolidar los esquemas para su generación. Se trata de un formato muy utilizado en la comunidad de investigación IoT [212]. En concreto, para el servicio de cálculo de CIs se ha diseñado un modelo de datos customizado siguiendo este estándar (véase sección 3.3.1). Además, esta decisión también permite acercar el requisito técnico expresado de portabilidad, haciendo que los desarrolladores de servicios puedan centrarse en estos formatos como entrada y salida sin preocuparse de analizar las fuentes iniciales.
- La comunicación entre módulos a través de interfaces estandarizadas. Se ha determinado que los módulos sean capaces de comunicarse preferentemente a través de APIs REST sobre el protocolo de transporte TCP y aplicación HTTP.

***Multi instanciación:***

Como requisito expresado para la confección técnica de la solución se encuentra la multi instanciación [213]. Conviene hacer un breve repaso de sus diferencias con la multi-tenancia para justificar esta decisión y especificar en qué se traslada en el diseño de la arquitectura.

- La multi-tenancia se considera una característica de aquel sistema software que, con una sola instancia (despliegue) puede servir a múltiples clientes. En la práctica, se trata de una aplicación desplegada en un entorno y a la que pueden acceder (y contiene datos de) múltiples clientes (conviene diferenciar aquí usuarios de clientes), separando la información de ellos de una forma clara y específica, compartida con los mismos. Cada uno de estos clientes es un *tenant* de la solución, capaz de configurar algunas partes de su usabilidad, pero sin poder de decisión ni modificación de la estructura o el código de la aplicación.
- En cambio, una aplicación multi instancia es aquella donde cada cliente de un sistema ejecuta su propia instancia de la aplicación, teniendo la capacidad de saber que sus datos no residen en el mismo espacio que el de otros clientes y pudiendo realizar modificaciones de mayor alcance sobre la misma.

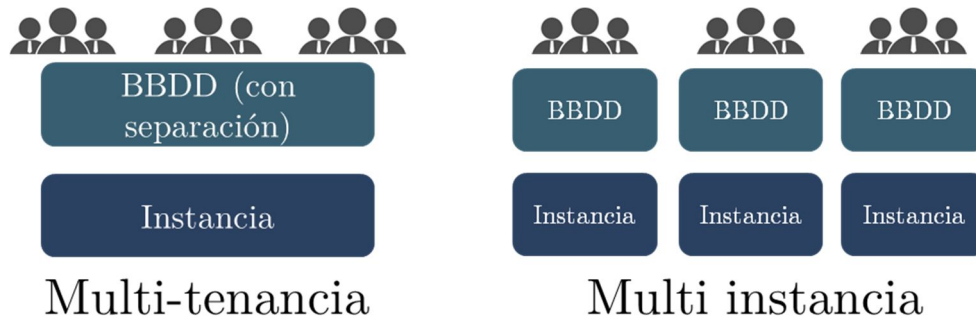
Mientras que con la primera se aprovechan los principios de reducción de costes para el desarrollador de la solución a partir de los principios de eficiencia en uso de recursos y economía de escala, y resulta más cómoda y ágil la actualización y mejora del sistema a muchos clientes a la vez, existen muchas desventajas relacionadas con el punto de fallo único. Si existe algún problema en la instancia, el servicio deja de ser efectivo para todos los clientes, siendo difícil el aislamiento y la personalización. Por último, sus propias características de actualización pueden no ser deseables caso por caso.

Al contrario, adoptar un modelo multi instancia significa incrementar, por diseño, la seguridad y privacidad de cara al cliente/usuario. Aporta una mayor flexibilidad y capacidad de crecimiento de la solución y protege mucho mejor un tratamiento caso por caso de los problemas y actualizaciones. Aunque esto puede suponer un mayor gasto de recursos computacionales y humanos, para aquellos sistemas que no vayan a ser comercializados a una cantidad enorme de clientes, se recomienda esta aproximación.

En cuanto a la selección para la arquitectura de la tesis doctoral, el despliegue se realizará siguiendo los principios de multi instanciación. El servicio de cálculo de índices compuestos se encuentra originalmente pensado para su adopción por parte de un único cliente. Aunque se trata de una solución que contempla la instalación por parte de varios clientes (escenarios de comparación y ranking), estos no se plantean suficientemente relevantes como para un diseño único multi-tenancia. Además, una de las razones principales es que se ha demostrado que es una aproximación más segura y que protege mejor la privacidad de los datos. Teniendo

en cuenta los diferentes contextos en los que la solución está diseñada para ser validada (véase sección 5), tener un control total sobre cada instancia es una ventaja para ser más adaptable a las circunstancias de cada escenario.

La siguiente figura muestra de forma gráfica la diferencia técnica entre ambas aproximaciones:



**Figura 3.7:** Multi-tenancia versus multi instancia en un sistema software

Además, elegir la opción multi instancia se antoja una mejor de cara a un modelo de negocio (véase sección 6), ya que es más aceptable desde el punto de vista de una empresa tener los datos completamente separados de otros potenciales clientes.

### ***Conclusiones sobre los principios de diseño:***

Aunque este tipo de recomendaciones no suelen hacerse a la hora de especificar una arquitectura de referencia, al estar la solución de esta tesis doctoral aplicado a una aplicación en entornos reales, se procedió a concluir sobre las siguientes directrices de implementación, evitando la ambigüedad y facilitando su uso:

- Arquitectura por capas de microservicios.
- Modularización de elementos, comunicándose entre sí de forma estandarizada, preferiblemente a través de interfaces API REST.
- Utilización de Docker.
- Multi instanciación frente a multi-tenancia.
- Acuerdo en modelos de datos NGSI.
- Despliegue híbrido basado en una selección por módulo y por caso.

### 3.2.2. Estructura modular y vistas

Como se ha estudiado en la sección 2.2, una arquitectura en el campo del Internet de las Cosas (IoT) puede describirse como un modelo de uso y combinación que propone unas directrices para implementar un sistema IoT. Previa descripción de la primera aproximación al diseño de la arquitectura de esta tesis, conviene reflexionar sobre lo siguiente: el estándar ISO/IEC/IEEE 42010 [31], base sobre la que se construyen la mayoría de arquitecturas modernas, especifica que dicho modelo de uso debe responder a varios conceptos iniciales. Entre ellos, se encuentran las perspectivas, los puntos de vista o los usuarios. De cara a la arquitectura de este trabajo, los aspectos más relevantes son los grupos de interés y las vistas.

**Un grupo de interés** es aquel individuo, equipo u organización que tienen un interés en la arquitectura [214]. En la práctica, se traduce en la recopilación de todos aquellos actores que, de un modo u otro, hacen uso de la arquitectura o están interesados en intervenir en su diseño. Las buenas prácticas indican que todos ellos deben ser considerados a la hora de especificar la arquitectura. Cada uno de estos grupos de interés se caracteriza por una serie de preocupaciones (*concerns*) para con la arquitectura que, unidos, forman un punto de vista particular (*viewpoint*).

Por otro lado, el concepto de **vista** refleja las expresiones gráficas o esquemáticas de una arquitectura desde el punto de vista de los actores mencionados, indicando cómo cada uno de ellos plantea su interacción con la misma. Según realizaciones prácticas del estándar mencionado [215], las vistas más importantes a considerar en una arquitectura son la Funcional, la de Despliegue, la Informacional, la de Concurrencia y la Operacional.

Según lo estudiado en el estado del arte y analizando las decisiones de arquitectura tomadas hasta este punto, se consideró que aquellas relevantes para la tesis propuesta son las siguientes:

- *Vista Funcional*: describe los principales elementos funcionales del sistema, sus responsabilidades, interfaces e interacciones principales.
- *Vista Informacional*: presenta una perspectiva de alto nivel sobre la recogida, el procesamiento y el consumo de datos, estableciendo una suerte de flujo de la información considerando las acciones específicas que se realizarían sobre los mismos.
- *Vista de Despliegue*: representa cómo debe desplegarse la arquitectura, proponiendo un análisis topológico incluyendo nodos *hardware* y los módulos software de la misma.

En la siguiente tabla trata de resumirse el trabajo que se realizó para la identificación de los grupos de interés y las vistas, así como de su encaje en cada una de las vistas.

**Tabla 3.4:** Vistas y grupos de interés en la arquitectura

| Vista         | Grupo de interés                                      | Preocupación / punto de vista   |
|---------------|---|---|
| Funcional     | Desarrolladores y soporte                             | Estructuración de los módulos, extensibilidad, API, comunicación.                                   |
|               | Usuarios / entidad adoptante                          | Funcionalidades y características del sistema IoT.  |
| Informacional | Ingenieros de datos / profesionales TIC de la entidad | Entender los datos que vayan a procesarse, sabiendo qué formato esperar e interpretar               |
|               | Desarrolladores y soporte                             | Conversión de formatos en los conectores y en el desarrollo de las interfaces de comunicación.      |
| De despliegue | Administradores del sistema                           | Estudiar la topología del despliegue y seleccionar la aproximación híbrida más apropiada por módulo |
|               | Desarrolladores y soporte                             | Desarrollo de varias materializaciones de la arquitectura   |

El resultado del análisis que condujo a la tabla permitió la creación de dichas vistas, ayudando a los desarrollos posteriores documentados en el resto de la sección.

***Vista funcional:***

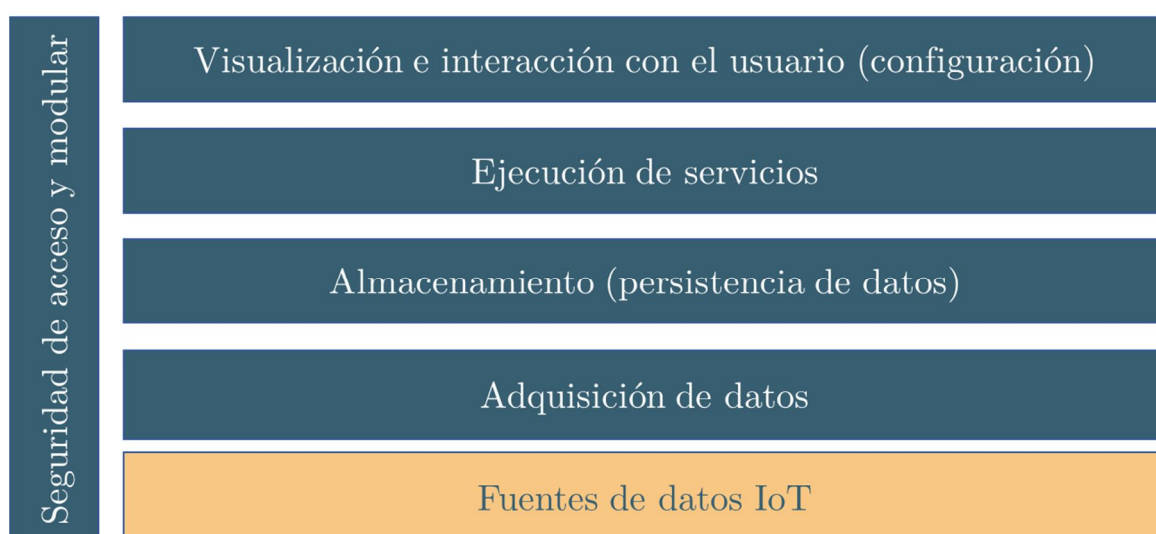
Uno de los principales requisitos de la arquitectura propuesta es su modularidad. La funcionalidad a proveer por parte de este sistema dependerá de la combinación de dichos módulos, cada uno de los cuales tendrá una “función específica”. Es, por tanto, claro que la vista funcional de la arquitectura en este caso está directamente relacionada con dichos módulos.

Inspirándose en las arquitecturas de referencia más utilizadas en la actualidad en entornos IoT (véase sección 2.2 para más detalles), se ha propuesto un modelo de módulos asimilables a capas, partiendo del siguiente razonamiento:

En primer lugar, desde la perspectiva técnica del IoT, se antoja necesario recoger y preprocesar los datos. Después, es necesario alinearlos sintácticamente para que sean interoperables y explotados por servicios unificados. La visualización y el almacenamiento también son necesarios (al menos, en una expresión mínima) en cada sistema de servicios IoT [216].

En segundo lugar, de acuerdo con la naturaleza del servicio que se va a prestar (véase Sección 4), se deben incorporar una serie de características funcionales; (i) valores de previsión del indicador, (ii) almacenamiento a largo plazo para el entrenamiento de modelos predictivos y para el análisis estadístico, (iii) ejecución del cálculo del índice compuesto y (iv) mecanismos para organizar el flujo de datos y procesos.





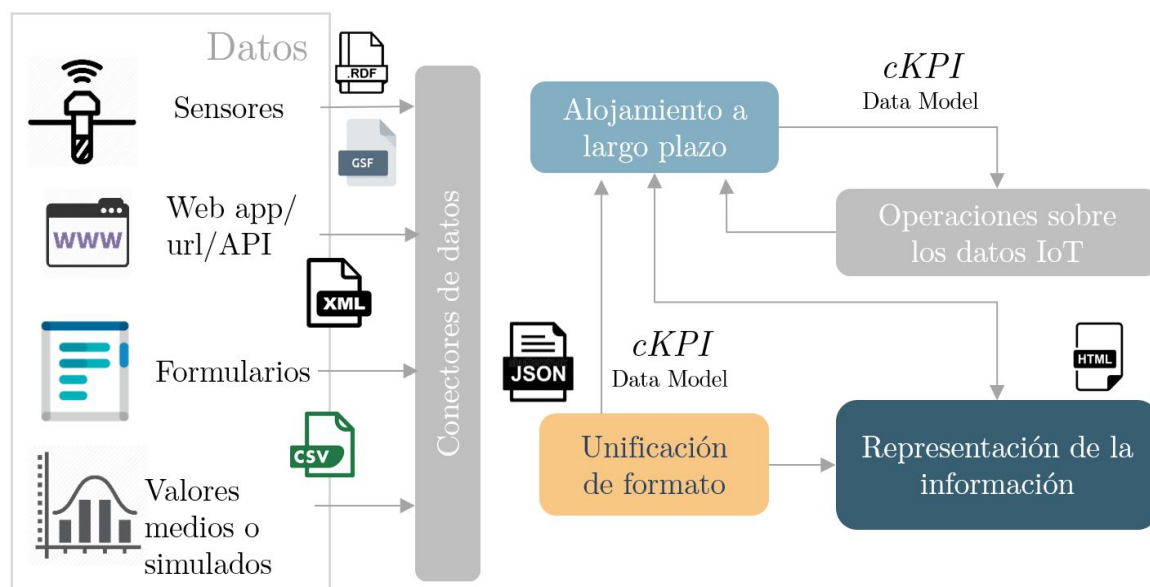
**Figura 3.8:** Diseño de la arquitectura (vista Funcional)

La estructura planteada responde a los siguientes módulos:

- **Módulo de adquisición de datos:** Componente clásico de IoT con la misión clara de obtener los datos de las fuentes originales y reunirlos en un elemento central asociados a “entidades”. Un concepto crucial en este módulo es la “consistencia del formato de los datos”, por ello este módulo debe proporcionar la base para el uso de interfaces de comunicación hacia capas superiores.
- **Módulo de almacenamiento:** El objetivo de este módulo es garantizar la persistencia de los datos que alimentan los servicios en la capa superior (en el caso de esta tesis, el indicador compuesto). Las condiciones más importantes que debe reunir este módulo son una alta disponibilidad, la containerización y el bajo consumo de recursos.
- **Módulo de ejecución de servicios:** Actúa como orquestador proporcionando la inteligencia al procedimiento. Como se ha comentado, una de las decisiones fue desplegar el servicio objetivo de esta tesis (cálculo y predicción de CI), así como otros adicionales, como elementos “independientes” usando contenedores Docker. Este módulo se encarga de coordinar el proceso para ejecutar rítmicamente dichos servicios, indicando con qué datos deben ser alimentados y otras variables provenientes de las preferencias del usuario.
- **Módulo de visualización:** Según las arquitecturas consultadas, separar la función de representación de datos, resultados e interacción con el usuario del resto de módulos es una buena práctica.
- **Módulo de seguridad y privacidad:** Un elemento clave en la arquitectura, puerta de entrada al sistema y aspecto transversal entre todos los módulos.

### Vista informacional

Esta vista (Informacional o de datos) permite entender el sistema a alto nivel desde el punto de vista de tratamiento de la información, omitiendo aquellos detalles técnicos normalmente necesarios para implementar el software diseñado (como la descripción de cada mensaje en un esquema de autenticación). Para ello, se utilizan abstracciones, indicando el rol de cada módulo pero centrándose en los cambios esperados y en las indicaciones oportunas para los grupos de interés involucrados.



**Figura 3.9:** Vista Informacional de la arquitectura

Como se observa, la vista provee información sobre los formatos de datos a lo largo del proceso, así como de las diferentes interacciones en relación a los mismos. El formato *cKPI* se revelará en la siguiente sub-sección.

### Vista de despliegue:

La vista de despliegue trata de representar la forma en que los nodos de (a lo largo del continuo de computación) y los módulos están dispersos por una infraestructura específica. Esta vista tendrá una materialización particular en cada escenario de aplicación.

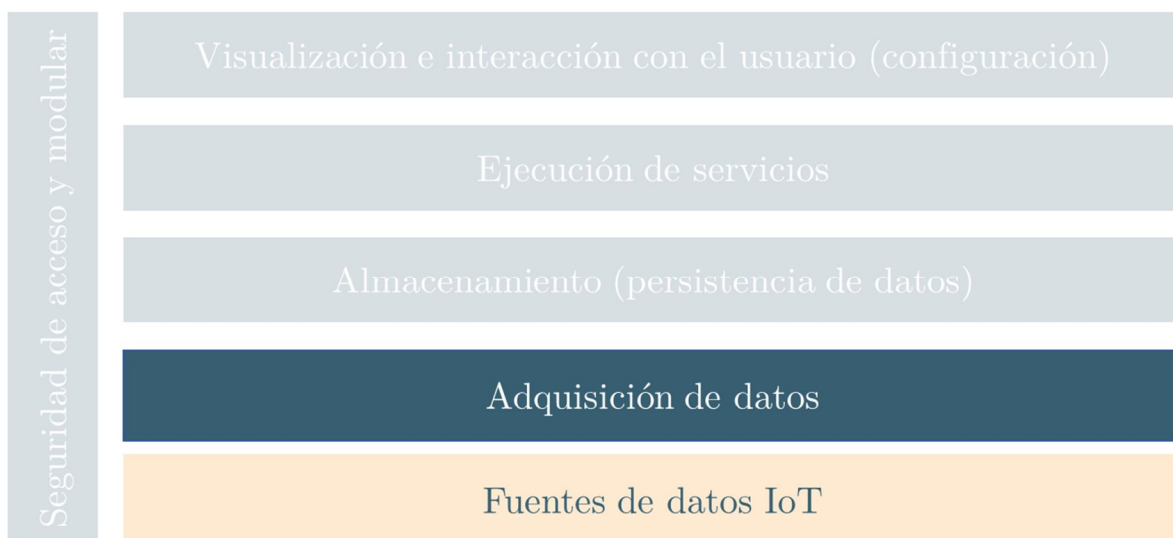
Para comprender qué representación gráfica se busca de cara a los actores interesados, puede consultarse la Figura 3.6. Además, en las secciones de validación de la arquitectura se muestra precisamente esta vista para ilustrar la instanciación de la arquitectura en cada caso (véase secciones 5.3, 5.4, 5.5, 5.6 y 5.7).

Esta vista dependerá de la topología de nodos (y sus características) de las que se disponga en cada escenario, siendo especialmente relevante en aquellos donde se prime la computación en borde.

### 3.3. Adquisición de datos

Este módulo es el encargado de actuar de puerta de entrada a la solución de los datos provenientes de fuentes IoT (y otras). Se trata del elemento que captura información del mundo real para que sea procesada por el resto de los componentes de la arquitectura para proveer un valor añadido.

En la siguiente figura se resalta su posición dentro de la arquitectura propuesta:



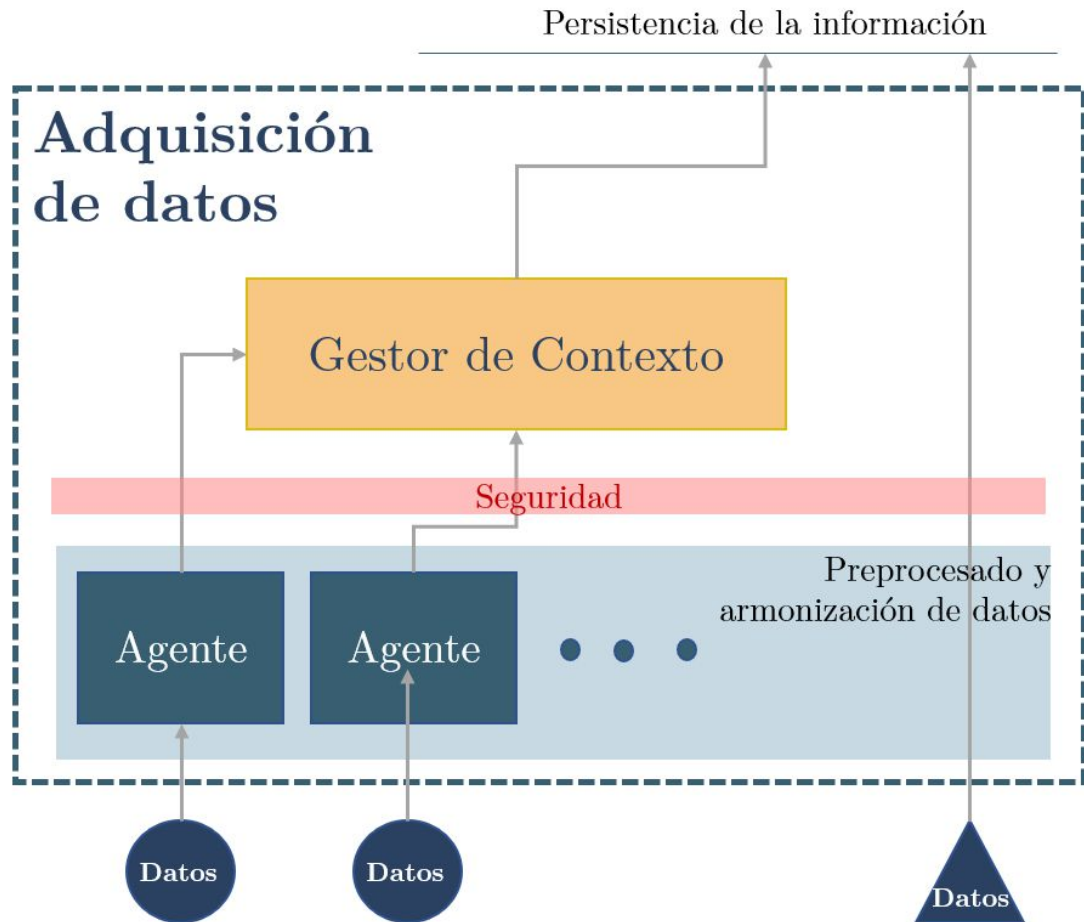
**Figura 3.10:** Módulo de adquisición de datos dentro de la arquitectura propuesta

Este módulo tiene como objetivo cubrir los requisitos funcionales de capacidad de importación de datos de información histórica, incorporación de datos en tiempo real e introducción manual de datos. Además, es el encargado de garantizar la interoperabilidad sintáctica de los datos.

Analizando la estructura de este módulo, decidió separarlo en dos grandes bloques:

- 1) **Preprocesamiento de la información:** Captura de los datos desde su fuente de origen, siendo capaz de recibir información o de pedirla síncrona o asíncronamente. Este subcomponente también debería incluir la armonización de los datos a un formato común.
- 2) **Gestor de contexto:** Concentración de la información contextual de los sensores (o elementos IoT – que actualizan información dinámicamente en el tiempo). Este elemento debe ser considerado de forma individual ya que ha de poseer una serie de características específicas.

La siguiente figura muestra la estructura general de este módulo, emplazando al lector a las siguientes subsecciones para descubrir el diseño de cada uno de ellos.



**Figura 3.11:** Subcomponentes en el módulo de adquisición de datos

Se observa que en el esquema aparece un componente adicional: la entrada al sistema a través de la seguridad. Este módulo no se ha detallado todavía pero resultaría incompleto un diseño sin tener en cuenta este elemento.

### 3.3.1. Preprocesado y armonización de datos

El propósito del preprocesado de datos es proporcionar una forma estándar de importar los datos de las fuentes de datos existentes. Debido a la gran variedad de fuentes y formatos de los datos IoT que pueden utilizarse para el cálculo de índices compuestos, no todos ellos pueden integrarse en la plataforma de la misma manera. Además, tal y como se especifica en los requisitos, el sistema debe permitir la carga de información histórica para el entrenamiento de los algoritmos predictivos.

Este elemento de la arquitectura no se trata de una novedad de diseño. Al contrario, en la gran mayoría de arquitecturas IoT se incorpora la necesidad de un elemento intermedio de captura de datos y conversión a un formato común. En

ocasiones este elemento ha sido denominado *bridge*, conector de datos o agente, que es la denominación que se ha preferido en esta tesis.

Existen algunos para aplicaciones muy concretas, sobre fuentes de datos preestablecidas (estandarizadas) como es el caso de AZURE FHIR [217] (en el ámbito médico) o GeoIoT [218], entre otras. Por otro lado, también existen conectores genéricos (como el ofrecido por IBM [219], Apache Edgent [220] o Azure IoT edge [221]), que buscan un objetivo similar al planteado por este componente pero son específicos para plataformas comerciales. De hecho, debido a la gran cantidad de aplicaciones y posibles usos de este tipo de conectores, la mayoría de soluciones/lenguajes/técnicas/protocolos utilizados son propietarios, especializándose en un campo particular.

Por otro lado, existen conectores específicos preparados para la recogida de información en un protocolo específico. Por ejemplo, el protocolo MQTT es ampliamente utilizado en entornos IoT, y existen soluciones preparadas para recibir información publicada usando dicha especificación, actuando de elemento intermedio que podría incluir la armonización esperada en la solución de este trabajo [222]. FIWARE; con sus librerías genéricas en Node.js propone una plantilla para la implementación de dichos agentes (*IoT Agent*) en los casos de disponer de conectores LoraWAN, LwM2M u OPC-UA. Esta serie de herramientas resultaron tremendamente interesantes y fueron seriamente consideradas para recomendarse dentro de la arquitectura de la tesis. No obstante, otra opción se antojó mejor (más adelante).

Muchas veces, este tipo de elementos software se encuentran también integrados dentro de las pasarelas IoT, actuando directamente sobre el *hardware* que embebe los sensores [194] [223] [224]. Aunque este tipo de aplicaciones son interesantes, no puede entenderse como una respuesta para la solución propuesta, ya que la construcción de índices compuestos tendrá que valerse (en el caso general) de información proveniente de diferentes elementos, sensores, orígenes hardware, etc. Por tanto, ha de plantearse una alternativa más extensible. No obstante, si estas pasarelas ofrecen la posibilidad de customización directa de la información desde el punto de recogida [165], esto podría aprovecharse y sería un elemento muy positivo en el aspecto *edge computing* de dicho despliegue.

Cabe destacar que se han detectado tecnologías que pretenden abstraer el desarrollo de estos agentes, como UIC (del inglés *Universal IoT Connector*) [225] o el SDK del proyecto DataPorts [226], pero ninguna ofrece una aplicación fiable real del software a fecha de finalización de este trabajo.

Por tanto, para la arquitectura diseñada en esta tesis, decidió abogarse por una aproximación uno a uno, aprovechando los conocimientos adquiridos y una herramienta comentada más adelante, evitando así caer en problemas de integración y silos que alejarían la orientación al servicio de índices compuestos.

Por otro lado, con respecto al formato estandarizado que deben proporcionar como “salida” estos agentes es otro campo de creciente interés en la comunidad de

investigación y aplicación de IoT. Son muchas las entidades relevantes que desde hace unos años están proponiendo la creación de ontologías para definir los modelos de datos que podrían usarse en arquitecturas IoT para manejar una gran cantidad de datos y fuentes diferentes.

En 2015, W3C propuso una ontología ligera basada en SSN (*Semantic Sensor Network*) para la representación de recursos, entidades y servicios [227]. Más tarde, en 2018, el consorcio OGC (*Open Geospatial Consortium*) se unió con W3C para proponer una evolución de la misma (SOSA) [228], incluyendo muestreo y actuación en los modelos contemplados.

Continuando con las propuestas oficiales por parte de estándares, uno de los modelos que cuenta con mayor adopción es SAREF (*Smart Appliances REFerence ontology*) [229], propuesto por la ETSI, y que ha sido utilizado en campos como las *Smart Cities* o *Smart Agri-Food*.

Desde ese momento diversas propuestas de ontologías y modelos de datos se han propuesto para todo tipo de aplicaciones IoT, normalmente especializadas por sectores (por ejemplo, eFTI en transporte [230], SAREF4Health en salud [231] o SAREF4agri [232] entre otras).

Desde otro punto de vista, existen también iniciativas separadas de los canales de estandarización oficial, pero que han dispuesto de cierta fama dentro de la comunidad de desarrollo IoT. Una de las más importantes, y una de las que se ha analizado para la tesis fue LOV4IoT (*Linked Open Vocabularies for Internet of Things*) [233], que no se trata de una ontología en sí, si no de un catálogo de las mismas organizado por sector, permitiendo a los diseñadores de sistemas escoger unas u otras en función de su campo de aplicación. El catálogo de ontologías LOV4IoT hace referencia a 499 proyectos de investigación basados en ontologías para IoT que cumplen con los principios de vocabularios abiertos enlazados (LOV del inglés *Linked Open Vocabularies*) y sus dominios de aplicación.

Tras un análisis de las opciones disponibles, se decidió por usar las especificaciones del API RESTful NGSI v2 [234]. Esta especificación proporciona, la posibilidad de establecer modelos de datos particulares basados en un estándar, así como una serie de características a cumplir por los sistemas que deseen comunicarse entre ellos usando dicha interfaz. Esta referencia de comunicación de datos está inspirada en la especificación OMA (*Open Mobile Alliance*) NGSI.

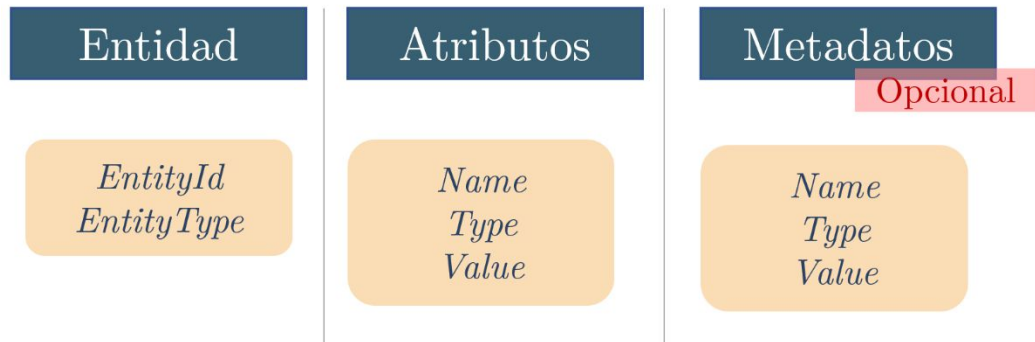
Según esta especificación (a seguir para la comunicación **en toda la solución planteada**), cualquier dato se representa a través de unas estructuras de datos genéricos referidos como elementos de contexto. Un elemento de contexto trata de recopilar toda aquella información útil asociada a una fuente de datos del sistema IoT (relevante para su procesamiento, análisis o actualización). Cada entidad posee una tripla de valores informacionales que la caracterizan:

- Toda entidad NGSI posee un identificador único que sirve para su referenciación a diferentes niveles. Además, al contrario que otros

sistemas como SSN o SAREF, este modelo especifica en este punto el tipo de entidad del que se trata (*EntityId* y *EntityType*).

- Atributos de la entidad (elemento de contexto), que representan características de la entidad (por ejemplo, “velocidad”, “carga”, “número de vehículos” o “temperatura”). Dentro del modelo, se expresan como una tripla de valores: *name*, *type* y *value*.
- Además, pueden existir metadatos, vinculados a los atributos a un elemento de contexto. Esta información es opcional y adquiere especial relevancia en aquellos casos donde varias ontologías iniciales son utilizadas y los requisitos de alineamiento semántico se evidencian notablemente. No es el caso de la presente tesis doctoral.

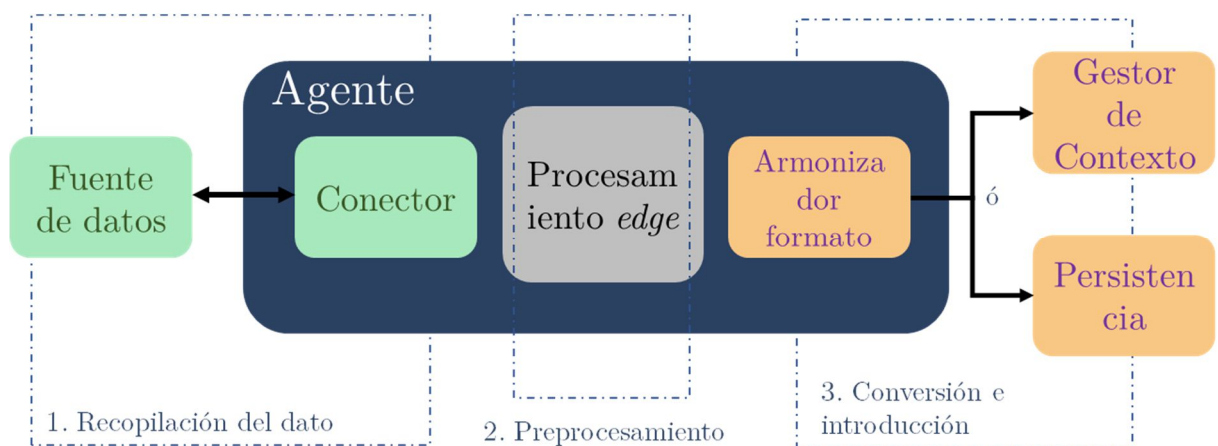
Usando estas indicaciones, cualquier entidad (física, de aplicación, simple, compuesta) existente en un entorno IoT y manejable por una arquitectura, es representada. Una ilustración gráfica de lo anterior se encuentra en la siguiente figura:



**Figura 3.12:** Perspectiva de elementos en NGSI

Analizado lo anterior, en las siguientes páginas se muestran las decisiones de diseño e implementación de agentes que se tomaron para la arquitectura de la tesis doctoral.

En primer lugar, se describe a continuación la estructura de los mismos:



**Figura 3.13:** Estructura de los agentes en la arquitectura IoT

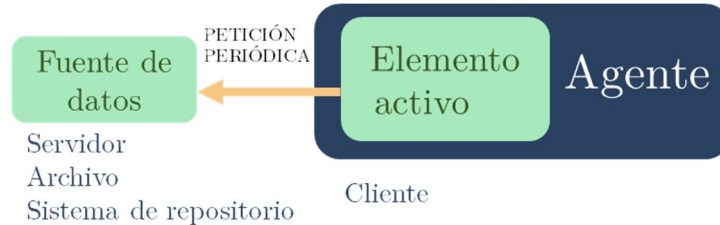
Los agentes (conectores de datos IoT) en la arquitectura están compuestos por tres partes principales:

- El conector directo con la fuente de datos.
- El procesador de datos, donde ocurre el preprocesamiento y la ejercitación de funciones de *edge computing*.
- El conector NGSI, que traslada la información final procesada al formato (modelo de datos compatible con NGSI escogido) y organiza el envío de información (publicación) al siguiente nivel.

A continuación se exponen las conclusiones obtenidas analizando cada uno de ellos y su aplicación en la solución propuesta.

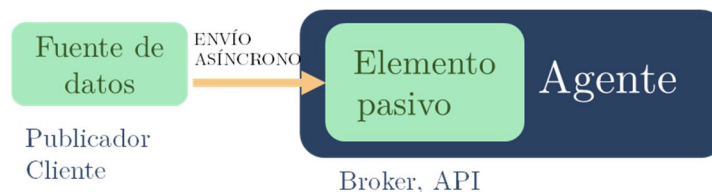
1. **En primer lugar, la conexión con la fuente de datos.** La implementación de este componente depende completamente de dicha fuente. Los diferentes tipos de agentes posibles según diseño son:

a) *Extracción directa desde la fuente (agente activo):* agente que se conecta a la fuente periódicamente para recuperar información. Si la fuente es un archivo estático, se trata de actualizaciones en un repositorio (por ejemplo, FTP) o proporciona una API para recuperar los datos, el agente para esta fuente de datos implementa un cliente y se conecta periódicamente (en función de su naturaleza) a la fuente para adquirir nuevos datos.



**Figura 3.14:** Tipo de agente activo

b) Agente que escucha los datos enviados por la fuente de datos (agente pasivo). Para algunas fuentes de datos, es más fácil enviar los datos que implementar una API (suele ser habitual en casos de sensores que vienen acompañados de microcontroladores, por ejemplo Arduino). Para esos casos, el Agente implementará algún sistema (API, *broker*...) para permitir a la fuente de datos enviar sus datos cuando quiera.



**Figura 3.15:** Tipo de agente pasivo



**2. En segundo lugar, el preprocesamiento.** La implementación de este componente depende mayoritariamente de los servicios esperados de ejecución sobre dichos datos. Se trata del elemento más abierto ya que su materialización es completamente personalizada, pudiendo incluir otros componentes dentro del mismo (bases de datos, conexión con otros sistemas, añadido de datos, corrección, etc.). Para el caso de estudio de esta tesis doctoral, se ha realizado un listado con las principales acciones que pueden tener lugar en este elemento de preprocesamiento en el *edge*.

- Conversión de diferencias entre sistemas de tiempo (zona horaria, expresiones UTC, no UTC...).
- Diferente granularidad de la información. La fuente de datos puede proveer datos con una frecuencia de muestro (por ejemplo, 50 ms) diferente a la requerida por el servicio o por la arquitectura general (por ejemplo, 1 s). En este caso, este preprocesamiento debe realizar un trabajo de consolidación de los datos.
- Correcta interpretación y transformación de los datos de origen en cuanto a aspectos semánticos, contemplando casos como: (i) diferenciación mayúsculas y minúsculas, (ii) desambiguación de sinónimos, (iii) generalización versus especialización (es decir, reduciendo o ampliando una información según conveniencia para el servicio), (iv) agregación de información, (v) datos en diferente codificación (ANSI, UTF-8...), (vi) unidades de medida o (vii) diferencias de precisión esperada (decimales, nivel de detalle...). Para una implementación parcialmente automatizada de los aspectos anteriores existen trabajos en el estado del arte [235] [236], no obstante en esta tesis se ha optado por una materialización caso por caso a nivel código de cada agente, ya que se trata de un campo de estudio muy profundo, más allá del alcance de esta tesis.
- Implementación de las operaciones locales según los esquemas de aprendizaje federado. Como se expone en la sección 2.4.2, las propuestas actuales de aprendizaje federado se basan en la ejecución de una serie de procesos de *machine learning* (inferencia, aprendizaje, filtrado, análisis, anonimización, privacidad, transformación de datos) que deben ocurrir en los elementos más cercanos a las fuentes de datos. Dentro de la arquitectura propuesta, éstos serían los agentes. Por tanto, en aquellos escenarios donde se quisiera reforzar el componente de computación de borde mediante la introducción de aprendizaje federado, este tipo de operaciones tendrían lugar en el ámbito de preprocesamiento. Otro preprocesado según conveniencia.

- 3. En tercer lugar**, la conversión de datos a un modelo interoperable y la introducción dentro del resto del sistema IoT. Al existir datos altamente variables en tipo de fuentes y formato, los agentes deben proporcionar una vista estándar con el fin de insertarlos siguiendo un mismo formato que permita la ejecución de los servicios. Por tanto, esta fase dentro de los agentes está altamente supeditada al uso posterior de los mismos (servicio de cálculo y predicción de índices compuestos).

En la práctica, la implementación de este componente en los agentes para esta arquitectura, ha dependido de la elección del modelo de datos (*Data Model*) a utilizar para el servicio de indicadores compuestos perseguido en la tesis. Según la Universidad de Princeton [237], un modelo de datos organiza cómo elementos reales se representan en un entorno digital. Atendiendo a la definición del servicio objetivo de esta tesis, **se trata de representar la realidad a través de “indicadores raíz” provenientes de datos -IoT- que representan valores del mundo real**. Por tanto, se debe seleccionar un modelo de datos para expresar estos indicadores.

La idea es conseguir una representación común de todo el conocimiento que requiere el servicio (que debe tener el sistema) considerando igualmente la información que puede provenir de las fuentes de datos.

Considerando las elecciones tecnológicas que se han realizado en otros módulos (véase Gestor de Contexto -3.3.2- o Seguridad -3.6), y siendo conscientes de los beneficios de flexibilidad de la especificación NGSI (modelo de representación de entidades escogido), pareció oportuno explorar los modelos de datos propuestos por FIWARE. A parte de los componentes de software abierto proporcionados (a utilizar), la comunidad de FIWARE trabaja en la definición de modelos de datos que puedan utilizarse para representar datos IoT y construir servicios sobre ellos. Estos modelos de datos se han armonizado para permitir la portabilidad de los datos (requisito de la arquitectura) para diferentes aplicaciones, incluidas, entre otras, las ciudades inteligentes. Además, la propia comunidad FIWARE fomenta la utilización de estos modelos en las arquitecturas basadas en sus *enablers*, permitiendo la extensión libre de dichos *data models*.

Finalmente, dentro de este grupo de trabajo se ofrece la posibilidad de solicitar la inclusión de nuevos (creados) modelos de datos dentro de su iniciativa *Smart Data Models* [238], mediante la cual puede convertirse dicho modelo en oficial de la comunidad, siendo un reclamo y un gran punto a favor de cara a una explotación de la solución (véase sección 6).

En este punto, por tanto, la pregunta formulada fue: *¿existe algún modelo de datos que puede representar estos indicadores raíz para la arquitectura orientada al servicio de CI?* En caso de no existir, *¿puede encontrarse alguna propuesta que pudiera ser extendida para alcanzar un esquema que satisficiera las necesidades de dicho servicio?*

A la hora de explorar los modelos disponibles, se tuvo en cuenta que el objetivo del modelo de datos era representar dichos indicadores cumpliendo con las siguientes condiciones.

- Debe ser significativo, capaz de indicar los atributos oportunos esperados por el servicio de cálculo de indicadores compuestos.
- Debe responder a los protocolos de comunicación (NGSI, HTTP REST) y al resto de especificaciones técnicas expuestas en este capítulo.
- Debe ser útil para expresar cada uno de los indicadores a procesar, identificando los tipos de datos, unidades y diferentes casuísticas que puedan aparecer.
- Debe ser flexible, de tal manera que puedan añadirse nuevos campos a medida que el servicio crezca en complejidad.

Atendiendo a todo lo anterior, se eligió extender el modelo de datos propuesto por FIWARE para KPIs, llamado *KeyPerformanceIndicator*. Este *data model* se encuentra obsoleto dentro de la referencia oficial de FIWARE, pero resultó el más oportuno para su utilización en la tesis. Además, al plantearse de todos modos una extensión, se reforzó la idea de seleccionar un modelo de datos que no estuviera siendo continuamente actualizado. En la siguiente figura se muestra un ejemplo de uso de este modelo de datos (extraído directamente de la página de referencia):

```
{
  "id": "kpi-2016-Ciudad-containers-faults",
  "type": "KeyPerformanceIndicator",
  "name": "Incidencias-Contenedores-Mensual",
  "description": "Number of incidences raised on containers per month",
  "category": ["quantitative"],
  "organization": {
    "name": "Ayuntamiento de Ciudad"
  },
  "provider": {
    "name": "Cleaning Service Provider S.A."
  },
  "kpiValue": 20,
  "currentStanding": "good",
  "calculationPeriod": {
    "from": "2016-06-01",
    "to": "2016-06-30"
  },
  "calculationMethod": "automatic",
  "calculationFrequency": "monthly",
  "dateModified": "2016-06-29T15:59:09.224Z",
  "dateNextCalculation": "2016-07-31",
  "address": {
    "addressLocality": "Ciudad",
    "addressCountry": "ESP"
  }
}
```

**Figura 3.16:** Modelo de datos KeyPerformanceIndicator (FIWARE)

A la hora de realizar la extensión, se tuvo en cuenta el avance de la investigación que se reporta en la sección 4. Esta información aún no se ha aportado, pero conviene reflejar en este punto de la memoria cuál fue el resultado (proceso iterativo - 0) del diseño de este modelo de datos, que se denominó *cKPI – composedKeyPerformanceIndicator*:

- Los campos “*id*”, “*type*”, son obligatorios según la especificación NGSI. Los campos “*name*” y “*description*” se mantuvieron para definir la fuente de datos asociada a dicho indicador. Los campos “*process*” y “*calculationMethod*” decidieron mantenerse y resultaron muy útiles para el cálculo del indicador compuesto secundario, orientado a reflejar el grado de fiabilidad (IoT) del indicador mostrado, representando el proceso de adquisición de datos empleado y el método de cálculo. El valor “*kpiValue*” es auto explicativo. Los campos “*calculationPeriod*”, “*dateModified*” y “*dateNextCalculation*” se mantuvieron, teniendo que ser completados por la lógica interna del agente.
- Se añadieron los campos “*ciccategory*”, para representar el índice, “*ciclevel*” para indicar el nivel dentro del árbol de cálculo del CI, “*sourceCIC*”, para indicar la subdivisión de primer nivel y “*unit*”, para expresar en qué unidades se está enviando ese *cKPI*.

Una referencia detallada de dichos valores se encuentra en la sección 4.2.1.

```
{
  "id": "cKpi-CH4-ships",
  "type": "ComposedKeyPerformanceIndicator",
  "name": "CH4",
  "description": "CH4 emissions ships",
  "category": ["quantitative"],
  "organization": {
    "name": "THPA"
  },
  "process": "vessel calls and air and emission factors",
  "calculationFrequency": "weekly",
  "calculationMethod": "automatic",
  "calculationPeriod": {
    "from": "2019-12-29",
    "to": "2020-01-04"
  },
  "kpiValue": 0.1,
  "dateNextCalculation": "2020-01-11",
  "dateModified": "2020-01-04T23:59:59.000Z",
  "ciccategory": "3",
  "ciclevel": "Indicator",
  "unit": "ton"
}
```

**Figura 3.17:** Ejemplo de modelo de datos propuesto - cKPI

Con respecto a la implementación de estos agentes, existieron varias opciones. En el análisis inicial de este componente de adquisición de datos se han comentado varias alternativas, destacando entre ellas los IoT Agents propuestos por FIWARE. Se trata de software predesarrollado (en Node.js) con unas normas muy específicas, orientado a la conexión directa con sensores MQTT, LoraWAN, OPC UA y LwM2M. Esta solución, aunque factible, resultaba demasiado limitante principalmente en la segunda parte de actividad de los potenciales agentes (véase Figura 3.13: Estructura de los agentes en la arquitectura IoT).

Una opción más interesante fue la utilización de la librería *pyngsi* [239]. La elección de esta librería responde a tres razones principales: (i) el candidato colaboró en su diseño, que se llevó a cabo durante el proyecto PIXEL y por parte de uno de los socios del mismo (ORANGE), (ii) ha sido utilizada en varios de los escenarios de validación (sección 5), confirmando su conveniencia para el tipo de aplicaciones como el servicio de CI, (iii) su flexibilidad, incorporando nativamente aspectos como la indicación de la dirección del Gestor de Contexto ORION (elegido en la arquitectura) por defecto, entre otras.

La versión actual de esta librería soporta de forma nativa tres tipos diferentes de fuentes de datos: (1) Exponer una API para recoger datos en formato CSV o JSON, (2) Recoger datos de una fuente que proporcione JSON o CSV desde una API o FTP y (3) Suscribirse a un canal MQTT. La utilización de *pyngsi* para otro tipo de fuentes de datos es posible, pero es necesario desarrollar mayor cantidad de código adicional.

Una vez desarrollados, los agentes deben ser instalados en el sistema donde resida la solución IoT completa. Para ello, siguiendo los principios de diseño de la arquitectura, debe materializarse en imágenes Docker que sean posteriormente ejecutadas.

Para conseguir esta ejecución de los agentes, se encontraron dos opciones. La primera fue la utilización del software *Data Acquisition Layer* del proyecto PIXEL [19]. Esta opción fue considerada perfectamente válida para la arquitectura, y de hecho resulta natural, al provenir del mismo proyecto donde el doctorando realizó validaciones de la arquitectura y siendo absolutamente compatible con *pyngsi*. No obstante, se trata de un software con complejidades innecesarias para la solución de la tesis. Además (este fue un factor diferencial), está preparado para una arquitectura multi-tenancia en lugar de multi instancia, perdiendo las bondades que la caracterizan al optar por la segunda. Por otro lado, el SDK de conectividad semántica de datos del proyecto DataPorts [226] promete ofrecer una funcionalidad similar, permitiendo (mediante un asistente gráfico) obtener unos agentes pre-desarrollados, gestionables después a través de una cómoda interfaz de usuario. No obstante, esta herramienta no se encuentra disponible en abierto en el momento de finalización de esta tesis.

A partir de lo anterior, y observando las directrices generales de la arquitectura, se decidió que cada agente debía ser desarrollado, probado e instalado

de forma manual y ejecutado en el entorno seleccionado vía la utilización de comandos Docker por consola.

En las siguientes imágenes se muestran un ejemplo de utilización de *pyngsi* y un ejemplo de una serie de agentes funcionando utilizando esta metodología.

```
FROM ubuntu
LABEL ngsiagent="cic"
LABEL ngsiagent.type="scheduled"
LABEL ngsiagent.scheduled="* * * * *"
LABEL ngsiagent.datasources=["urn:DataSource:cKPI"]
LABEL ngsiagent.datamodels=["/composedKeyPerformanceIndicator.json"]
ENV PIXEL=test
ENV MYTEST=test
ENV SCHEDULED_DELAY=0
COPY docker_entrypoint.sh /docker_entrypoint.sh
RUN chmod u+rx /docker_entrypoint.sh
ENTRYPOINT ["/docker_entrypoint.sh"]
```

Figura 3.18: Instalación de un agente mediante comandos

```
nacho@cover1:~/image-3/image-1$ sudo docker ps -a
CONTAINER ID   IMAGE                                COMMAND                                CREATED        STATUS
a768ddb93b41   frovlad/alpine-python2             "/bin/sh"                                12 minutes ago   Exited
1569e221a568   node                                "docker-entrypoint.s..."              13 minutes ago   Exited
55fc1753bf66   traffic-thpa-trucks-rfid           "/pyngsi_run"                            13 minutes ago   Created
a60780f7941e   vessels-nachoimo                   "/pyngsi_run"                            15 minutes ago   Created
265990f5444d   traffic-agente                      "/pyngsi_run"                            21 minutes ago   Created
```

Figura 3.19: Agentes instalados funcionando según indicaciones

Como parte de las recomendaciones de la arquitectura, para probar el funcionamiento de un agente lo más rápidamente posible, se propone la utilización de herramientas alternativas a lo definido para su despliegue. Para acelerar el proceso de integración, es esencial que el proceso de importación de datos pueda validarse rápidamente y que los datos estén disponibles para las demás partes del sistema. Se propone la utilización de herramientas de programación basadas en flujo para el prototipado y comprobación rápida de la validez de los agentes: Node RED [240] y Apache NIFI [241]].

En resumen, la implementación propuesta para los agentes NGSI en la arquitectura de la tesis ha sido:

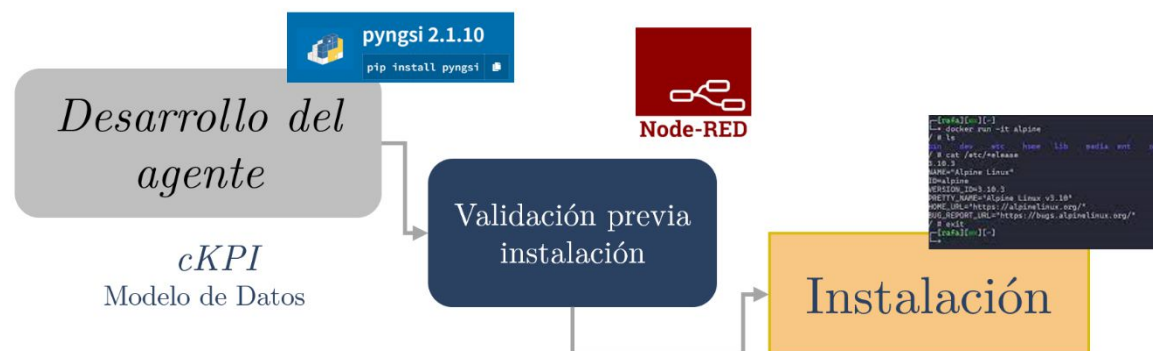


Figura 3.20: Flujo instalación de agentes sobre la arquitectura

### 3.3.2. Concentración de información contextual

En toda arquitectura IoT se debe disponer de un sistema de concentración de la información proveniente de los dispositivos (o agentes conectores de datos). Esta funcionalidad es necesaria debido a que los sistemas IoT, por definición, deben dar acceso a la información emitida por un gran número de fuentes, y realizar esto de una manera individualizada sería un planteamiento inasumible en términos de recursos y de escalabilidad [242]. Existen nuevas alternativas como arquitecturas *brokerless*, sin madurez tecnológica suficiente y fuera del alcance de esta tesis [243].

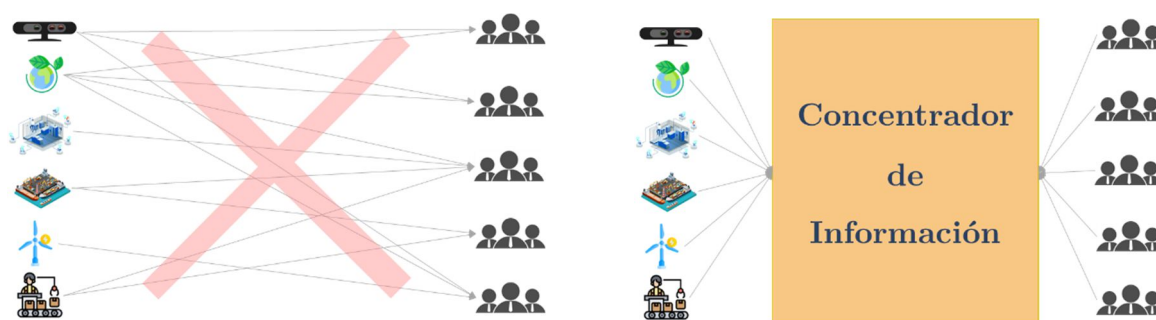


Figura 3.21: Agentes instalados funcionando según indicaciones

Para conseguir esta comunicación a partir de un elemento concentrador, se hace normalmente uso de tecnologías basadas en mecanismos de publicación-suscripción [244]. Este patrón de comunicación consiste en la existencia de emisores (clientes o *publishers*) que, en lugar de enviar mensajes a un receptor de manera directa, los clasifican en temas predefinidos (*topics*) y los envían a un elemento central (gestor de mensajería), sin importarles quién va a recibir esos mensajes. Por otro lado, dichos receptores (suscriptores o *subscribers*) declaran su interés en recibir cualquier nuevo mensaje que se publique en alguna de las clasificaciones definidas. La clave de este sistema se encuentra en el componente “puente” (*broker*), que incorpora la capacidad de registrar dichos “temas”, escuchar los mensajes provenientes de clientes y repartirlos (*dispatch*) convenientemente a los suscriptores correspondientes. La incorporación de este patrón resuelve el problema mencionado, así como relaja las condiciones de desarrollo de los conectores de datos (agentes).

Este mecanismo fue trasladado a un protocolo de comunicación sobre redes TCP/IP creado por IBM en 1999, llamado MQTT (*Message Query Telemetry Transport*) [245]. Se trata de un protocolo ligero que consume muy poco ancho de banda y es capaz de soportar condiciones de conectividad poco fiables [222]. Por todo ello, se trata del más utilizado a la hora de realizar esta “pasarela” de conexión de la información proveniente de fuentes IoT.

Una implementación básica de esta tecnología se encuentra presente en la mayoría de los despliegues IoT, instanciada a través de *brokers* MQTT sencillos.

Estos *brokers* son, principalmente, Mosquitto [246], Aedes [247] y Moquette [248]. Aunque existen algunas soluciones que evolucionan dichos *brokers* sencillos, dotándolos de alguna funcionalidad adicional (por ejemplo, VerneMQ [249]) este tipo de implementaciones actúan simplemente como distribuidores de la información, exponiendo una serie de *topics* y consiguiendo que varios consumidores accedan a información de múltiples dispositivos/fuentes asociadas a los mismos, siendo insuficientes si se desea incorporar funcionalidades más complejas (monitorización de mensajes, control de la comunicación, API de programación y consulta, entre otras).

Como se ha mencionado, la creación del protocolo MQTT data de hace más de 20 años. En una línea similar, el protocolo AMQP (*Advanced Message Queing Protocol*) fue creado en el año 2004 para evolucionar MQTT incluyendo características como transaccionalidad similar a HTTP [222]. Más tarde, en 2005, surgió STOMP (*Simple -or Streaming- Text Orientated Messaging Protocol*) [250], planteado como una vertiente orientada al trabajo con *websockets* y a través de mensajes de texto. Los tres protocolos son anteriores a la aparición del IoT (2010) [251]. En este tiempo, soluciones sobre MQTT han surgido en el estado del arte orientadas a otro tipo de aplicaciones de conexión M2M, yendo más allá de la simple conexión entre publicadores y suscriptores.

Estas soluciones son las conocidas como gestores de mensajería (*message brokers*) y son utilizados en diferentes sectores para gestionar una comunicación asíncrona entre componentes de una arquitectura de software, orientados principalmente a dotar de escalabilidad a los sistemas. Para el estudio de este módulo de la arquitectura de la tesis, se estimó oportuno profundizar en dichas implementaciones y sus capacidades.

Las soluciones de gestión de mensajería más utilizadas son Apache Kafka [252], RabbitMQ [253], ActiveMQ [254], HiveMQ [255] y ZeroMQ [256], cada una de ellas especializadas en alguno de los protocolos mencionados o en alguna característica en particular. No obstante, las dos más ampliamente aceptadas, que abarcan las funcionalidades avanzadas mencionadas son Apache Kafka y RabbitMQ.

**Apache Kafka** es un gestor de mensajería orientado a la distribución con una alta tasa de refresco, alcanzando *throughputs* de 100.000 mensajes por segundo. Como capacidades adicionales a los gestores de mensajes habituales en IoT, permite histórico de *stream* de mensajes hasta una duración temporal fija. No obstante, se trata de un software con más requisitos de implementación, necesitando estar acompañado de un elemento adicional (Apache Zookeeper [257]) para agrupar *topics* y suscripciones, lo que supone mayor requerimiento de hardware. Además, su utilización por parte de clientes no-Java no está en estadios maduros de utilización.



**RabbitMQ**, por su parte, es una solución muy robusta que incorpora AMQP como protocolo principal, pero es compatible con los otros dos a través de *plugins*. Al contrario que implementaciones ligeras sobre MQTT (ideales para entornos con mala conectividad y pocos recursos en el lado del cliente/publicador), RabbitMQ se centra en asegurar el envío consistente de los mensajes, implementando diversos mecanismos con este fin (confirmación de recepción, conservación de histórico hasta que existe garantía de recibo). Como características interesantes que lo convirtieron en un claro candidato fue la existencia de una API para la interacción con esta tecnología y la incorporación de forma nativa de una interfaz web para su utilización. Además, un aspecto relevante es la inclusión de un mecanismo de seguridad desde el cliente y desde el suscriptor a través de certificados X.509 [258]. No obstante, presenta una complejidad excesiva para entornos en que existan condiciones relajadas de recepción de mensajes, alcanzando un *throughput* máximo de únicamente 20.000 mensajes por segundo.

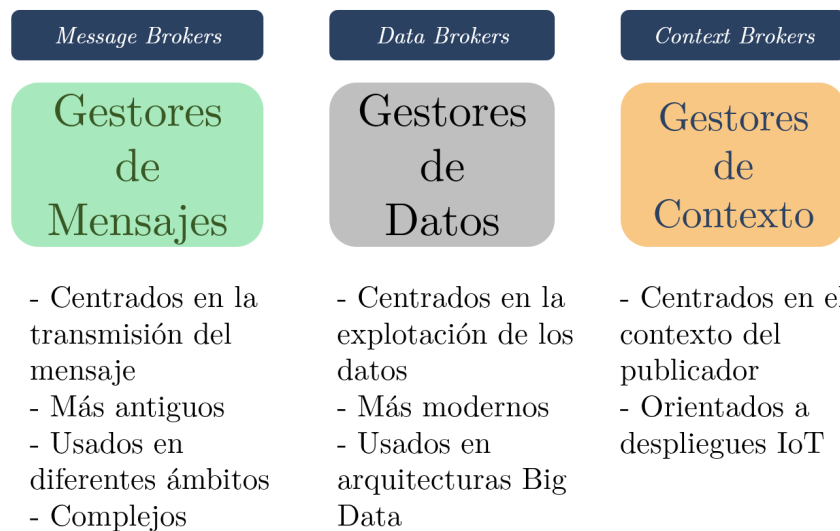
Todos los ejemplos vistos hasta ahora se fijan en la transmisión del mensaje, no en la gestión de los datos que estos pueden estar transportando o en el contexto de los mismos (qué entidades, de qué tipo, cuáles han sido los cambios, etc.).

En este sentido, otras tecnologías que trabajan sobre los principios de mensajería son los *Data Brokers*. Estas soluciones, aun pudiendo ser comprendidas como una evolución de lo anterior, se encuentran más orientadas a la extracción de los datos a distribuir, para realizar un análisis de los mismos [259]. Siendo más comunes en los entornos y arquitecturas de Big Data, su principal objetivo es la agregación de datos previo envío a suscriptores, difiriendo de los gestores de mensajería en la intervención sobre la información originalmente recibida (habitualmente comercialización de la misma). Tecnológicamente, se caracterizan por ser más complejos, sobretodo en sus elementos centrales de análisis de la información. Pueden incluir un amplio abanico de tipos de bases de datos, con requerimientos dispares de hardware. Este tipo de gestores, orientados a la explotación de los datos, fueron descartados de la tesis doctoral.

No obstante, se han encontrado implementaciones de este tipo de gestores en los campos de IoT, como el broker de Cumulocity [260] o, más recientemente, la solución de código abierto Yggio [261]. Esta última fue de especial interés, ya que se centra en el procesamiento de series temporales (relacionado con el servicio objetivo de esta tesis), y permite la creación de funcionalidades de agregación y creación de valor sobre su elemento central (core), añadidas a las nativamente incorporadas de control de acceso e informe de dispositivos. Además, incluye la integración de agentes provenientes de fuentes habituales en IoT tales como LoRa [262], Z-Wave [263] o HTTP. Sin embargo, es cierto que Yggio es un software de muy reciente creación (2021), y la decisión de elección de arquitectura se llevó a cabo mucho antes en el cronograma de la tesis. Además, no incorpora funcionalidades que propone la tecnología finalmente escogida.

Estos últimos se fijan en los datos y en su gestión, pero prestan una menor atención al contexto de los mismos (qué entidades, de qué tipo, los cambios, etc.).

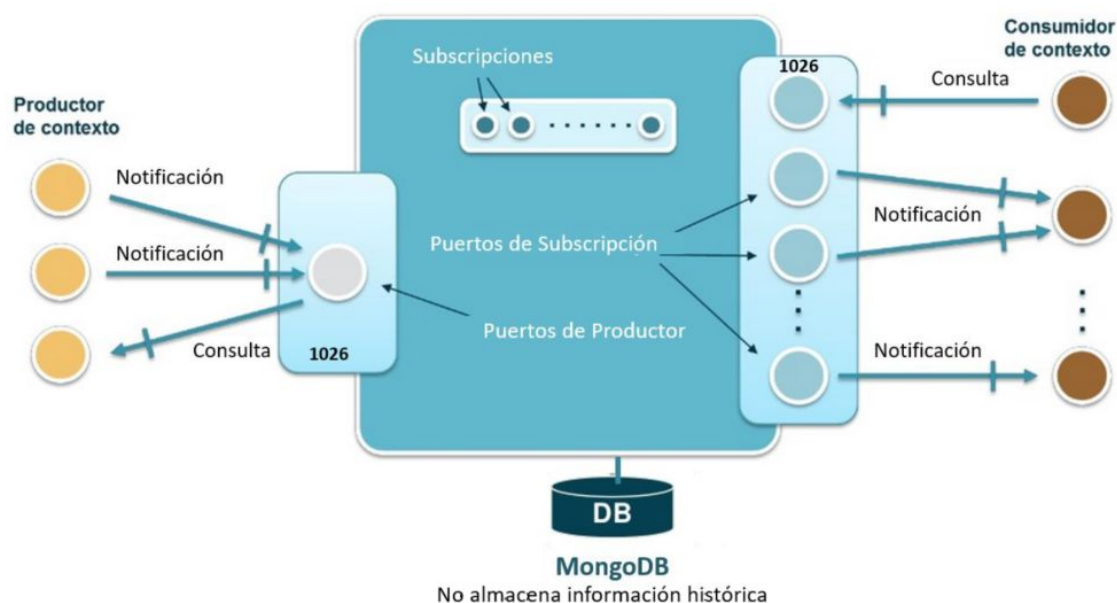
En este sentido, el último tipo de concentradores en la información que se analizaron fueron los **gestores de contexto o de información contextual** (en inglés, *Context Broker*). Este tipo de soluciones se basan igualmente en el patrón de publicación-suscripción, pero se centran en el contexto del envío de dichos mensajes, incorporando funcionalidades como el registro de dicho contexto, descubrimiento de entidades, mediación o notificación. La introducción de “contexto” propone cambiar el paradigma utilizado en los dos tipos anteriores. Hasta ahora, cada publicador era entendido como un elemento generador de información (mensajes) que debían alcanzar un destino (primer caso) o que poseían valiosos datos a explotar (segundo caso). Los gestores de contexto plantean entender estos publicadores como entidades en sí mismas (asimilables a objetos o elementos del mundo real – IoT), que pueden variar en sus condiciones contextuales. Estas condiciones se expresan en forma de atributos que se almacenan en dichos gestores de contexto (en lugar de ser únicamente piezas de información que esperan en una cola para ser atendidas). Se trata de herramientas muy potentes ya que permiten la gestión de mensajes desde diferentes dominios de aplicación y con formatos de datos dispares, cuyo registro y publicación para aquellos potenciales suscriptores es también una ventaja. Estos concentradores ofrecen siempre una API de acceso externo para la modificación, consulta, suscripción y configuración del concentrador. Como puede comprenderse, la utilización de estos gestores de contexto enriquece enormemente un sistema, especialmente en el campo del Internet de las Cosas.



**Figura 3.22:** Comparación entre concentradores de información

Por tanto, para la arquitectura de la tesis se apostó claramente por el uso de un gestor de contexto con la finalidad de cubrir esta necesidad de conexión centralidad de la información. Existen varias tecnologías que responden a este modelo, como Ratatosk de Sensative [264], pero sin lugar a dudas la principal implementación de este concentrador es ORION de FIWARE [265].

Este software implementa una potente API basada en el modelo de información NGSi, la cual permite registrar entidades proveedoras de información de contexto, obtener y actualizar los datos almacenados en la base de datos, notificar a las entidades consumidoras de cambios ocurridos en los datos, etc. Además, este módulo posee una base de datos propia que permite almacenar el último valor recibido de cada una de las entidades registradas para que pueda ser consultada en todo momento. También dentro de la API incluye un *endpoint* de notificación sobre el cual otros componentes (consumidores de contexto) pueden declarar al Gestor de Contexto que desean ser notificados cuando una entidad presente una actualización de cualquier tipo.



**Figura 3.23:** Esquema de gestor de contexto ORION. Fuente: [265]

Se trata del elemento central de la *suite* tecnológica IoT de la iniciativa FIWARE [266], permitiendo considerar cualquier despliegue de IoT que contenga este elemento (*enabler*) como compatible con FIWARE (*-compliant*).

En él, cada productor de contexto es referenciado como una entidad en el sistema, que poseerá valores, información de contexto y compondrá en sí misma un elemento que, más allá de enviar un flujo de información, podrá ser actualizada, consultada y consumidores de contexto podrán suscribirse a sus cambios (individualmente o de forma agrupada por tipo de entidades, etc.).

Tecnológicamente, es un software desarrollado en C++ que se apoya en una base de datos MongoDB para registrar las entidades y dar soporte al registro de los últimos valores recibidos, todo ello a través de exponer un API en el puerto 1026. Se encuentra disponible para su ejecución como contenedor Docker, teniendo varias imágenes públicas preparadas para su despliegue en diferentes entornos.

Además de su principal presencia como gestor de contexto disponible, las siguientes razones llevaron a tomar la decisión de seleccionarlo como la pieza principal del módulo de adquisición de datos de la arquitectura:

- Se trata de un software Europeo y de código libre.
- La iniciativa FIWARE fue iniciada por Telefónica y Orange en 2012 (dos de las principales operadoras de telecomunicaciones europeas [267]) junto a grandes consultoras tecnológicas como Engineering y ATOS. Otras grandes entidades como NEC, NTT Data, Deloitte, Red Hat o AWS se han unido y han apoyado esta iniciativa desde entonces.
- En el año 2018, ORION fue declarado un componente esencial para la gestión de datos, siendo adoptado como elemento oficial del programa CEF (*Connecting Europe Facility*).
- Posee una amplia comunidad, activa en varios foros y que propone constantes nuevas tecnologías (*enablers*) que parten de este Gestor de Contexto [268].
- Se trata de una tecnología ampliamente utilizada en proyectos de I+D. Algunos ejemplos: DEMETER, DataPorts, INTER-IoT, PIXEL, FogProtect, entre otros.
- Varias ciudades han adoptado ORION y otros productos de FIWARE como la base de sus plataformas Smart City (por ejemplo Valencia [269], Santander, Amberes, Bruselas o Berlín, entre muchas otras [270]). Este punto es especialmente relevante de cara a un modelo de negocio (sección 6) para la solución propuesta en esta tesis doctoral
- Al tratarse de un software ligero, sin necesidad de demasiado consumo de recursos, y al haber sido compilado (imagen Docker) con compatibilidad en varias arquitecturas y SOs [271], resulta una elección ideal para enfatizar el componente de *edge computing* de la arquitectura. En función de las necesidades del caso de uso, puede diseñarse su ejecución en un entorno más centralizado (asimilable a *cloud*, o más cercano a la fuente).

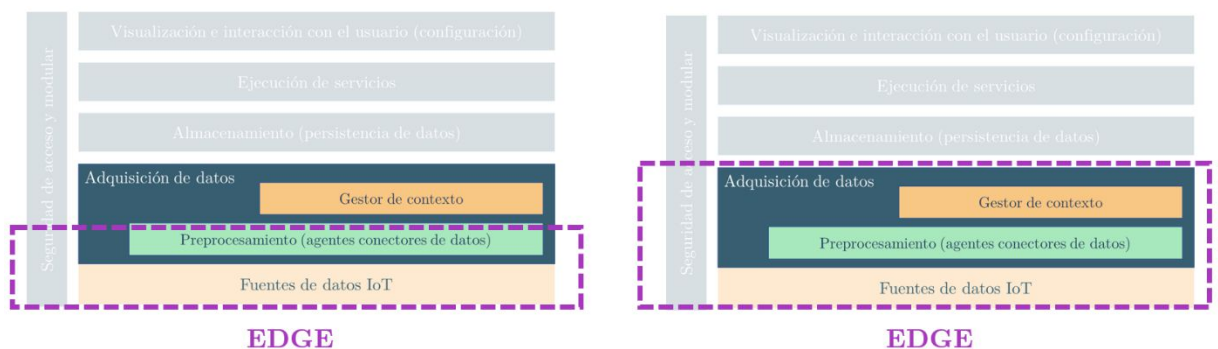


Figura 3.24: Dualidad *edge-cloud* del módulo de adquisición de datos

De hecho, esta fue la primera selección tecnológica seria de la arquitectura (tras los principios de diseño). El hecho de escoger ORION ha marcado fuertemente al resto de módulos y subcomponentes, buscando siempre (en la medida de lo posible) software compatible (o desplegado junto habitualmente con ORION, como los componentes de seguridad - 3.6) y guiando las interfaces (NGSI) y los modelos de datos de todo el sistema. Como notas negativas, ORION no dispone de una interfaz visual para la interacción con sus componentes, consulta de entidades, alta de suscripciones, etc. Estas acciones, junto al resto de opciones de configuración, únicamente pueden realizarse a través de consola o de su API NGSI. No obstante, esto no fue considerado un problema ya que, por requisitos de la arquitectura, la solución solo debe tener una interfaz única de administración, siendo el resto configuraciones por parte de los equipos técnicos de instalación/despliegue.

A la hora de introducir las entidades como elementos publicadores en la arquitectura de la tesis, ya se ha hablado de los agentes y del modelo *cKPI*. En cuanto al resto de consideraciones específicas:

- ORION permite usar una utilidad de separación de entornos. Si un mismo Gestor de Contexto se usara para varias aplicaciones (por ejemplo, tráfico y medioambiente en una ciudad), podría utilizarse el elemento declarativo Fiware-Service. En el caso de esta tesis se ha usado un único: “*CIC*”.
- ORION permite la declaración jerárquica de ámbitos de aplicación, para separar las entidades publicadoras en categorías o agrupaciones sin necesidad de la utilización de atributos de contexto adicionales. Esta utilidad se denomina Fiware-ServicePath [272], pero decidió no usarse en la solución propuesta.

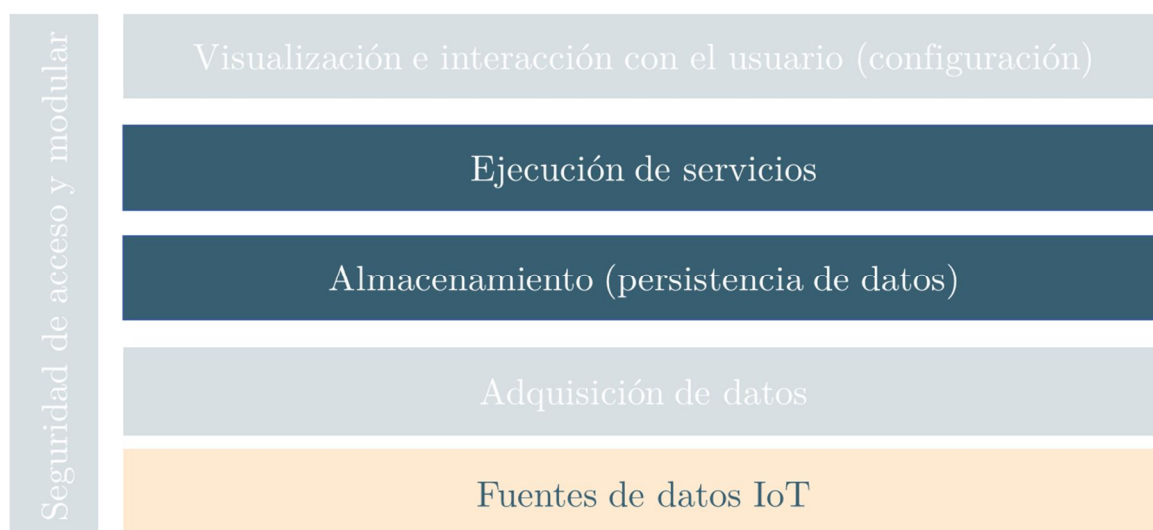
#### **Recapitulación del módulo de adquisición de datos:**

El objetivo principal es trabajar bajo estructuras sintácticas y semánticas uniformes para todas las fuentes de datos entrantes en el sistema IoT. Para ello, se incorpora un elemento de procesamiento inicial (agente de interoperabilidad) para que los datos brutos se ajusten al esquema previsto. Aprovechando el uso de la referencia de código abierto para las soluciones IoT, todos los formatos de datos de este experimento han sido diseñados para seguir los modelos de datos FIWARE NGSI. Además, como corresponde a un sistema IoT en tiempo real, la arquitectura debe poseer un componente de publicación/suscripción que almacene el último valor insertado por cada entidad y sobre el que orbite el resto de la solución. Aunque la tecnología seleccionada para el módulo de Gestor de Contexto es inflexible, indicando FIWARE ORION como la elección escogida, esta arquitectura no obliga a una tecnología en concreto para la implementación de los agentes de interoperabilidad (aunque sí los mecanismos de construcción de datos y su formato final – *cKPI*). La recomendación es utilizar el *framework pyngsi* Python proporcionado por el proyecto PIXEL.

### 3.4. Procesamiento de datos

Una vez se ha conseguido trasladar datos de una fuente IoT a un elemento consumidor de la información, la arquitectura debe abordar el procesamiento de la misma. El objetivo de la tesis es el desarrollo de un servicio de cálculo de indicadores compuestos, por lo que este procesamiento de datos debe estar orientado a conseguir la ejecución de dicho servicio.

En la siguiente figura se resalta su posición dentro de la arquitectura propuesta. Se trata de la combinación de dos de los módulos expuestos inicialmente, como se explica posteriormente:



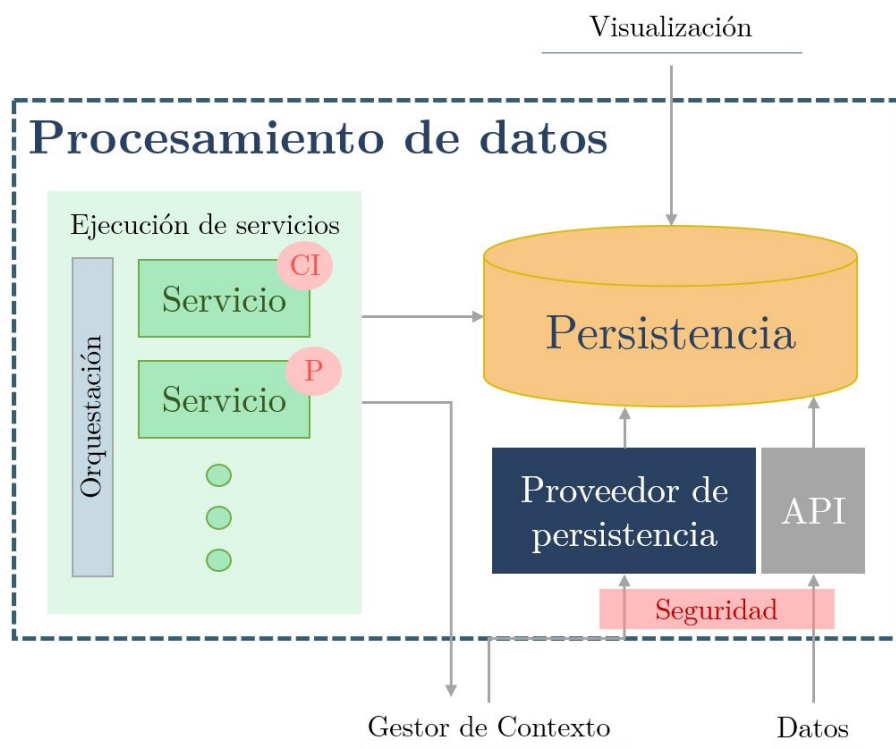
**Figura 3.25:** Módulos de procesamiento de datos en la arquitectura propuesta

Este módulo tiene como objetivo cubrir los requisitos funcionales de almacenamiento a largo y corto plazo de la información, análisis de datos históricos, instanciación de servicios y ejecución automática periódica de los mismos. En definitiva, se trata de los módulos que permitirán al servicio de la tesis doctoral tener lugar y proveer valor añadido a partir de los datos recogidos y ofrecidos por el módulo anterior.

Analizando las acciones necesarias, en esta sección se definen, de hecho, dos módulos de la arquitectura que responden a dichos requerimientos:

- 3) **Módulo de almacenamiento:** Persiste los datos a largo plazo. El gestor de contexto únicamente conserva el último dato disponible de cada entidad. Por tanto, para poder ejecutar servicios sobre datos históricos es necesario introducir este componente.
- 4) **Ejecución de servicios:** Orquestación y despliegue efectivo de los servicios (cálculo y predicción de CIs, y en el futuro otro tipo de aplicaciones) sobre los datos almacenados.

La siguiente figura muestra la estructura general de este módulo, emplazando al lector a las siguientes subsecciones para descubrir el diseño de cada uno de ellos.



**Figura 3.26:** Módulos de procesamiento de datos: división

En las siguientes subsecciones se analiza cada uno de ellos, razonando sobre su necesidad, las tecnologías disponibles, las elecciones justificadas y cómo encajan en el global de la arquitectura. Del mismo modo, se aportan detalles sobre la implementación de estos módulos.

### 3.4.1. Persistencia de la información

La persistencia de datos es necesaria en la arquitectura. El servicio de cálculo de CIs es una herramienta retrospectiva que puede tener una determinada periodicidad (ver sección 4). Por otro lado, el servicio de predicción requerirá la alimentación de datos históricos para su entrenamiento y (una menor cantidad de) datos más recientes para la inferencia. Además, la interfaz de visualización deberá permitir la consulta de información previa al momento actual para aplicar sus reglas de negocio. Es por ello que una base de datos central de persistencia se antoja necesaria más allá del gestor de contexto.

Similarmente a la acción realizada en la captura de requisitos, se antojó necesario un análisis de los datos que van a deber guardarse en dicha base de datos, de tal modo que la selección tecnológica y las decisiones de arquitectura sean las correctas. Para ello se tuvieron en cuenta, principalmente, el tipo de datos que van



a recibirse (y la frecuencia esperada), el objetivo del servicio (cálculo y predicción de CIs) y los requisitos de dicho servicio (véase sección 3.1.1.3).

- La solución debe ser capaz de persistir información proveniente de sensores IoT. Pese a que los servicios requieren un modelo de datos determinado, se establece la posibilidad de insertar de forma directa (a través del gestor de contexto) datos desde los sensores. Estos pueden ser variopintos, pero normalmente se tratará de series numéricas.
- La principal información que deberá almacenarse serán series temporales (introducidas por los agentes) de información en formato JSON (acorde a la especificación NGSI).
- Puede existir una gran cantidad de fuentes alimentando los indicadores raíz, por tanto, la base de datos debe ser escalable horizontalmente (nuevas entidades) y verticalmente (en volumen).
- Carga de documentos binarios, ya que deberá almacenar información como reglas, recomendaciones e informes de resultados de ejecuciones del servicio.
- No se establece la necesidad de almacenar información de imágenes o *streams* de vídeo ni documentos textuales.

Con estas premisas básicas en mente, se procedió a inspeccionar el estado del arte para comprender las posibles opciones disponibles.

En la actualidad existen más de 300 tecnologías de bases de datos disponibles, y creciendo cada día [273]. Se clasifican en muy diversos tipos, y dependiendo de cada aplicación y de los requerimientos particulares, una arquitectura debe decantarse por una o más. En muchas ocasiones, las arquitecturas IoT cuentan con más de un sistema de gestión de bases de datos para cumplir diversos fines. En este caso, se trata de analizar la mejor opción para la persistencia a largo plazo de la información de indicadores. Es por ello que el objetivo es encontrar una base de datos que contenga la información necesaria, que permita una gran cantidad de consultas y sea capaz de escalar horizontalmente (nuevas entidades a registrar). Debe ofrecer un alto rendimiento y ser escalable en el sentido de permitir búsquedas de larga duración.

A la hora de realizar una primera clasificación de las mismas, debe comprenderse que existen dos grandes grupos: (a) bases de datos relacionales (conocidas como SQL, del inglés *Simple Query Language*) y (b) bases de datos no relacionales (o NoSQL -*Not only SQL*) [274]. El primer tipo es el más antiguo, proponiendo una forma de almacenamiento de propósito general basado en tablas que se relacionan entre sí a través de sus identificadores únicos (primera columna por tabla). Es ampliamente usada en muchos tipos de aplicaciones, gozando de un amplio soporte por la comunidad, documentación y grandes compañías tecnológicas que ofrecen sus servicios haciendo uso de ellas. Además, poseen cualidades denominadas ACID [275], asegurando en todo momento consistencia y estabilidad en los datos. No obstante, este tipo de bases de datos tienen varios inconvenientes



como (i) la necesidad de establecer una estructura fija de tablas y relaciones desde el principio (diseño de la solución) que dificultan su escalabilidad o introducción de nuevas entidades, (ii) requieren que las nuevas entradas se adapten siempre a estas estructuras fijas, (iii) se basan en un sistema de consultas atómicas lo que reduce su eficiencia [276]. Algunas de las tecnologías más utilizadas que implementan este mecanismo son MySQL [277], MariaDB [278] Oracle [279], MicrosoftSQL [280] o PostgreSQL [281]. Por otro lado, las bases de datos NoSQL se caracterizan por su flexibilidad y escalabilidad. A pesar de ser relativamente recientes, su acogida es destacable ya que supera alguno de los inconvenientes mencionados de las bases de datos relaciones. Soportan estructuras distribuidas y con diferentes formatos, permitiendo relaciones más flexibles, una mayor tolerancia a fallos y el cambio dinámico en las estructuras sin necesidad de revisar todo el sistema. Por otro lado, permiten crecimiento horizontal en términos de máquinas distribuidas, sin necesidad de centralizar todo el almacenamiento en un mismo entorno. Son, además, ideales para aquellas aplicaciones en las que se precise una estructuración tipo objeto, clases y atributos, con anidaciones y jerarquías imposibles de conseguir en una base de datos SQL. No obstante, no ofrecen la consistencia transaccional ACID (SQL) que proveen las relacionales y al ser más recientes no han sido tan ampliamente utilizadas.

Analizando lo anterior, resulta claro que la mejor opción para despliegues IoT son las bases de datos no relacionales (o NoSQL). Dentro de esta categoría, se hallaron varias subclasificaciones, orientadas a algún tipo de funcionalidad o característica concreta. No todas ellas tienen el mismo grado de madurez o adopción, algo igualmente a considerar. En particular, resulta interesante hacer un repaso de bases de datos tipo clave-valor, bases de datos de familia de columnas, de documentos, de grafos y especializadas en motores de búsqueda [274].

Las bases de datos tipo clave valor almacenan un objeto (o diccionario) otorgándole una clave única identificativa. Aceptan diferentes campos en cada registro, siendo una buena opción para despliegues rápidos que requieran mucha velocidad de almacenamiento y respuesta a consultas [273]. Las tecnologías más populares en esta subclasificación son Redis [282] y DynamoDB [283]. No obstante, estas bases de datos se descartaron ya que no se permite al sistema de gestión consultar el contenido de los registros (solo obtener el objeto completo a partir de su clave).

Las bases de datos de familias de columnas son, en esencia, clave-valor en dos dimensiones. Se trata de tablas (similitud con SQL) donde cada fila puede ser un registro individual tipo-clave valor diferente al resto de filas. Son principalmente usadas en entornos de escalabilidad masiva, añadiendo recursos a clusters en función de la necesidad de llegada de nuevos registros. Además, definen un tipo especial de lenguaje de consulta de registros, CQL (*Cassandra Query Language*), una variante de SQL para este tipo de bases de datos. Las tecnologías más representativas son Cassandra [284] y Hbase [285] (ambas de Apache Foundation) o DataStax [286].

Las bases de datos de documentos almacenan objetos JSON sin un formato estructurado. Cada registro puede provenir con un formato diferentes siempre que responda a la sintaxis JSON, identificando cada uno de ellos con un identificador de documento. Son relativamente similares a las bases de datos de clave-valor con la diferencia que: (i) en este segundo caso esta identificación forma parte del objeto registrado y (ii) los diferentes campos de los JSON pueden ser consultados particularmente y navegados por el sistema de gestión, mientras que en las bases de datos de clave-valor el contenido permanece opaco (tipo caja negra). Las tecnologías más utilizadas son MongoDB [287], Couchbase [288] y CouchDB [289]. Una evolución de estas bases de datos de documentos son las especializadas en motores de búsqueda, con la principal diferencia que enfatizan la capacidad de navegar por el contenido de los documentos a partir de consultas (*queries*) basadas en texto. Este tipo de bases de datos ponen el foco en la accesibilidad a los registros de una forma cómoda y muy estructurada (aunque dichas entradas no respondan a una estructura fija), por ejemplo, permitiendo expresiones que busquen cadenas de texto parciales, agregaciones por contenido, rankings... [290]–[292]. Algunas de las tecnologías más utilizadas son Elasticsearch [293], Splunk [294] o Sphinx [295]. Tanto las bases de datos documentales como de motores de búsqueda resultan altamente interesantes para la solución propuesta en la tesis. Teniendo en cuenta que los registros a almacenar tendrán un formato JSON (especificación NGSI), se antojan claras candidatas para la selección.

Las bases de datos orientadas a grafos permiten la representación de las relaciones entre elementos almacenados como una red de nodos. Se utilizan principalmente en aplicaciones de clasificación (por ejemplo, catálogos) donde la relación entre los diferentes registros es más importante que el contenido de los mismos. Este tipo de base de datos responde más bien a un paradigma y no a unas obligaciones de implementación, ya que por debajo pueden utilizar archivos tipo clave-valor incluso tablas relacionales. Además, dentro de esta subclasificación han surgido variantes de lenguajes de consulta como AQL (*Arango Query Language*) o GraphQL [296]. Las tecnologías más utilizadas son ArangoDB [297], Neo4J [298] u OrientDB [299]. Este tipo de bases de datos se descartaron directamente para la utilización en esta tesis ya que debía primarse más el contenido de los registros que la relación categórica entre los mismos.

Por último, las bases de datos orientadas a series temporales prometieron ser uno de los principales candidatos para la arquitectura IoT. Se trata de sistemas optimizados para gestionar datos indexados por una marca de tiempo (*timestamp*). Permiten agregación temporal, filtrado, operaciones sobre los registros basados en el momento en que fueron registrados. Tradicionalmente, se han asociado a aplicaciones industriales para la captura de datos de sensores, por tanto, su uso en una arquitectura IoT estaría más que justificado. Las tecnologías más utilizadas son Amazon Timestream [300], Prometheus [301] e InfluxDB [302].

A partir de los análisis anteriores, comprendiendo los requerimientos de la solución propuesta, se redujo la búsqueda a bases de datos de clave-valor,

orientadas a documentos y centradas en motores de búsqueda. De entre las mismas, coincidiendo en gran parte con las bases de datos más utilizadas en arquitecturas IoT [292], se realizó una preselección donde se incluyó Cassandra, InfluxDB, Elasticsearch, Prometheus y MongoDB.

En la siguiente tabla se muestra un resumen de su estudio particularizado y de aquellos aspectos más destacados considerando el objetivo de la tesis doctoral:

**Tabla 3.5:** Comparación entre bases de datos candidatas

| BBDD          | Motivos para ser candidata   | Desventajas   |
|---------------|--|---|
| Cassandra     | <ul style="list-style-type: none"> <li>- Disponibilidad 100%</li> <li>- Escalabilidad distribuida</li> </ul>   | <ul style="list-style-type: none"> <li>- Los registros son tipo clave-valor, y los datos a manejar en la solución tienen una estructura más compleja.</li> </ul>  |
| Elasticsearch | <ul style="list-style-type: none"> <li>- Registros por documento</li> <li>- Agregación y navegación por datos internos de registros</li> <li>- Facilidad en la búsqueda</li> <li>- Buena API</li> <li>- Incluida en pila ELK [303]</li> <li>- Multi-idioma</li> <li>- Herramienta de visualización con Kibana</li> </ul> | <ul style="list-style-type: none"> <li>- Requiere muchos recursos.</li> <li>- En caso de volumen de peticiones inmenso (del orden TB por día) su rendimiento y eficiencia son cuestionables.</li> <li>- Curva de aprendizaje.</li> </ul>          |
| MongoDB       | <ul style="list-style-type: none"> <li>- Ligero.</li> <li>- Muy utilizado en despliegues de IoT.</li> <li>- FIWARE ORION ya equipa una base de datos MongoDB.</li> </ul>   | <ul style="list-style-type: none"> <li>- Las consultas se realizan utilizando estructura JSON</li> <li>- Límite de tamaño de documentos.</li> <li>- Necesidad de crear <code>_id</code> inicial para cada documento.</li> </ul>                   |
| InfluxDB      | <ul style="list-style-type: none"> <li>- Orientada a series temporales.</li> <li>- Agregación y filtrado temporal.</li> <li>- Rendimiento y rapidez.</li> </ul>  | <ul style="list-style-type: none"> <li>- La marca temporal se inserta como identificador, no como parte del mensaje.</li> <li>- Poco soporte documental.</li> <li>- Tecnología muy reciente.</li> <li>- Complejo lenguaje de consulta.</li> </ul> |
| Prometheus    | <ul style="list-style-type: none"> <li>- Utilizado en entornos de edge computing</li> </ul>  | <ul style="list-style-type: none"> <li>- Enfocado principalmente a la obtención de métricas</li> <li>- Escasa documentación</li> <li>- No permite estructuras complejas ni navegar indexadamente por documentos</li> </ul>                        |

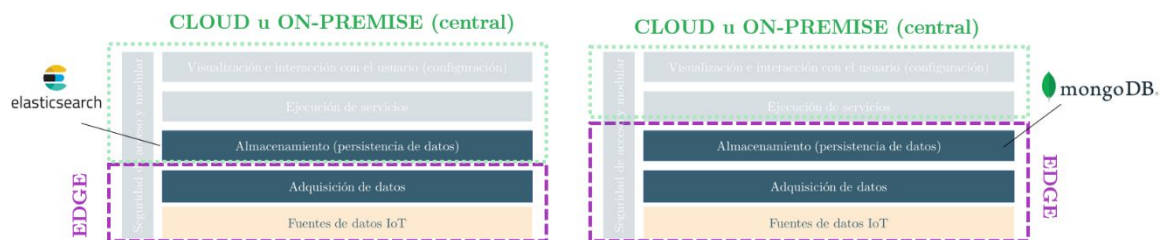
Atendiendo al análisis anterior, se eligió como tecnología para basar la persistencia de la arquitectura **Elasticsearch**. Es una tecnología madura y ampliamente utilizada que maneja documentos (JSON) y permite un indexado avanzado según campos de su contenido, posee una API REST extensiva (bien

documentada) que evita que haya de crearse una personalizada en la arquitectura, y permite una conexión fácil y directa con el resto de componentes. Igualmente, en caso de necesitarse, permite la utilización de esquemas clave-valor y ofrece altos grados de rendimiento y velocidad. Incluye también propiedades relevantes para la arquitectura, como un aplanamiento de datos completo, una interfaz de programación fácil de usar y capacidades de filtrado automatizadas. Además, posee una gran comunidad activa y permite una escalabilidad en caso de ser necesario. Por último, y como uno de los factores decisivos, se encuentra integrada en la pila ELK que, empleándose en esta arquitectura permite por sí sola (sin necesidad de componentes adicionales) satisfacer los requisitos de análisis de datos y visualización de datos en tiempo real a través de Kibana [304].

No obstante, Elasticsearch presenta un problema al reflexionar sobre la intención de *edge computing* de la arquitectura. En un software que requiere una gran cantidad de recursos de computación (mínimo de 8GB de RAM dedicados en un mínimo de 2 cores) [305]. Este tipo de capacidad solo puede darse en entornos *cloud* o con la instalación de servidores *on-premise* en un entorno central. Por tanto, si se busca un despliegue *edge* completo de la plataforma, no podría optarse por esta tecnología, ya que estos requerimientos superan las dotaciones habituales de los equipos *far-edge* y *edge*.

Es por ello que, a pesar de mantener Elasticsearch como la base de datos recomendada por arquitectura, se decidió crear una sugerencia de tecnología alternativa para los casos en que se desee desplegar la solución en un entorno *edge* completo. Anteriormente, en la Figura 3.24: Dualidad *edge-cloud* del módulo de adquisición de datos, se ha visto que el módulo de adquisición de datos es un claro candidato a residir en entornos de computación de borde. Para implementar el módulo de procesamiento de datos (almacenamiento de datos) en un entorno similar, se debe proceder con esta alternativa.

Se propuso, en este caso, la sustitución de MongoDB para funcionar como la base de datos de almacenamiento en la solución.



**Figura 3.27:** Alternativas tecnológicas propuestas para almacenamiento de datos

MongoDB tiene unos requisitos mucho más relajados de implementación (1GB de RAM) [306], que son cumplibles por equipos del *edge computing* [307]. MongoDB funciona con estructuras de documentos JSON, lo que llevó al candidato a escoger esta tecnología para la alternativa frente a otras como InfluxDB o Prometheus. Además, se tuvo en cuenta que MongoDB es la base de datos por defecto del gestor de contexto (ORION), así como la tecnología más extendida para las BBDD

expuestas a través de un API REST [308]. Su API para consulta e inserción de registros es realmente potente y permite escalabilidad horizontal.

En relación a los archivos complejos a almacenar (informes en PDF, modelos de *machine learning*), tanto MongoDB como Elasticsearch aceptan el registro de datos binarios (BSON) [309] [310].

Los dos inconvenientes principales que presenta la utilización de esta arquitectura alternativa son:

- MongoDB es capaz de almacenar documentos de tamaño máximo 16MB. Esto puede ser una limitación en determinados casos; no obstante, para el servicio al que se orienta esta arquitectura, aún no se ha superado ese tamaño (los escenarios de validación pueden consultarse en la sección 5).
- MongoDB, al no encontrarse integrado dentro de la pila ELK ni en otra estructura similar, no ofrece nativamente capacidades de visualización de los datos a través de otro servicio incorporado (como sucedía con Kibana). Este punto requiere la adición de nuevo software para satisfacer el requisito funcional de observación de datos en tiempo real y análisis de datos históricos. Este requisito puede relajarse, o no, en función del escenario de validación.

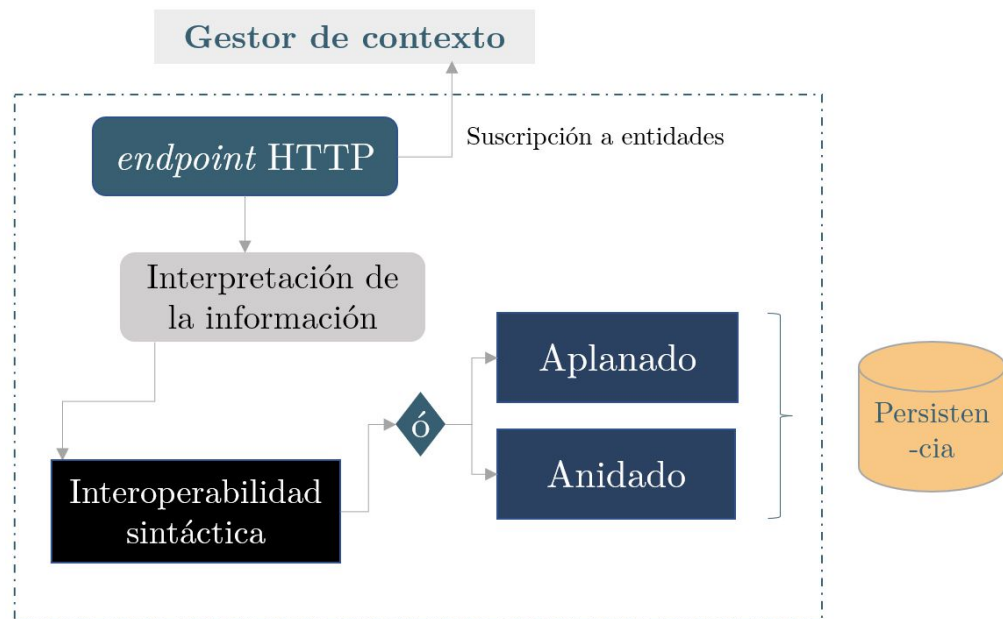
Además de la base de datos en sí, para popularla se requiere un componente adicional (el proveedor de persistencia señalado en la Figura 3.26: Módulos de procesamiento de datos: división), que se encargará de insertar cada nuevo registro que llegue al gestor de contexto (puerta de entrada del sistema) a dicho almacenamiento persistente.

Es decir, para poder almacenar un histórico de todos los cambios en todas las entidades del gestor de contexto, que solo almacena el último valor de cada una de estas entidades, se debe incluir en la arquitectura un componente que sea capaz de atender a las actualizaciones de dichas entidades y asegurarse de que cada cambio es persistido, asegurándose de incluir un registro temporal. Esta actuación consiste en almacenar un histórico de versiones de todas las entidades para tener una referencia de los cambios que se han realizado a lo largo del tiempo y para que sirva de información para el módulo de aprendizaje automático.

Para la consecución de este objetivo, el subcomponente de proveedor de persistencia debe ser capaz de interactuar con el gestor de contexto (actualizada en tiempo real) y actuar de puente hacia la persistencia en el sistema. Para su implementación, la primera referencia fue acudir a los mecanismos nativos de FIWARE, preparados para realizar esta interacción con ORION y garantizar la persistencia de datos. En este sentido, FIWARE dispone de dos enablers a tal efecto: STH Comet [311], orientado a la persistencia a corto plazo (*Short Term Historic*), reteniendo la información durante un día o una semana para la agregación de los últimos datos; o LTS Cygnus [312], orientado a la persistencia a largo plazo (*Long Term Storage*), que es lo que se busca en la arquitectura de la

tesis. Este segundo *enabler* se trata de un módulo de la iniciativa FIWARE basado en la tecnología Apache Flume [313], encargado de trasladar la información en los modelos de datos de ORION a los modelos esperados por una gran variedad de bases de datos (CKAN, DynamoDB, PostgreSQL, MongoDB, Elasticsearch, Carto...). Funciona a través de un mecanismo de fuente (*source*) y sumidero (*sink*) donde para cada base de datos añadida se incluye un componente específico.

Se trataba de una elección fácil e idónea, no obstante, en la práctica decidió optarse por otra vía. FIWARE Cygnus requiere un elevado uso de recursos (1GB de RAM) lo que dificulta su convivencia con el resto de módulos de la arquitectura. Por otro lado, es una herramienta orientada a dotar de flexibilidad en la elección de base de datos, pudiendo ser sustituidas a lo largo del tiempo a través de dichos conectores. Para la solución de la tesis, se escogió una tecnología en particular (Elasticsearch) junto con otra alternativa (MongoDB), limitando a dos el número de “conectores” que podrían usarse. Por otro lado, su curva de aprendizaje, junto con lo expuesto anteriormente, llevaron al candidato a proponer proveedores de persistencia a medida para ajustarse a la implementación esperada.



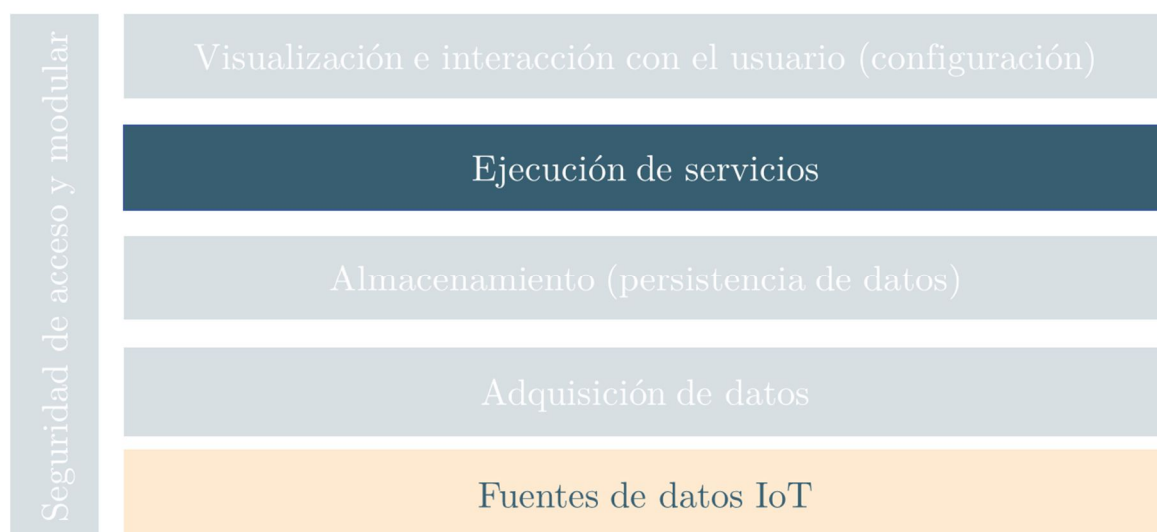
**Figura 3.28:** Alternativas tecnológicas propuestas para almacenamiento de datos

Debe existir por tanto un proveedor de almacenamiento para el caso general y otro para el caso *edge computing*. A nivel tecnológico, de cara a la validación de la arquitectura este componente se materializó utilizando Node.js, debido a su ligereza y poco consumo de recursos. Este componente implementa un *endpoint* HTTP que es suscrito a todas las entidades existentes en ORION. De este modo, cada vez que ocurre una actualización en cualquiera de ellas, este módulo (que se encuentra escuchando en un puerto web seleccionado), transforma la información a los modelos JSON esperados, aplanando o anidando la información correspondientemente (Elasticsearch o MongoDB) y asegurándose la persistencia.

### 3.4.2. Ejecución de servicios

La segunda parte del módulo de procesamiento de datos es una de las más importantes de la arquitectura. Para obtener valor añadido de los datos almacenados, la solución debe permitir la ejecución de servicios sobre los mismo (en este caso, de cálculo y predicción de CIs). Este componente se encarga de diseñar e implementar esta ejecución.

En la siguiente figura se resalta su posición dentro de la arquitectura propuesta:



**Figura 3.29:** Submódulo de ejecución de servicios de la arquitectura propuesta

Dentro del amplio concepto de “servicio” en una arquitectura IoT, para el caso de esta tesis doctoral se entiende como aquellas entidades que otorgan un valor al usuario, procesando un software de simulación o predicción con diferentes parámetros de entrada. En definitiva, es el punto donde se lleva a cabo la “inteligencia” del sistema.

Según los principios de diseño y los requerimientos técnicos, la arquitectura debe ser flexible, escalable y modular. Además, al seguir un paradigma de orientación a microservicios, se entendieron estos “servicios” como piezas de software independientes que pueden ser instanciadas por la plataforma siguiendo mecanismos acordados de orquestación. Además, según las disposiciones en la sección 3.2.1, se escogió Docker como la tecnología para la virtualización de los módulos y funcionalidades de la arquitectura.

Por tanto, se infirieron las siguientes tres condiciones para la ejecución de dichos servicios:

- Los módulos de la arquitectura han de comunicarse entre sí a través de APIs, por principios de diseño. Por tanto, se escoja el mecanismo

que se escoja, se debe asegurar que cualquier interacción entre módulos se efectuará de esta manera.

- Al tratarse de componentes containerizados, una posible propuesta para este submódulo sería contar con un *framework* de gestión de contenedores.
- Debe tenerse en cuenta que esta arquitectura está siendo diseñada para la ejecución de dos servicios: (i) cálculo y (ii) predicción de índices compuestos. Aunque sería interesante permitir dar cabida a mayor número y variedad de ellos, estos principios deben guiar la elección en este aspecto.

Por tanto, en este punto se analizaron varias opciones para recomendar una implementación en la arquitectura.

En primer lugar, se analizó la posibilidad de utilizar herramientas disponibles en abierto para la gestión de contenedores (conocidas como *frameworks* de orquestación de contenedores) [314]. Dentro de estas destacan, principalmente, Kubernetes (la más utilizada con diferencia) [151], Docker Swarm [152], Rancher [315] o Apache Mesos [153]. OpenShift [316] es también ampliamente utilizada, pero sobretodo en entornos empresariales cerrados, ya que se centra en la seguridad y es de pago. Este tipo de herramientas están principalmente enfocadas a la escalabilidad de despliegues de servicios basados en contenedores (habitualmente, Docker) en entornos de *cloud computing*. En estos entornos, muchas máquinas (físicas y virtuales) pueden unirse en *clusters* que son gestionados (orquestados) por estas herramientas, ofreciendo una excelente capacidad de mantenimiento, soporte en tiempo real, integración CI/CD, replicabilidad, continuidad de servicio, balanceo de carga, optimización de recursos, virtualización completa, etc. En general, están pensados para una gestión de una gran infraestructura (potencialmente deslocalizada), centrándose menos en el contenido de los (micro)servicios que despliegan sobre las mismas. Para el caso de esta tesis, la herramienta que se consideró potencialmente interesante para el mecanismo de ejecución de servicios fue Docker Swarm, por su nativa cercanía al motor de Docker y a su API y por ser ligera. No obstante, su objetivo principal (unificar ejecuciones de contenedores para gestionarlas como un único entorno, centrado en las réplicas y requerimientos de recursos) difiere sustancialmente de lo que se persigue en este módulo de la arquitectura: ordenar la ejecución de los servicios (contenedores) y conseguir una interacción satisfactoria con el resto de elementos de la solución para proveer el valor añadido.

Dentro de este mismo grupo se podrían incluir herramientas como Kong [317] o KrakenD [318], que hacen uso de los *frameworks* anteriores y de la API de Docker, pero aportando una interfaz visual que facilita el arranque, parada, gestión y mantenimiento de los contenedores. Estas soluciones son interesantes para la solución planteada pero plantean dos grandes inconvenientes: (1) no son herramientas ligeras (no aptas para entornos *edge computing*) y añadirían complejidad a la pila tecnológica de la solución, y (2) se trata de gestores de APIs,



lo cual significa que los servicios (de cálculo y predicción de CIs) deben implementar una API para poder ser interactuados (mayor carga y complejidad al sistema).

En segundo lugar, y como la opción más interesante, se encuentra el módulo de *Operational Tools* del proyecto PIXEL [319]. Se trata de la opción más directa ya que el diseño de la arquitectura de la tesis viene en gran parte inspirado por la participación del candidato en el mismo. Por tanto, los planteamientos tecnológicos se encuentran alineados. En el proyecto PIXEL también se manejó el concepto de servicio (denominado modelo y/o algoritmo predictivo) siendo un software “dockerizado” a gestionar por parte de la plataforma IoT (cabe recordar que la solución de esta tesis no es una plataforma). Las características y propiedades de este software (desarrollado en Java) son las siguientes:

- Publicar modelos y/o algoritmos predictivos.
- Editar y configurar los modelos y/o algoritmos predictivos.
- Ejecutar los modelos y/o algoritmos predictivos.
- Programar los modelos y/o algoritmos predictivos para que se ejecuten en un momento determinado una vez o periódicamente.
- Definir diferentes indicadores propios (KPI) operacionales y ambientales, basados en datos específicos disponibles en el centro de información para fines de seguimiento y monitoreo.
- Establecer algún mecanismo de detección de patrones. El más básico es el uso de disparadores.
- Obtener las tendencias de un modelo y/o algoritmo predictivo (por ejemplo, datos históricos).

En cuanto a su funcionamiento tecnológico, se trata de un software complejo desarrollado en Java que interactúa con la API de Docker para realizar operaciones sobre dichos servicios (modelos). Posee una interfaz gráfica compuesta por formularios anidados que permite la publicación de modelos dentro de la plataforma del proyecto PIXEL. Para ello, los modelos deben estar disponibles públicamente en el repositorio abierto DockerHub [320]. Tras su publicación, los modelos son ejecutados por dicha herramienta (orquestración) utilizando la API de Docker, pasando como parámetro a dichos servicios un archivo JSON (alrededor de 150 campos) que proviene de un formato y una información acordada previamente (esta información se introduce a través de un formulario web). La herramienta permite una selección de la periodicidad de ejecución (de si debe ser en tiempo real). Esta es una característica fantástica que fue considerada como muy interesante para la arquitectura de la solución. Esta herramienta se encarga también de gestionar los resultados de los servicios.

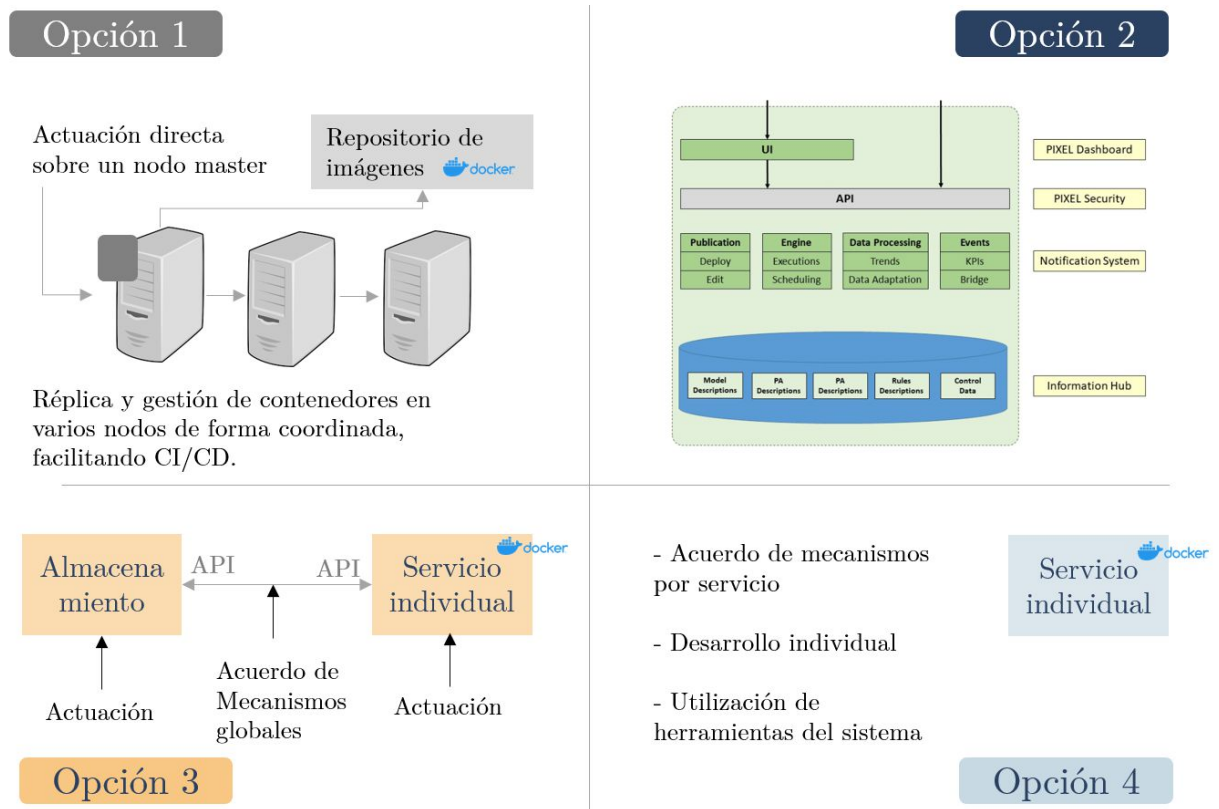
Como se ha visto, se trata de un excepcional candidato. No obstante, decidió no optarse por recomendar esta implementación para la arquitectura de la tesis por los siguientes motivos:

- El nivel de madurez tecnológica es reducido. Únicamente se ha utilizado en el proyecto de investigación donde se diseñó y en un número limitado de escenarios.
- Requiere la publicación en abierto de los servicios (en un repositorio público), lo cual dificulta ciertos aspectos relacionados con el modelo de negocio (véase sección 6). Además, este aspecto dificulta una correcta ejecución de la metodología DevSecOps que requiere pruebas y test sobre MVPs más ágilmente.
- Es una herramienta orientada a la gestión de gran cantidad de servicios. La arquitectura de la tesis está principalmente orientada al cálculo y predicción de CIs, por tanto la utilización de esta herramienta estaría añadiendo una complejidad innecesaria a la solución.
- Esta herramienta está incluida dentro de la misma visualización y espacio donde se encuentra el resto de la plataforma. La utilización de la misma requiere intervención altamente personalizada por parte de personal técnico (conocedores del nombre y *tag* de la imagen del modelo en DockerHub a elegir o de la URL donde se encuentra disponible la API del módulo de almacenamiento de datos) y del personal asociado a la interpretación del servicio.

En tercer y cuarto lugar, se analizaron opciones personalizadas que no pasaban por la utilización de herramientas de terceros sino por el desarrollo de mecanismos propios ajustados a la arquitectura en cuestión. En particular, se estudiaron dos posibles escenarios:

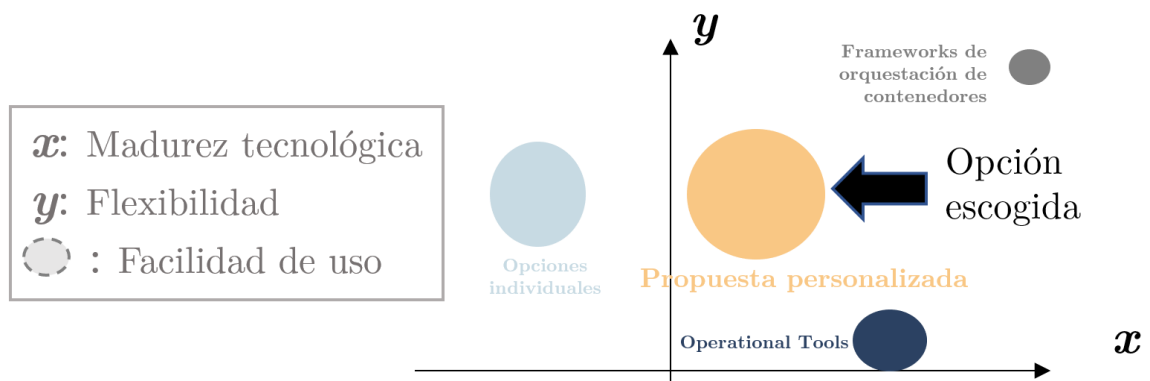
- Establecer una comunicación directa entre el servicio y los datos sin necesidad de mediación por parte un elemento orquestador. Se plantea un caso en que la orquestación se materialice en un mecanismo de acuerdos en el formato y contenido de la información a compartir (más bien, la información necesaria por parte del servicio para interactuar con la base de datos). Esta aproximación es razonablemente flexible ya que la comunicación se realiza a través de las APIs de los diferentes componentes (almacenamiento de datos y servicio) siempre y cuando ambos hayan garantizado una pequeña preparación por su parte.
- Una programación específica de cada servicio con la información necesaria para la interacción con la base de datos. Esta opción resulta cómoda (y apropiada ya que solo se esperan dos servicios) pero no se adhiere a los principios de diseño de separación de funciones, modularidad ni cumple con buenas prácticas, ya que la configuración se hace directamente sobre el código de cada servicio.

En la siguiente imagen se muestra un pequeño resumen de las reflexiones obtenidas tras el estudio de las posibles herramientas para la ejecución de servicios en la arquitectura:



**Figura 3.30:** Resumen del análisis de opciones para la ejecución de servicios

Para ofrecer una perspectiva alternativa sobre la elección de la herramienta (opción 3), se presenta la siguiente gráfica. En ella se comprueba que fue una decisión basada, principalmente, en la facilidad de uso y en la adecuación a los servicios y espíritu de la arquitectura de la tesis. No obstante, se han encontrado herramientas que proporcionan esta funcionalidad mucho más maduras y probadas (por ejemplo, Docker Swarm) y otras más flexibles y completas (*Operational Tools*), pero fueron descartadas por los motivos expuestos anteriormente.

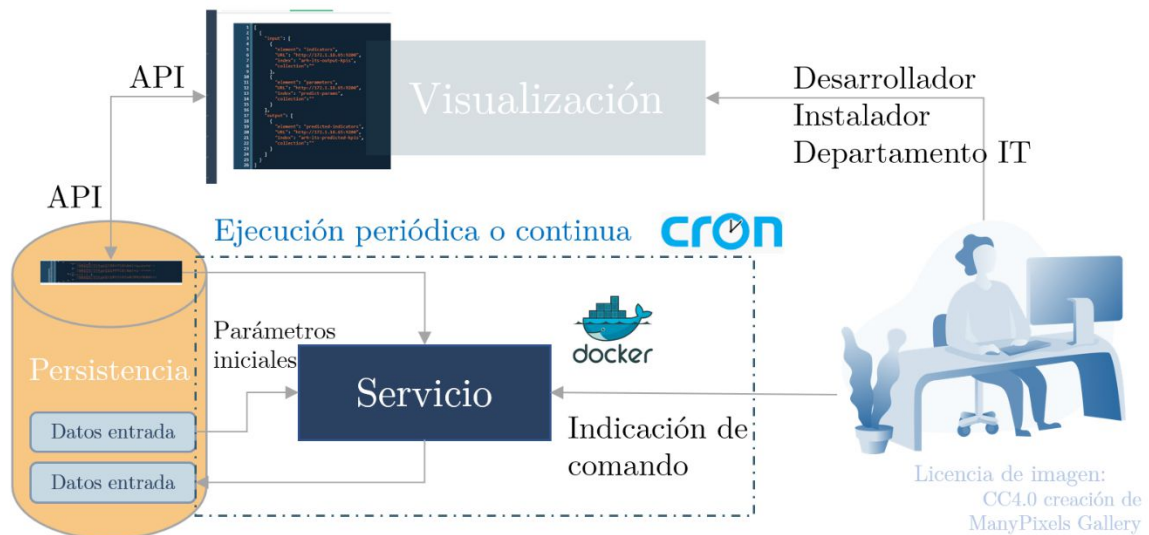


**Figura 3.31:** Perspectiva de las opciones de ejecución de servicios analizadas

Otro de los puntos para tomar esta decisión fue la descripción del requisito funcional “Instanciación de servicios”, en el que se lee que se debe implementar una instanciación ágil basada en una metodología clara. La solución propuesta presenta una metodología clara y simple basada en cuatro premisas.

1. Intervención del equipo técnico (desarrolladores, configuradores de la instalación) en dos puntos diferentes de la arquitectura.
2. Acuerdos de formato para el intercambio de la información de dicha configuración.
3. Comunicación a través de APIs.
4. Aprovechamiento de las herramientas del sistema.

En la siguiente imagen se presenta la implementación propuesta para este submódulo, aportándose consiguientemente unas pequeñas notas sobre su materialización sobre la arquitectura.



**Figura 3.32:** Orquestación simplificada propuesta en la arquitectura ([imagen](#))

El desarrollador utilizará un espacio específico en la GUI (*Graphic User Interface*) de la solución para introducir una serie de información técnica requerida por el servicio para funcionar (dirección de la base de datos, índices...). Esta información se puede encontrar en la sección 4.2, donde se detalla la materialización de estos espacios para cada uno de los servicios desarrollados. Esto dota de flexibilidad al mecanismo ya que podría cambiarse la base de datos, los índices u otros parámetros y mantener intacta la integridad de la imagen del servicio.

Esta información es guardada en un índice determinado de la base de datos. El valor de este índice es pasado como parámetro al servicio cuando éste es ejecutado. Este paso se realiza también por desarrollador/instalador/mantenedor del servicio, aislando así la interacción de diferentes usuarios con la arquitectura. El trabajo consiste en introducir una nueva entrada (línea) en la herramienta *cron* [321], nativa de Linux, del sistema que aloja la solución. Dicha línea contiene el

comando Docker de ejecución del contenedor (*docker run*), informando del nombre de la imagen (que debe encontrarse instalada en el equipo) y el paso de dicho parámetro.

```
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
* * * * 0 sudo docker run cic_service -e DB_URL='http://172.1.18.65:9200' DB_INDEX="ckpi-config"
```

**Figura 3.33:** Instrucción (comando) para la ejecución planificada del servicio

En este punto, conviene hacer dos comentarios relevantes. Esta línea podrá diferir en dos dimensiones:

- En la sección anterior se ha mencionado que se proponen dos implementaciones del almacenamiento de datos diferentes, en función de si se trata de un despliegue orientado al *edge computing* o no. En caso de así serlo, en este punto se deberá instanciar el servicio alternativo (conectando con MongoDB en lugar de con Elasticsearch).
- Cron cubre ciertas periodicidades siguiendo una nomenclatura (véase sección 5.2.4). Como puede observarse, la frecuencia mínima indicable es 1 minuto. Por tanto, la ejecución en tiempo real del CI, siguiendo este diseño de aplicación, sería con una periodicidad minutal. Aunque este tiempo podría ser aceptable para un servicio retrospectivo, la arquitectura no estaría cumpliendo los principios de *Tactile Internet* planteados en la sección 2. Por ello, en caso de que el servicio requiera una frecuencia de ejecución más alta (por ejemplo, cada segundo), deberán marcarse todos los espacios como ‘\*’, de tal manera que se trate de una ejecución continua (desde el punto de vista del sistema) y que sea el propio servicio quien regule su periodicidad según las capacidades del *host* y del lenguaje.

Una vez instanciado, el contenedor se ejecutará leyendo los parámetros pasados, alcanzando así la base de datos y leyendo el índice oportuno. En el mismo, encontrará la información necesaria para obtener los datos de entrada, efectuar su funcionalidad y almacenar el resultado en el lugar oportuno de la misma.

En cuanto a dicha funcionalidad (la de los servicios) se expone con detalle, así como su implementación tecnológica en la sección 4.2.

### **Recapitulación del módulo de procesamiento de datos:**

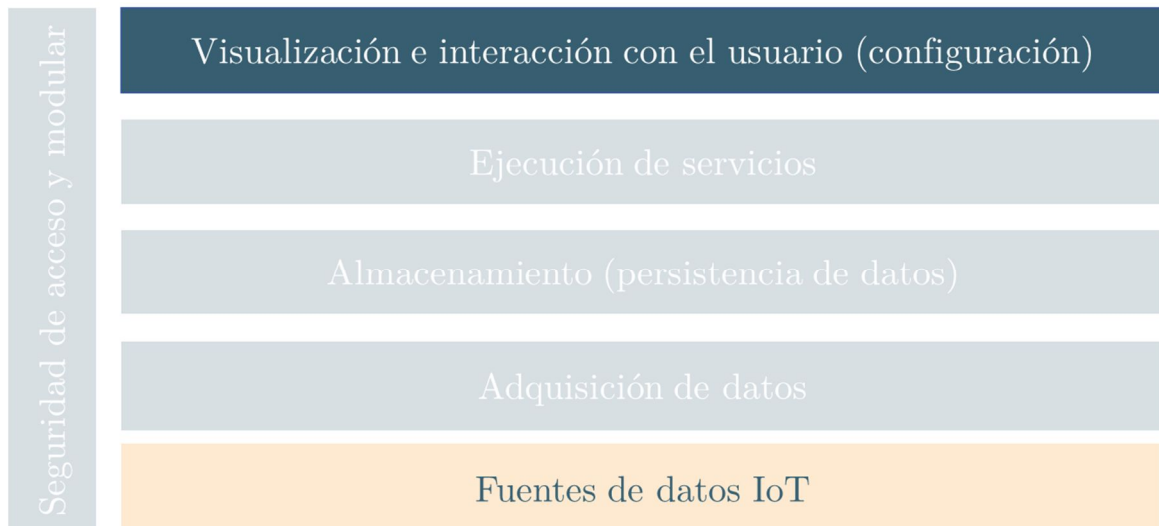
EL módulo de procesamiento de la arquitectura se divide en dos grandes submódulos. Por un lado, la persistencia a largo plazo de los datos es conseguida con un almacenamiento que, según estudiado, recomienda implementarse con la tecnología Elasticsearch. Considerando los exigentes requisitos de la misma, aquellos despliegues que requieran la presencia de este módulo en entornos de *edge computing*, deben optar por la base de datos MongoDB. Con respecto a la

realización efectiva de los servicios, se recomienda utilizar un planteamiento de eficiencia tecnológica, descartando soluciones más complejas como Docker Swarm o el módulo *Operational Tools* de PIXEL. Se propone la comunicación directa entre servicios (encapsulados en contenedores Docker) con la persistencia del sistema a partir de mecanismos acordados, la realización configuraciones por parte de los desarrolladores y aprovechar la utilidad de los sistemas Linux llamada *cron*.

## 3.5. Visualización

Los resultados obtenidos tras la ejecución de los servicios pierden su valor si no son capaces de ser representados de forma atractivamente visual para el usuario final. Este módulo tiene como objetivo ofrecer esta información, al igual que permitir al usuario de alto nivel configurar las condiciones del servicio deseadas.

En la siguiente figura se resalta su posición dentro de la arquitectura propuesta:



**Figura 3.34:** Módulo de visualización dentro de la arquitectura propuesta

Este módulo en la arquitectura proporciona una funcionalidad de GUI (*Graphic User Interface*) que satisfaga los requisitos funcionales y de servicio expresados en la sección 3.1.1. En este sentido, se especifica que el sistema debe ser usable por todos los perfiles a identificar en este tipo de despliegues IoT (desde el usuario final a los profesionales de desarrollo/mantenimiento). Además, debe permitir la observación de datos en tiempo real, la introducción manual de datos y la visualización (a través de una interfaz web) de los resultados de los servicios sobre el IoT. Por otro lado, se determinan una serie de funcionalidades de visualización asociadas a los servicios de cálculo y predicción de CIs que son tratadas al final de esta sección.

Al ser la visualización de información de la base de datos el componente principal esperado, se comenzó el diseño de este módulo con la indagación de las

tecnologías más utilizadas en el estado del arte que podrían encajar en los objetivos de la arquitectura.

Tras un primer análisis, se detectó que la decisión podía tomar tres caminos diferentes:

1. La utilización de plataformas IoT que incluyen dentro de su pila de funcionalidades la visualización de datos IoT:
2. Utilidades especializadas en esta función, más conocidas como *dashboards* o paneles, que incluyen grandes características de personalización, conexión a diferentes entornos y proporcionan atractivo visual. Este punto presenta una gran variedad y cantidad de herramientas, tanto en abierto como de pago.
3. La creación de un módulo de visualización a medida a partir de *frameworks* de desarrollo web disponibles.

Con respecto al primer camino, el número de plataformas IoT existente es abrumador [322]. Según fuentes, pueden encontrarse más de 600 en la actualidad [323]. En la sección 2.2 se realiza una pequeña introducción sobre este aspecto y los objetivos que persiguen las plataformas IoT. Con respecto a sus capacidades de visualización, de entre las más conocidas de ellas, se realizó un pequeño análisis de las siguientes:

#### **De acceso libre y gratuito:**

Como más interesantes, de instalación y servicio gratuito y de código abierto se encontraron ejemplos como Thingsboard [324], ThingWorx [325] o Thingspeak [326] (esta se descartó ya que se basa en analíticas efectuadas por el programa MatLab). Otras interesantes fueron Wolkabout [327], que se consideró por su rápido despliegue y capacidad de personalización de las visualizaciones de flujos de datos en tiempo real, IoTGo [328], como una opción ligera y que incluye *firmware* a instalar directamente sobre dispositivos de IoT o Kaa [329], que propone unos conceptos de arquitectura similares los de esta tesis doctoral (virtualización, basada en microservicios), pero con una remarcada orientación a *cloud computing* y, en general, a conceptos *cloud-native* [330].

Todas las plataformas anteriores incorporan en su pila interesantes servicios de visualización, pero el principal inconveniente que presentar es que son, precisamente, plataformas completas. Muchas de sus características (conexión con dispositivos, almacenamiento, gestión de datos) ya se encuentran cubiertas (mucho más ajustadas a los objetivos de la tesis) por los módulos propuestos anteriormente. Considerando que todas ellas requieren la instalación completa de sus funcionalidades (no-separación de visualización), se optó por descartar este tipo de soluciones. No obstante, sirvieron de inspiración para el desarrollo abordado finalmente.



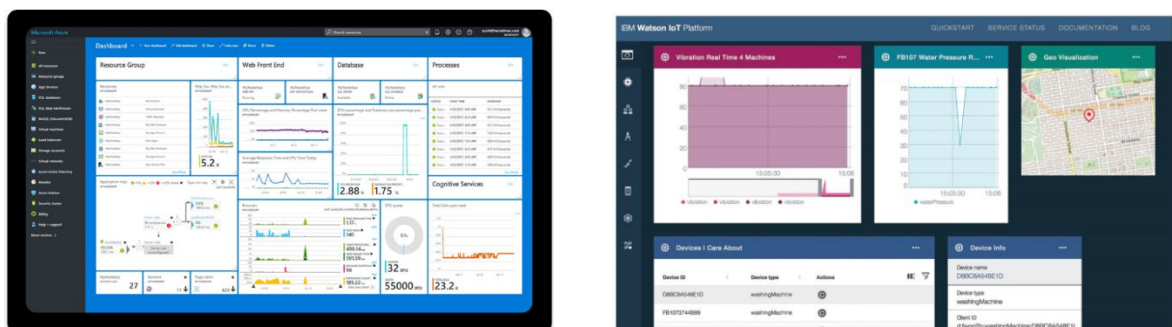
### Plataformas comerciales y de pago:

A pesar de que la reflexión recién planteada es de igual aplicación para plataformas comerciales (de hecho, con más énfasis, ya que debe pagarse para su utilización), éstas poseen una característica positiva. Al ser propiedad de grandes empresas en el sector IoT (e ICT en general) y ser mantenidas por las mismas, están muy optimizadas para dotar de una integración sencilla con multitud de sistemas externos (como sería la solución de esta tesis), ofreciendo facilidades de utilización. Además, normalmente, en despliegues discretos (a nivel tamaño, fuentes, velocidad de datos), los precios suelen ser reducidos [331]. Igualmente, otra ventaja es que permiten externalizar el soporte de infraestructura, ya que son alojadas en los entornos *cloud* de cada proveedor, permitiendo a la entidad adoptante despreocuparse del mantenimiento, etc. Por último, suelen ofrecer grandes capacidades de variedad de gráficos, agregaciones de datos, combinación de fuentes, etc. Como notas negativas, aparte de su coste e implementación como plataformas completas (solapa con la arquitectura propuesta), este tipo de soluciones suelen estar orientadas a grandes compañías industriales cuyo objetivo es observar grandes cantidades de datos; no están preparadas para interactuar con otros módulos relacionados con servicios con reglas concretas (véase 3.4.2).

En la práctica, las plataformas IoT comerciales que ofrecen un mayor interés desde el punto de vista de la visualización son:

Azure [332] ofrece el mejor servicio de composición de paneles y gráficos de visualización. Todo el ciclo de gestión de dispositivos, datos recibidos, combinación, agregación, evolución temporal, alarmas, alertas, construcción de flujos y más se encuentran disponibles entre sus servicios.

IBM Watson IoT [333] es una de las plataformas con más adopción y más consolidadas en cuanto a los despliegues IoT basados en *cloud*. Con una interfaz sensiblemente menos moderna que Azure ofrece la posibilidad de visualizar mapas, estados de los servicios (propios, a implementar según sus conexiones).



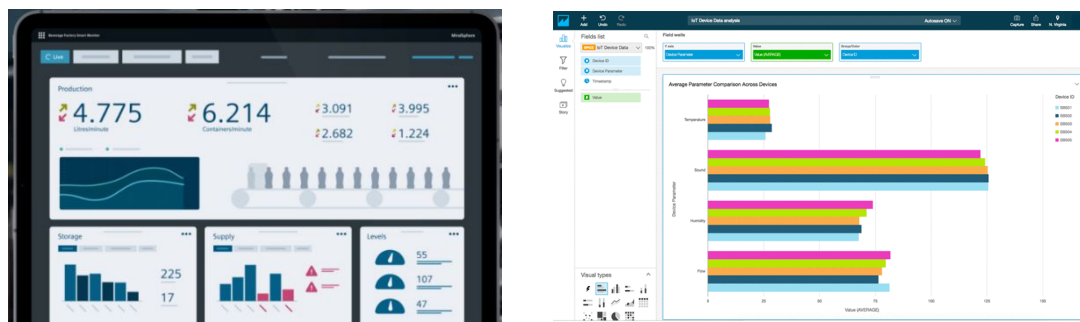
**Figura 3.35:** Vistas plataformas IoT industriales (I): Azure e IBM Watson IoT. Fuente: páginas oficiales de las compañías.

Amazon, por su parte, ofrece la plataforma AWS IoT [334] que también proporciona un servicio de visualización de aquellos dispositivos que hayan sido



conectados a través de su tecnología, enfocándose en la capacidad de actuación sobre los mismos.

MindSphere [335] es la plataforma *cloud* IoT de Siemens, cuyo entorno de visualización destaca por permitir el desarrollo web sobre el mismo, flexibilizando la personalización de la interfaz. Se trata de una alternativa interesante pero que, de nuevo, requiere la conexión de dispositivos a partir de dicha plataforma.



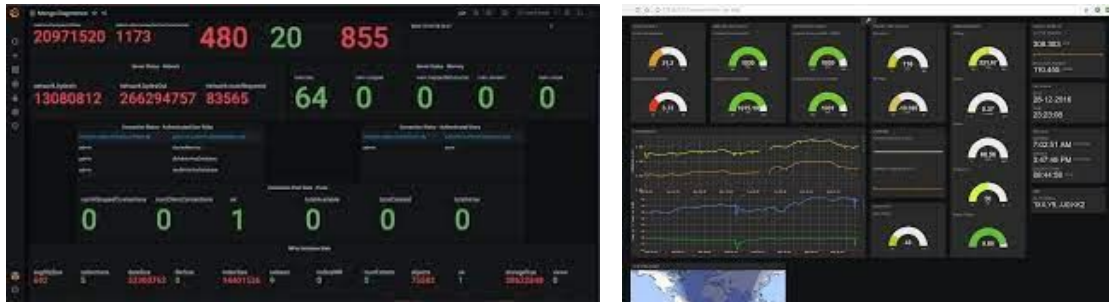
**Figura 3.36:** Vistas plataformas IoT industriales (II): AWS IoT y MindSphere. Fuente: páginas oficiales de las compañías.

Como reflexión final sobre este tipo de plataformas, el hecho de seleccionarlas como módulos recomendados en la arquitectura de la tesis supondría depender de servicios externos y requerir la interacción por parte del usuario con varias interfaces digitalmente deslocalizadas. Esto va en contra del requisito funcional “Administración centralizada de la solución”.

Con respecto al segundo camino, existe una amplia variedad de *dashboards* que pueden utilizarse sobre la arquitectura planteada:

Grafana [336] es una herramienta de código abierto tremendamente interesante para su utilización en la arquitectura. Es gratuita y permite una gran flexibilidad a la hora de crear paneles gráficos, pudiendo seleccionar varias fuentes. Su integración con diferentes bases de datos es notable, incluyendo conectores con las tecnologías recomendadas para esta arquitectura (Elasticsearch y MongoDB, y también con InfluxDB y Prometheus, otras candidatas). Su objetivo principal es el análisis de datos y monitorización en tiempo real, permitiendo la compartición (exportación) de los *dashboard* creados, incluyendo autenticación propia federada con otros servicios. Como otra característica interesante, permite la creación de alertas, por lo que un usuario podría indicar valores de referencia en los indicadores o en los resultados de los servicios para obtener notificaciones. No obstante, Grafana tiene una curva de aprendizaje muy inclinada, con una documentación limitada, lo que dificulta su configuración y extensibilidad [337]. Además, y como uno de los factores principales para no optar por Grafana, no permite nativamente la creación de visualización de mapas ni de velocímetros *-gauge meters* (por requerimiento de servicio, el CI ha de mostrarse con este tipo de gráfico).

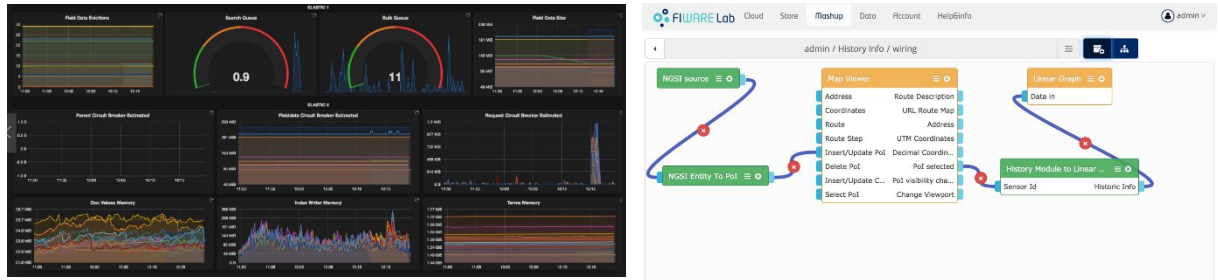
Por otro lado, una herramienta similar se encuentra en Freeboard [338], que aunque menos potente que la primera, tiene una clara orientación al Internet de las Cosas, permitiendo la observación de datos desde diferentes fuentes (muy interesante) de manera predefinida sin necesidad de crear consultas específicas. Se trata de una tecnología menos utilizada y con menor capacidad de flexibilidad. Además, no permite el desarrollo de interfaces propias para, por ejemplo, establecer mecanismos de configuración de servicios (objetivo buscado en la solución propuesta).



**Figura 3.37:** Vistas plataformas IoT industriales (III): Grafana y Freeboard. Fuente: páginas oficiales de las compañías.

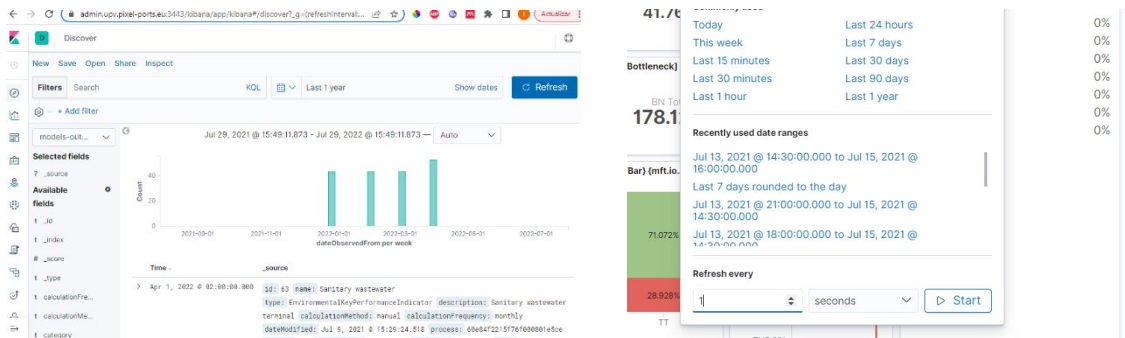
Existen también herramientas comerciales como Tableau [339] o PowerBI [340] que permiten la creación de *dashboards* muy complejos desde conectores de fuentes de muchas bases de datos, pero están orientadas a objetivos de inteligencia de negocio (Business Intelligence) y no proporcionan capacidades de configuración, orientación a IoT ni una clara integración con la arquitectura propuesta. Además, son herramientas de pago tipo SaaS, por tanto, se descartaron.

Posteriormente se analizaron dos herramientas que fueron grandes candidatas a ser recomendadas para la arquitectura. Por un lado, Graphite [341] es una herramienta *open source* creada específicamente para la creación dinámica de paneles de información. Las particularidades interesantes de esta opción fueron su clara orientación a datos de series temporales (caso de aplicación) y su potencia de cara al *edge computing*. Aunque tiene requerimientos considerables ya que incluye su propia base de datos de series temporales, es capaz de agregar información de diferentes máquinas en sus gráficas. A su favor, ofrece mucha flexibilidad y soporta una gran cantidad de lenguajes de programación. Por otro lado, el enabler FIWARE Wirecloud [335] fue seriamente considerado. Principalmente, se trata de un componente de la familia FIWARE, por lo que su cabida en la arquitectura propuesta es directa, pudiendo ser integrada con facilidad con otros módulos como la seguridad. Además, es un software libre que permite la creación de *dashboards* interpretando la información de contexto de las fuentes de datos y permitiendo la creación de flujos de información basados en los campos de las entidades gestionadas por el sistema.



**Figura 3.38:** Vistas plataformas IoT industriales (IV): Graphite y Wirecloud. Fuente: páginas oficiales de las compañías.

No obstante, la principal herramienta escogida para la visualización fue Kibana [342]. Se trata de una plataforma de análisis de datos abierta y gratuita que forma parte de la pila ELK (por tanto, incluida junto con Elasticsearch). Este fue uno de los principales motivos para escogerla. Tiene una gran capacidad para trabajar con muchos datos, y la creación de visualizaciones sobre ellos es intuitiva y no requiere conectores o consultas adicionales. Además, posee otras funcionalidades como la capacidad de crear alertas, generar informes en PDF o añadir otros *plugins* para incluir nuevas características. Es también de remarcar su capacidad para conectar con módulos independientes, a través de su API (interesante para la arquitectura). Permite, además, cumplir con el requisito de observación de datos en tiempo real, como se puede comprobar en la siguiente figura:



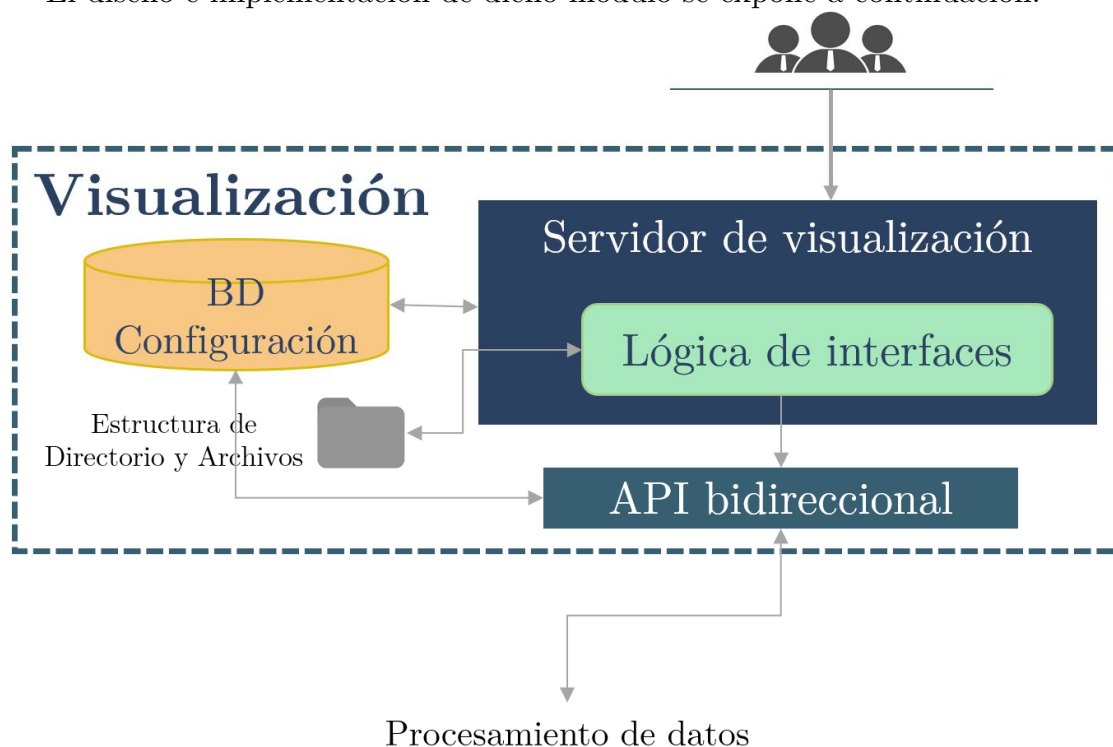
**Figura 3.39:** Herramienta de visualización de datos escogida: Kibana

No obstante, Kibana es una herramienta únicamente orientada a observar datos almacenados en Elasticsearch, sin incluir la posibilidad de configuración de servicios (aspecto interesante en la solución propuesta). La potencia otorgada por esta herramienta para la creación de gráficas queda relegada a un segundo plano en el caso de esta tesis, ya que no se indicó como requisito necesario la posibilidad de crear nuevas gráficas o visualizaciones. De nuevo, la utilización de Kibana implica que el usuario debe conectarse a otra interfaz para la observación de estos datos.

Otro componente sobre el que reflexionar en este sentido fue la visualización en casos de despliegue en entornos *edge*. Como se ha definido en 3.4.1, en los casos que se requiera este tipo de despliegue es probable la necesidad de usar MongoDB en lugar de Elasticsearch. No hay una herramienta de visualización tipo Dashboard elegible para ejecutar sobre MongoDB que tenga una clara perspectiva *edge computing*. Soluciones como *plugins* sobre Grafana [343] (que además es de pago) o el ofrecido por los propios creadores de la base de datos, MongoDB Atlas [344], solo pueden residir en entornos *cloud* (el caso de Atlas, se trata de un servicio *Software-as-a-Service* (SaaS) ofrecido por la compañía).

Por tanto, decidió optarse por recomendar Elasticsearch como herramienta de visualización de la información en tiempo real y por la creación de un módulo personalizado de visualización en la que particularizar los datos del servicio y ofrecer una experiencia de **GUI centralizada al usuario final**.

El diseño e implementación de dicho módulo se expone a continuación:



**Figura 3.40:** Estructura del módulo de visualización de la arquitectura

Primeramente, el usuario final debe ser capaz de observar los resultados del servicio de cálculo y predicción de CIs en unas interfaces claras y atractivas siguiendo los requisitos de servicio (sección 3.1.1.3). Por otro lado, el usuario debe ser capaz de configurar permisos de acceso para otros usuarios. Adicionalmente, los servicios de CI (cálculo y predicción) requieren cierta configuración (tanto por parte del usuario final como del desarrollador/instalador del sistema).

Se proponen los siguientes componentes para cubrir dichas funcionalidades. Se indican además las tecnologías utilizadas para su desarrollo de cara a los escenarios de validación.

- **Servidor de visualización:** Un servidor web. Es el elemento que permite la recepción de peticiones por parte del cliente (navegador), redirigiendo dichas consultas a los espacios del directorio de archivos correspondiente. Incluye suficiente lógica de control de acceso que es aprovechada por el resto de componentes. En los casos de validación se ha implementado con un servidor Apache [345], uno de los dos más utilizado hoy en día en Internet (en continua pugna con *nginx*) [346].
- **Lógica de interfaces:** Archivos suficientes para ser servidos tras peticiones, conteniendo información estática y dinámica (tras consultas a bases de datos). Esta lógica de interfaces hace uso de la estructura de archivos del sistema (virtualizado en contenedores Docker, en este caso) y se compone de archivos HTML, CSS y Javascript. Normalmente, los desarrollos de este tipo de interfaces se utilizan *frameworks* que facilitan su creación. Existe una amplia variedad de ellos, con diferentes ventajas e inconvenientes en función del objetivo de las páginas y de las preferencias de los desarrolladores [347]. Para los casos de validación se empleó el *framework* Vue.js [348]. En las siguientes páginas se muestran las gráficas y funcionalidades recomendadas para el sistema de cálculo y predicción de índices compuestos.
- **API bidireccional:** Se trata de un componente intermedio de cara a la interacción con el resto de módulos de la arquitectura. Los archivos dinámicos de las interfaces (*.js*) deberán realizar peticiones tanto de consulta (para representación de datos) como de inserción (parámetros de configuración) sobre la base de datos de almacenamiento del sistema. Esta API permitirá canalizar estas peticiones de tal manera que: (i) el módulo respete los principios de diseño de la arquitectura (sección 3.2.1), (ii) se siga las buenas prácticas de separación *front-end* y *back-end* web (estructura modelo-vista-controlador [349]) (iii) se eviten los problemas de CORS en las peticiones [350]. Es bidireccional porque también debe permitir que otros módulos consulten parámetros específicos almacenados en la base de datos propia. La implementación sugerida es a través de tecnologías como Flask [351].
- **BBDD de configuración:** Almacena información de configuración introducida por el usuario sobre datos relacionados con las interfaces: formularios, usuarios, permisos... Se ha materializado con MongoDB. Esta base de datos debía estar separada de la BD central del sistema para respetar el diseño de microservicios definido (sección 3.2.1).

En las siguientes páginas se ofrece una pequeña explicación sobre las visualizaciones implementadas para cubrir los requerimientos de servicio del cálculo y predicción de CIs. Se han utilizado, principalmente, componentes prediseñados de un *framework* denominado Element.js [352], que trabaja a su vez sobre el *framework* Vue.js, como se ha indicado anteriormente.

El gráfico que muestra el resultado del último CI calculado es un velocímetro principal que se acompaña con otro más pequeño informando sobre la fiabilidad de dicho índice (véase sección 4.2.4). La periodicidad de muestra de los valores depende de cómo se haya configurado el servicio.

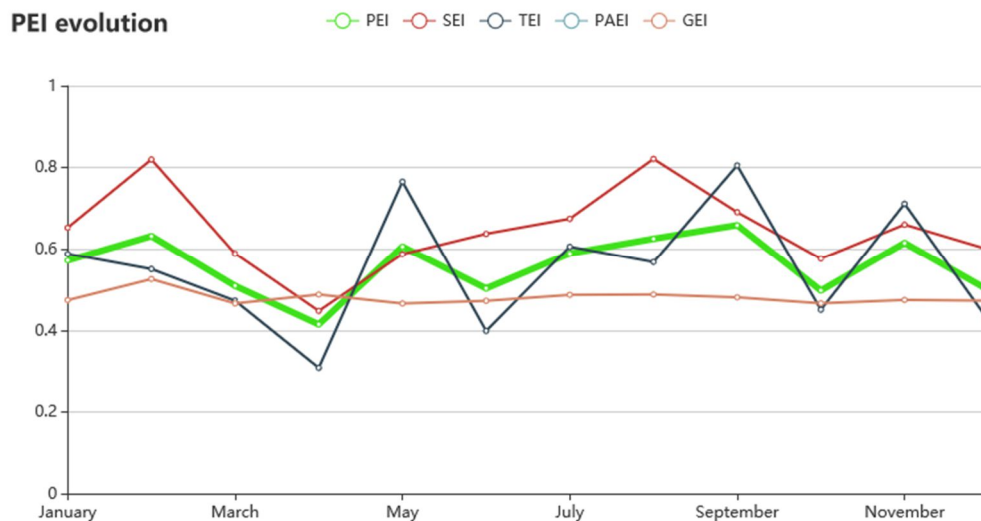


Figura 3.41: Visualización específica para servicio CI (1)

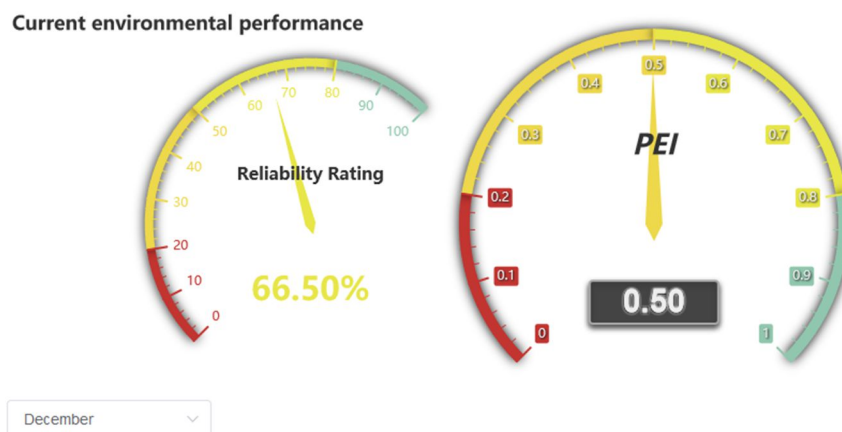


Figura 3.42: Visualización específica para servicio CI (2)

Se muestran también como ejemplos las visualizaciones de las últimas “N” ejecuciones del servicio de CI y la tabla en la que se muestra un ranking de las entidades que calculan dicho CI (si corresponde). Esta característica del servicio es muy especial y sólo se validó en un escenario (PEI – sección 5.4). No obstante, se ha incluido aquí de manera informativa. En caso de no ser necesaria, esta gráfica únicamente muestra el valor obtenido en dicho periodo para la entidad adoptante de la solución.



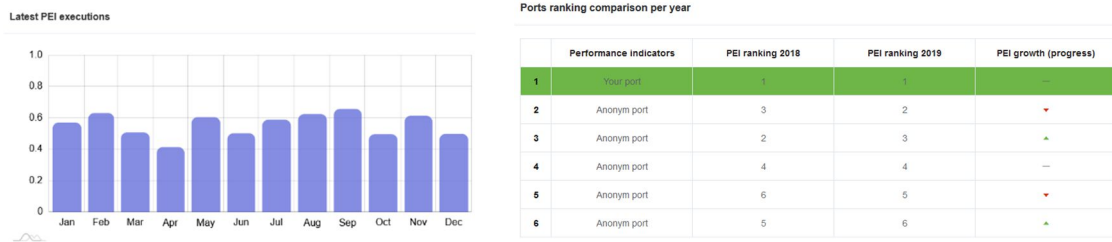


Figura 3.43: Visualización específica para servicio CI (3)

Se incluyó también una visualización radial para poder observar los indicadores resultantes del cálculo del CI, especificados por origen y seleccionables por periodo de ejecución.

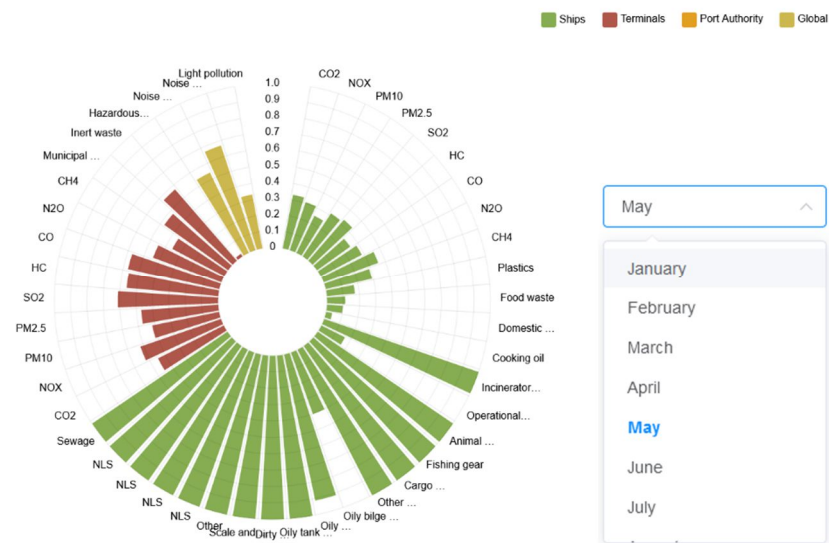


Figura 3.44: Visualización específica para servicio CI (4)

El servicio requiere también configuración de los métodos matemáticos y estructura de cálculo. Esta opción entra dentro de las opciones de configuración por parte del usuario final.

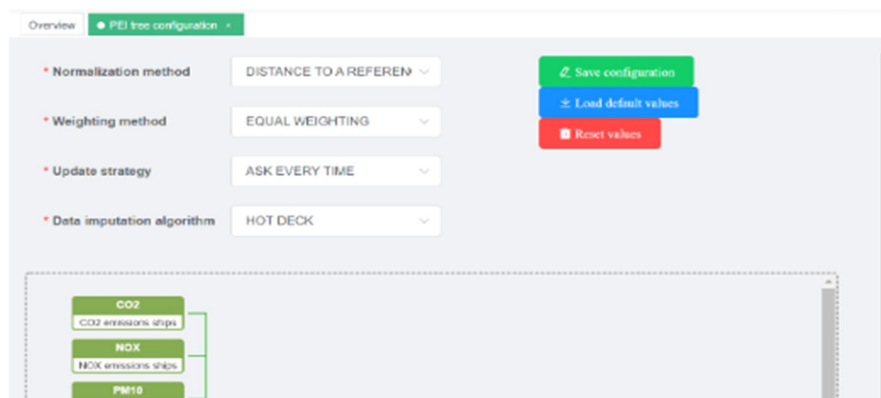
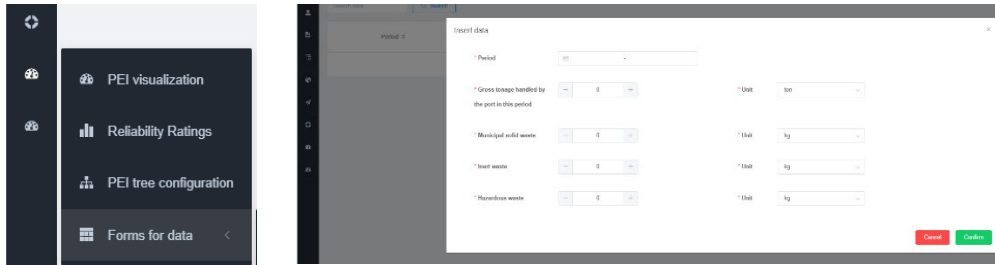


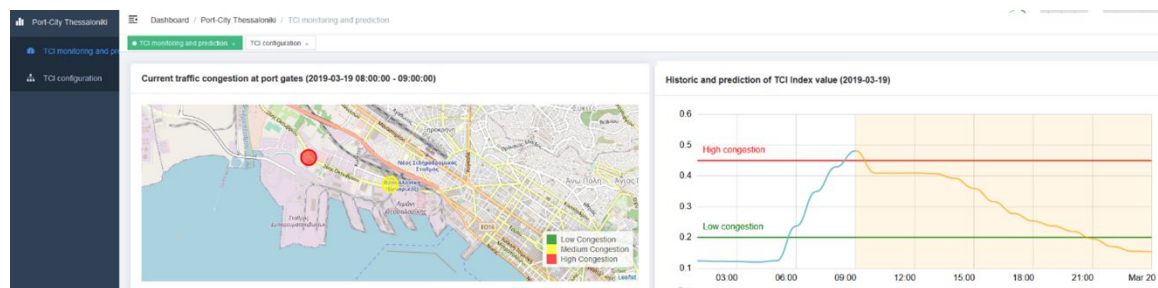
Figura 3.45: Visualización específica para servicio CI (5)

Además, según requisito funcional se debe permitir la introducción manual de información así que se habilitaron una serie de formularios que posteriormente envían la información al almacenamiento del sistema. Este componente, en caso de necesitar adiciones con formatos específicos, debería modificarse sobre la lógica de interfaces.



**Figura 3.46:** Visualización específica para servicio CI (6)

Por último, la introducción de este *framework* como recomendación en la arquitectura permite la adición de otros componentes gráficos según las necesidades del servicio, como fue en los casos de ciertos escenarios incluir un mapa (usando Leaflet.js [353] o gráficas más avanzadas de evolución temporal en varios sectores (usando amCharts [354]).



**Figura 3.47:** Visualización específica para servicio CI (7)

**Rescapitulación del módulo de visualización:**

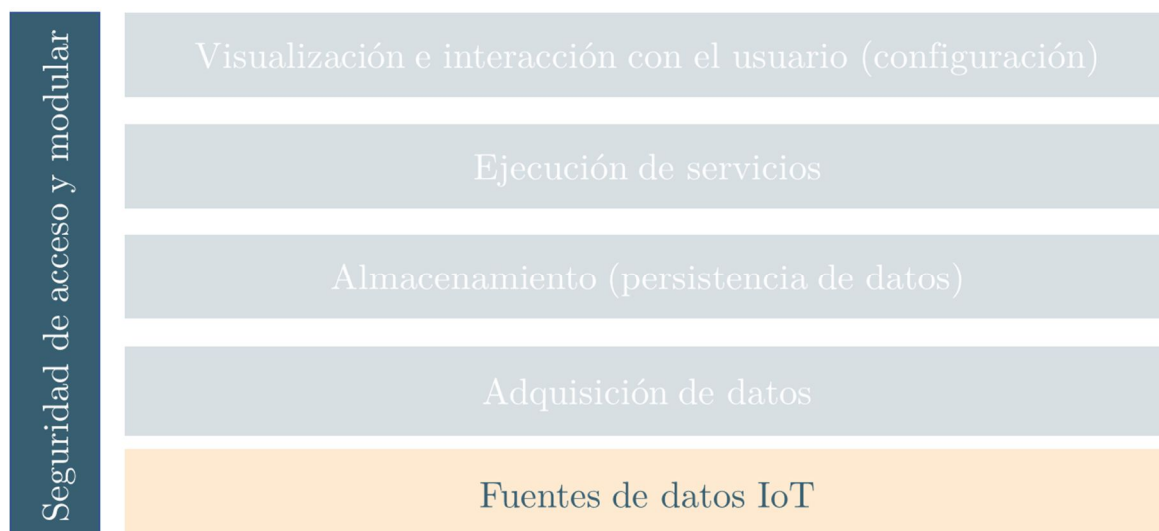
Las opciones para visualizar indicadores compuestos son variadas. Desde funcionalidades ofrecidas por plataformas IoT completas, herramientas de conexión directa a la base de datos, como Kibana [355] o Grafana [356], para una simple inspección, hasta un diseño totalmente personalizado. El estado del arte ofrece muchas oportunidades. A partir de los requisitos establecidos, la elección tenía que ser lo suficientemente personalizable como para permitir a los usuarios explorar los datos y resultados de los servicios y determinar ciertos formularios de configuración. Por lo tanto, se seleccionó el framework Vue.js (JavaScript) [348] para un desarrollo personalizado de este módulo. En este sentido, se planteó el desarrollo a medida con gráficos dinámicos para representar la información asociada a indicadores compuestos.



### 3.6.Seguridad y privacidad

En la sección 2.5.1 se expuso un extracto de la relevancia de la seguridad y la privacidad en los sistemas IoT de Nueva Generación. Proteger los servicios y asegurar cumplimiento con normativas de privacidad es primordial para que una solución pueda ser adoptada más allá de despliegues en fase de prototipo.

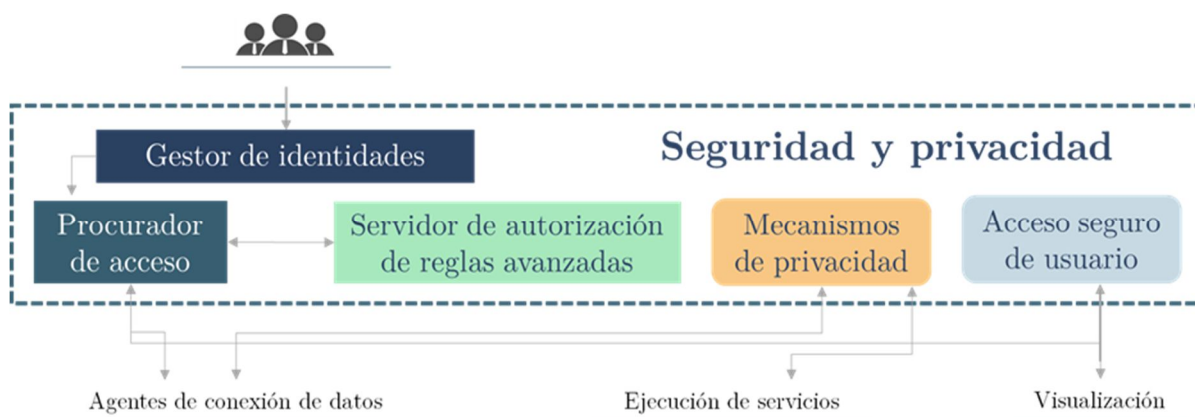
En la siguiente figura se resalta su posición dentro de la arquitectura propuesta:



**Figura 3.48:** Módulo de visualización dentro de la arquitectura propuesta

Al haberse realizado un estado del arte de las soluciones existentes y utilizadas en arquitectura IoT en el capítulo 2, esta subsección se centra en las decisiones de diseño y selección de tecnologías para implementar estas protecciones en la arquitectura propuesta en la tesis doctoral.

La siguiente imagen muestra un esquema de los componentes internos que se identificaron para este módulo, cuyo detalle es elaborado en las páginas posteriores:



**Figura 3.49:** Esquema de componentes del módulo de seguridad y privacidad

Con respecto a la seguridad, dentro de los requerimientos de primer nivel en (requerimientos técnicos – sección 3.1.1.1) se expresa la necesidad de (a) garantizar seguridad en el acceso a la solución y (b) garantizar la seguridad entre componentes. En la práctica, esto debía traducirse en:

- Integrar una solución para la gestión de identidades: usuarios, organizaciones, roles y aplicaciones.
- Asegurar que todos los accesos desde entidades externas a los componentes atraviesan suficientes comprobaciones de seguridad.

Analizando la arquitectura como estructura modular (3.7), se dividieron los requerimientos anteriores en tres acciones tecnológicas a implementar:

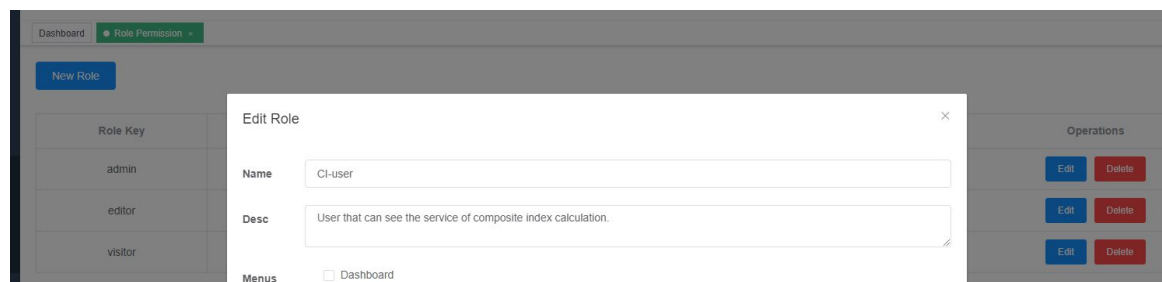
### 1. Primer nivel de control de acceso a la solución por parte de usuarios.

Se trata de proveer de una capa de seguridad al acceso a través de la interfaz web de la arquitectura. Como se ha presentado en el capítulo correspondiente (3.5), la interacción del usuario final con los servicios planteados es a través de una aplicación web gestionada por un servidor Apache.

Para garantizar seguridad desde este punto de acceso, se ha abordado desde dos puntos de vista: (1) la utilización del protocolo TLS (*Transport Layer Security*), exponiendo el puerto web a través de un certificado generado por la autoridad de certificación *Let's Encrypt* [357] (la utilización de otras entidades es posible y recomendada) y (2) el establecimiento de una serie de perfiles de usuario con acceso al sistema, utilizando para ello un mecanismo de clave usuario-contraseña y discriminación de acceso a módulos de la interfaz.

Existe un segundo nivel de seguridad (véase punto 2 a continuación) sobre el que se cimienta toda la solución, pero en el caso de la interfaz de visualización las indicaciones comienzan en este punto.

Para la creación de estos usuarios, debe primero accederse a través del rol primigenio de administración (*admin*). El usuario y contraseña iniciales de este rol se encuentran protegidas dentro de la base de datos de configuración del módulo de visualización (Figura 3.40: Estructura del módulo de visualización de la arquitectura ). Una vez accedido, este usuario puede crear el resto de perfiles y gestionar sus credenciales de acceso desde el mismo espacio. Esta información se almacena (cifrada) en dicha base de datos.



**Figura 3.50:** Seguridad en acceso de usuario: roles y permisos

## 2. Control de acceso global de la solución

Las necesidades expresadas en las primeras reflexiones del capítulo son, de hecho, cuestiones estándar tratadas ampliamente en sistemas de IoT (y generalmente en arquitecturas distribuidas [358]), por tanto, se debe plantear una selección basada en mecanismos estándar. De esta manera, se introducirá un menor riesgo dentro de la solución y los esfuerzos pueden centrarse en otros elementos más innovadores de la arquitectura [19].

Se eligió diseñar un **sistema de gestión basado en políticas** (*Policy Based Management Systems* [359]). Este tipo de sistemas establecen mecanismos en los que una petición de acceso a un recurso es consultada (a través de un elemento intermedio centralizador – *proxy*) a una base de datos central que contiene información sobre la identidad de los usuarios permitidos en el sistema y una serie de políticas (pueden llegar a ser muy complejas) para dictaminar si dicho usuario puede acceder a ese recurso.

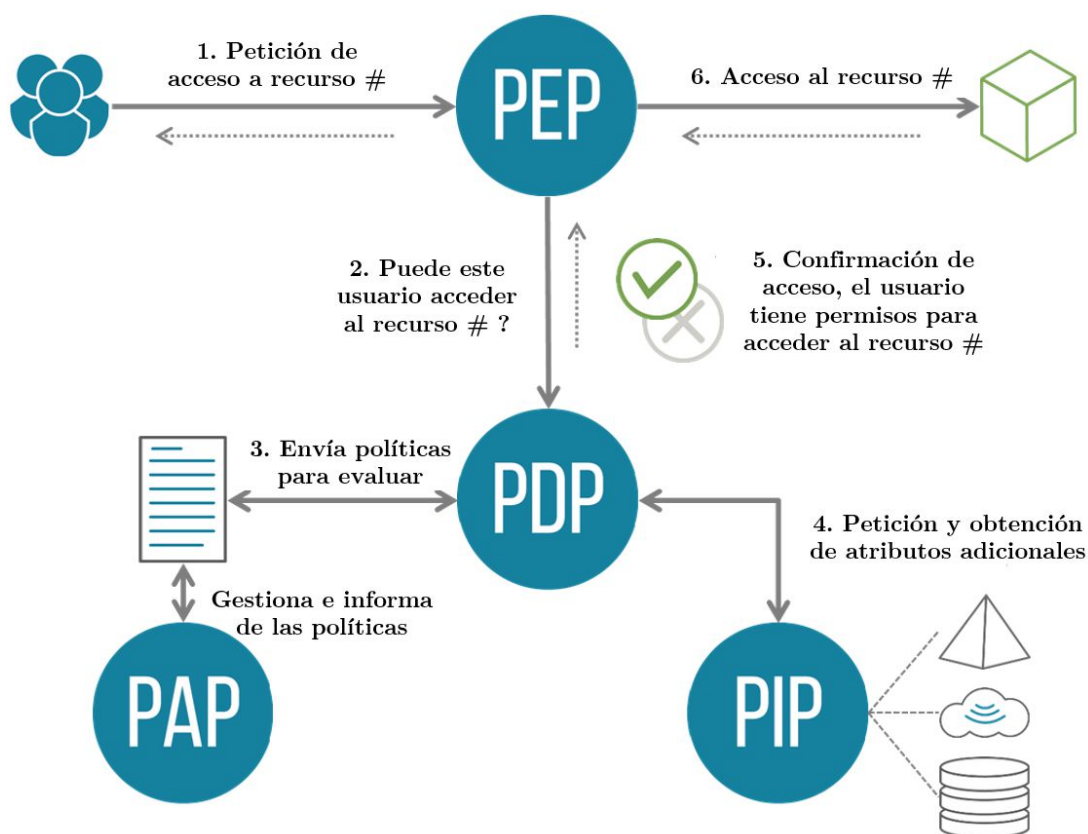


Figura 3.51: Esquema de funcionamiento de PBMS. Fuente: [359]

Estos mecanismos se materializan en la existencia de los siguientes elementos:

- **Gestor de identidades:** responsable de identificar y autenticar las peticiones, como primera fase de aceptación de la petición de acceso a recursos por parte de un usuario. Una vez validada la identidad, puede derivar la decisión al elemento de autorización. Este elemento se conoce como *Policy Administration Point (PAP)*.

- **Centro de autorización:** elemento encargado de evaluar una petición que incluye información completa del recurso al que se desea acceder y el usuario solicitante, sus roles, organizaciones y otros detalles. Este elemento se conoce como *Policy Decision Point* (**PDP**).
- **Repositorios de conocimiento:** base de dato (una o más) consultada(s) por el PDP para evaluar la decisión de política requerida, analizando tanto los atributos asociados a la petición y al recurso como las políticas establecidas. Se conocen como *Policy Information/Retrieval Point* (**PIP** and **PRP**).
- **Aplicador de las políticas:** elemento de interacción principal con el usuario, encargado tanto de iniciar el proceso de análisis de la petición como de aplicar la decisión inferida. Se conoce como *Policy Enforcement Point* (**PEP**).

Como se ha indicado previo al repaso de este sistema, la idea de implementación en la arquitectura es seguir mecanismos y tecnologías estándar. Por tanto, se tomaron las siguientes decisiones:

- Para la gestión de la identidad y el acceso a la API, la solución más empleada en el estado del arte es usar el protocolo **OAuth2** [360], que proporciona un marco para autenticar a usuarios y dispositivos (en el caso de esta arquitectura, agentes conectores de datos) y para conceder un acceso limitado para una aplicación de terceros a una API. Esta es además una decisión estratégica, ya que si una compañía quisiera posteriormente integrar la solución propuesta (de una forma segura) con otras herramientas empresariales, podría realizarse mediante la adición a través de una interfaz OAuth2.0 [361].
- Para un mayor nivel de granularidad (especificando condiciones y políticas de autorización), se ha decidido utilizar el estándar **XACML** (*eXtensible Access Control Markup Language*) [362]. XACML es el principal estándar internacional de OASIS [363] para el lenguaje de control de acceso y los formatos de solicitud-respuesta, que aborda la mayoría de los casos de uso del control de acceso. Este estándar define un lenguaje declarativo de políticas de acceso de grano fino basado en atributos, una arquitectura y un modelo de procesamiento que describe cómo evaluar las solicitudes de acceso según las reglas definidas en las políticas

Una vez decidido lo anterior, la cuestión fue qué tecnología elegir para implementarlo. Al ser un tipo de sistema estandarizado, existen multitud de opciones disponibles.

El elemento PEP actúa como puerta de acceso al resto de componentes del sistema en cuanto a la identificación y seguridad tras la petición de un recurso. Es por ello que este elemento debió analizarse en primer lugar.

Existe un tipo de tecnologías específicas para la implementación del PEP conocidas como *API Gateways* [364]. Se trata de elementos intermedios HTTP (*proxy*) cuya misión es redirigir una petición al microservicio correspondiente (los elementos asociados PAP, PDP, PIP y PRP se tratan por separado). Este análisis resultó especialmente útil ya que las *API Gateways* componen un elemento clave en el diseño de sistemas microservicios basados en la exposición de APIs que deben ser accedidos por entidades externas.

A continuación, se muestra un breve resumen sobre las más interesantes (para mayor referencia del estado del arte, véase sección 2.5.1).

WSO2 API Manager es una de las soluciones más utilizadas [365]. Este *framework* se especializa en la protección de APIs (interesante para la arquitectura), implementando varios tipos de acceso basados en OAuth2.0. Mientras que la creación de APIs dentro del sistema (posee interfaz gráfica) es relativamente sencilla, se trata de una solución muy costosa a nivel computacional y que requiere una curva de aprendizaje y conocimientos para instalación [366] que excedieron (i) los objetivos de esta tesis y (ii) las necesidades de una arquitectura IoT orientada al cálculo y predicción de CIs.

Kong [317] es el producto open source de API Gateway más popular en la actualidad. Está basado en un servidor web *nginx*, una base de datos PostgreSQL donde se almacena toda la información de cada API a conectar y una serie de scripts de Lua para personalizar la herramienta. En este caso, el resto de elementos del sistema de gestión de políticas deberían de implementarse a partir de éstos, en forma de componentes adicionales (*plugins*). A pesar de que no dispone de una interfaz gráfica para su configuración (se ha de instalar también Konga [367]), sus características son muy destacables. Como inconvenientes, se trata de una solución enfocada en el enrutamiento, pero resulta poco escalable, no se ajusta a los objetivos de la arquitectura y requiere *hardware* dedicado potente para poder ser ejecutada. KrakenD [318] es la alternativa española a Kong, más ágil y ligera y fácil de instalar, gozando de una gran aceptación en la comunidad como herramienta de centralización de APIs de microservicios, contando con una buena documentación.

En el otro lado del espectro, Google ofrece su propia solución de *API Gateway*, APIGEE [368], que fue directamente descartada al no disponer de versión gratuita ni de código abierto.

Por último, se analizó APInf/Umbrella [369], una solución de ámbito más reducido pero que fue incorporada a los candidatos para implementar el elemento PEP por su integración con FIWARE. Se trata de un *enabler* oficial de la iniciativa FIWARE que implementa un *API Manager*, permitiendo agregar diferentes microservicios (actuando como API Gateway). No obstante, únicamente actúa como elemento concentrador, sin incluir control de acceso ni autorización, que debía conseguirse con otras tecnologías.

Observando las propuestas existentes, todas ellas requieren de componentes adicionales para implementar un sistema de gestión de políticas, que es lo que se

busca integrar en la arquitectura. La cercanía a los *enablers* de FIWARE en varias partes de la solución llevó al candidato a explorar el catálogo completo [268] para entender si pudiera alcanzarse este sistema de seguridad únicamente con dichos componentes.

Las opciones en este sentido fueron las tres siguientes:

- a) La utilización de APIInf como elemento PEP en combinación con el *enabler* KeyRock [370] actuando como gestor de identidad. En este caso ya se estaría garantizando una cobertura básica de los requisitos, cumpliendo con el protocolo OAuth2.0 y disponiendo de un *API Gateway* para la concentración de APIs de microservicios (módulos de la arquitectura).
- b) La utilización de un *enabler* alternativo, Wilma [371], como *proxy* de PEP por cada aplicación a proteger y KeyRock como gestor de identidad. Esta opción fue preferible ya que KeyRock y Wilma son dos productos usados en conjunción, estando integrados y siendo usables a través de una interfaz gráfica de usuario. Provee de mayor control que la opción a). Otra de las ventajas aportadas por la combinación de estos *enablers* es la federación de accesos, permitiendo una sola autenticación en el gestor de identidad que sería válida para todas las aplicaciones insertadas como PEP *proxys* (lo que se conoce como *Single Sign On* – SSO).
- c) Evolución de la opción b) añadiendo un componente adicional: AuthZForce [372] para la inclusión de las políticas XACML de cara a un control de acceso más fino. Con las dos opciones anteriores, el elemento de decisión se contextualizaba en KeyRock, permitiendo gestionar únicamente usuarios, roles y organizaciones. La introducción de un servidor AuthZ permite el enriquecimiento de la información de la petición recibida (por URLs) y la adición de políticas de acceso más complejas. Esta opción, por ser la más completa y la que permite implementar un ciclo completo de sistema de gestión de políticas, fue la escogida como recomendada para la arquitectura de la solución.

En la siguiente tabla se hace un resumen de las tecnologías de la opción escogida: c). Se expone el objetivo de cada una de ellas, su rol dentro de los mecanismos de gestión de políticas para la seguridad en la arquitectura y se realizan algunos apuntes relevantes sobre su implementación.

**Tabla 3.6:** Tecnologías elegidas para el sistema de gestión de políticas

| Elemento funcional   | Tecnología            | Roles                                |
|--|-----------------------|--------------------------------------|
| <i>Gestor de identidades:</i>  | <i>FIWARE KeyRock</i> | $\frac{1}{2}$ <i>PDP + PIP + PRP</i> |
| KeyRock actúa como el servidor OAuth2.0 en la estructura propuesta. Es decir, contiene la información sobre usuarios, aplicaciones, roles y organizaciones. Es el elemento central de control de acceso en el sistema escogido. KeyRock ofrece |                       |                                      |

funcionalidades para la creación, edición, eliminación y actualización (operaciones CRUD [373]) de dichos atributos a través de una API REST y de una interfaz gráfica. KeyRock permite la federación de identificación entre múltiples aplicaciones (en conjunto con Wilma) y la funcionalidad de SSO. Para ofrecer estos servicios, cuenta con una base de datos MySQL donde almacena de forma encriptada la información de usuarios, roles. Se aprovecha también esta BBDD para almacenar políticas provenientes del elemento de autorización.

| Elemento funcional            | Tecnología          | Roles      |
|-------------------------------|---------------------|------------|
| <i>Aplicador de políticas</i> | <i>FIWARE Wilma</i> | <i>PEP</i> |

Este elemento ofrece una seguridad basada en OAuth2.0 a aplicaciones (API) específicas en conexión con componentes de identificación y autorización. Se trata de un intermediador inverso (*reverse proxy*) HTTP que, tras la correspondiente comunicación con los elementos apropiados, deja o no deja una petición “alcanzar” el recurso demandado (acceso a una API en este caso). Dentro del esquema propuesto, debe implementarse un Wilma por cada aplicación (API) a proteger; todos ellos conectados al gestor de identidad y de políticas .

| Elemento funcional            | Tecnología               | Roles            |
|-------------------------------|--------------------------|------------------|
| <i>Gestor de identidades:</i> | <i>FIWARE AuthZForce</i> | <i>PAP+ ½PDP</i> |

AuthZForce es el elemento que permite implementar reglas avanzadas basadas en URL (Localizador Universal de Recursos) sobre el rol asociado a la identidad OAuth2. AuthzForce soporta el lenguaje CORE XACML 3.0 completo [363], por lo que permite aplicar políticas de control genéricas y complejas. Proporciona los siguientes servicios:

PDP: En combinación con KeyRock (que únicamente maneja usuarios, roles y organizaciones), AuthZForce da soporte a las decisiones de autorización calculadas por su motor de control de acceso. Expone un API para la interacción.

PAP: Expone una API RESTful descrita con un documento WSDL [374] sobre la cual se puede actuar creando, modificando, editando y eliminando políticas XACML a través de la misma.

En este punto, una vez escogida la estructura y la tecnología para el control de acceso basado en políticas, se realizó un análisis de cómo (en qué puntos, en qué orden, sobre qué módulos...) aplicarlas en la arquitectura diseñada.

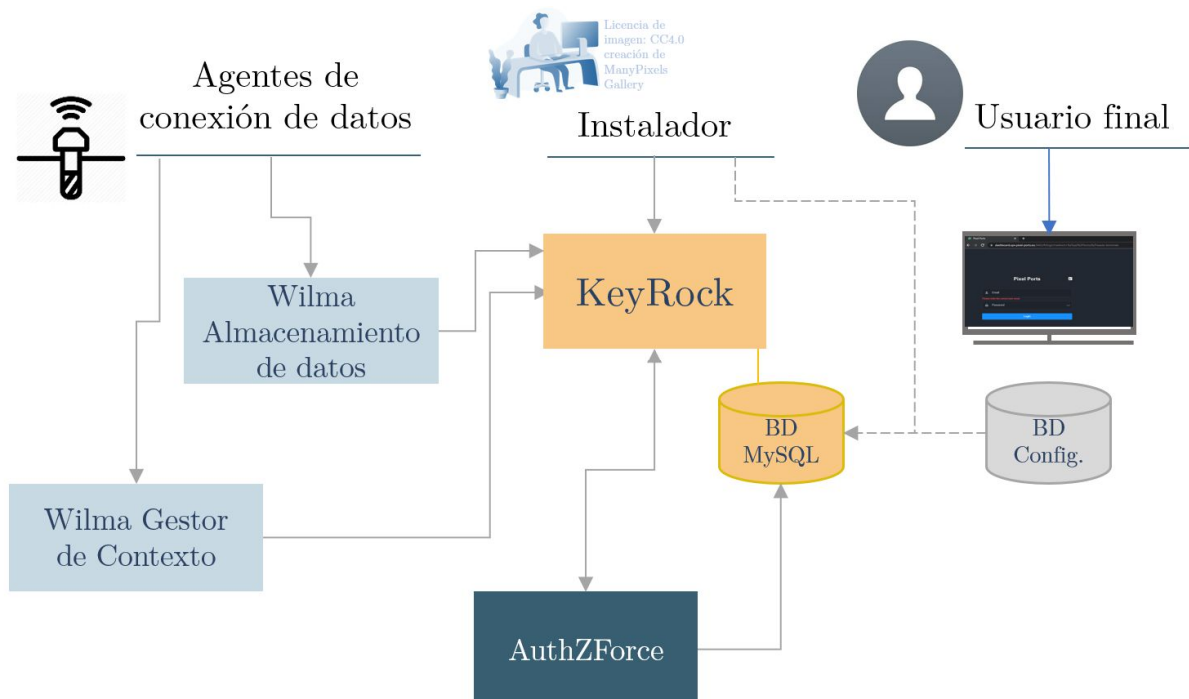
El resultado de dicho análisis se expone en los siguientes puntos:

- La existencia de conectores de datos (agentes) que pueden introducir información al sistema desde diferentes puntos (geográficamente y lógicamente) desde diferentes redes impone una necesidad de securización a abordar por parte de este módulo. Además, en este caso se trata del principal punto de acceso al sistema. La arquitectura que se plantea está orientada a la obtención de una gran cantidad de datos desde diferentes fuentes heterogéneas.



- La API asociada al módulo de visualización (API bidireccional en Figura 3.40: Estructura del módulo de visualización de la arquitectura ) no requiere su inclusión (mediante un PEP dedicado) en el sistema de gestión de políticas ya que solo podrá ser accedida por parte del módulo de procesamiento, que residirá (tecnológicamente) en el mismo espacio lógico. No obstante, con respecto al acceso de usuario, las credenciales y roles creadas desde la interfaz (figura YY), aún sin efecto en la autorización para otros módulos, debe sincronizarse con las identidades existentes en KeyRock. De esta manera, se garantiza una consistencia dentro del sistema y se podrán establecer unas políticas XACML más ajustadas. Esta acción debe realizarse de forma manual por parte de los instaladores/mantenedores de la solución.
- Atendiendo a los puntos anteriores, únicamente dos APIs (dos microservicios) de la arquitectura deben estar expuestos para la consulta por parte de entidades externas: (1) el gestor de contexto – desde los agentes conectores de datos enviando información en tiempo real- y (2) el almacenamiento de datos del sistema – desde agentes enviando datos *batch*. Por tanto, será necesaria la implementación de dos elementos PEP (Wilma). Mientras, se debe garantizar una sincronización entre las credenciales introducidas en el sistema de visualización con las existentes en el gestor de identidades (KeyRock).

En la siguiente figura se muestra un esquema resumido del sistema de seguridad diseñado para la solución propuesta:



**Figura 3.52:** Seguridad en acceso de usuario: roles y permisos



### 3. Excepciones relacionadas con despliegues *edge computing*

Existen excepciones al sistema diseñado que están relacionadas con la flexibilidad de la solución a la hora de proponer despliegues cercanos al paradigma de *edge computing*. Como se ha comentado en 3.4.1, se plantea la posibilidad de mover el almacenamiento del sistema a un entorno más cerca del origen de los datos. Este mismo principio puede aplicarse al procesamiento de los servicios y a la visualización (ambos módulos ligeros). No obstante, a nivel de seguridad estas excepciones plantean dos problemas:

1. En caso de mover el gestor de contexto a un entorno *edge computing* y mantener el almacenamiento de dato en un entorno *cloud* u *on-premises* centralizado. El módulo de procesamiento de datos dispone de un punto de notificaciones de suscripción (*endpoint*) al que se envía la información de cada actualización de entidad acaecida en el gestor de contexto. Cuando ambos módulos (gestor de contexto y almacenamiento) se encuentran alojados en el mismo *host*, este *endpoint* no plantea ningún problema de seguridad; sin embargo, al encontrarse en diferentes localizaciones (ergo, muy probablemente, en diferentes redes), dicho *endpoint* debe existir y poder ser alcanzable por el gestor de contexto de una manera segura. Para asegurar este hecho, dicho punto de acceso será expuesto mediante el protocolo SSL (*https*) y la correcta comunicación deberá garantizarse mediante: a) el establecimiento de una VPN (*Virtual Private Network*) entre ambas localizaciones (*edge* y *cloud/on-premise*) o mediante una correcta gestión de dominios y URLs a la hora de introducir el *endpoint* de suscripción en la API del gestor de contexto.

2. Cuando se desee que el sistema sea híbrido (ciertos módulos en entornos *edge* y ciertos módulos en entornos *cloud/on-premise*) y quiera establecerse una gestión de la seguridad descentralizada. Descentralización significa mover capacidad de decisión a localizaciones distribuidas. En el contexto de la seguridad de la arquitectura, esto significaría querer disponer de un servidor de autorización (PDP) en diferentes elementos de computación en diversas localizaciones. En este escenario, podrían existir instancias de estos servidores que requirieran la creación de políticas generales de acceso de aplicación a todo el sistema (funcionalidad de PAP). Al no estar centralizado, habría otras instancias que no poseerían en sus repositorios locales estas políticas actualizadas (funcionalidad de PRP). Para conseguir que el sistema se mantenga actualizado y seguro con estas premisas, existen dos opciones: (1) que el despliegue mantenga un elemento central que se encargara de recoger las nuevas políticas (enviadas desde un elemento *edge*) y distribuir las a los diferentes repositorios locales (PRPs); en este caso, el proceso de identificación y autorización sería exactamente el expuesto en la Figura 3.49, con la única diferencia que el elemento PAP y PDP se encontrarían en diferentes máquinas. O (2), en caso de ser un despliegue completamente descentralizado (sin elemento central), hacer que uno de dichos elementos actuara como “virtualmente

centralizado”, en cuyo caso las llamadas remotas a la hora de consultar los detalles de autorización se harían contra esta instancia del despliegue.

**Con respecto a la privacidad,**

Como se ha destacado en las conclusiones del estudio del estado del arte en la sección 2.5.1, no se ha requerido la implementación de una solución tecnológica de privacidad en la arquitectura de la solución. Esto se debe principalmente a:

- En general (y principalmente en los casos de validación planteados), la arquitectura no está diseñada para manejar datos personales ni sensibles de las corporaciones.
- Se trata de una herramienta para ser utilizada por la compañía/entidad adoptante, y generalmente limitada a un número reducido de usuarios finales, no por una gran multitud de clientes ni socios.
- No está pensada para ser expuesta al exterior, entendiéndose como exterior elementos no controlados por la propia entidad (los agentes serán desarrollados *ex profeso* con el fin de conectar los datos IoT con la solución).
- No se tratará, salvo casos particulares, de un sistema crítico para la empresa ni que maneje eventos que deban ser registrados en una tecnología de bloques (DLT). En caso de requerirse, la arquitectura ha sido diseñada con suficiente flexibilidad y escalabilidad como para poder ser incorporado como un nuevo microservicio.

Por tanto, en relación a la privacidad, lo que ha decidido incluirse ha sido una serie de normas y reglas a tener en cuenta durante (a) la instalación de la solución, (b) el desarrollo de agentes conectores de datos y (c) el desarrollo de servicios.

La siguiente tabla trata de resumir estas reglas recomendadas:

**Tabla 3.7:** Reglas recomendadas relacionadas con la privacidad

| Reglas sobre la privacidad en la fase de instalación de la solución:  |
|---|
| Creación de contraseñas para acceso inicial del usuario deberán ser encriptadas.  |
| La gestión de dominios, IPs y otras configuraciones deben ser realizadas a través de la funcionalidad “ <i>secrets</i> ” de Docker  |
| El acceso a los equipos de instalación debe realizarse a través del protocolo SSH ( <i>Secure Shell</i> ) accediendo mediante certificado.  |
| Reglas sobre privacidad a considerar durante el desarrollo de agentes   |
| Muestras de cada conjunto datos deben ser inspeccionadas para comprobar la posible inclusión de datos personales. Si contiene datos personales, esos fragmentos deben ser anonimizados y encriptados. Por ejemplo: localización de conductores de camiones, nombres de los conductores, edad... |
| Si contiene datos personales, cada uno de ellos debe estar etiquetado e identificado de forma unívoca (id), y debe establecerse una interfaz para permitir  |

siempre la eliminación de ese dato y que la información personal identificable (PII) no pueda ser rastreada. Por ejemplo: la densidad de trabajadores en un área en una terminal portuaria puede ser útil pero no debe poder permitir trazar quiénes son individualmente dichas personas.

“Principio de minimización de datos”. Sólo debe capturarse la información necesaria. Si, por ejemplo, la información personal sobre la compañía o el dueño de un sensor no es relevante para el servicio, debe descartarse.

**Reglas sobre privacidad a considerar durante el desarrollo de servicios**

Obligación de explicar el servicio que va a utilizar esos datos, incluidos los resultados esperados (por ejemplo, la inferencia)

No incluir ningún dato personal o sensible en las recomendaciones ni en el motor de reglas.

Almacenar la información de reglas y recomendaciones encriptada en la BBDD.

Si los datos contienen conjuntos de datos reutilizados o datos personales reutilizados, el conjunto de datos debe ser etiquetado y tratado de acuerdo con el resto de las normas.

**Recapitulación del módulo de seguridad y privacidad:**

El objetivo principal es dotar a la arquitectura de protección ante peticiones de acceso a recursos por parte de entidades externas a la solución. Según lo estudiado, los elementos a proteger están relacionados con el acceso del usuario final al módulo de visualización y con los accesos al almacenamiento de datos y al gestor de contexto por parte de los agentes conectores de datos. Para garantizar esta seguridad, se ha optado por la utilización de un sistema de gestión basado en políticas materializado por los componentes software de FIWARE KeyRock, Wilma y AuthzForce. Se ha definido una implementación ajustada a las necesidades de la arquitectura. Con respecto a la privacidad, se ha estudiado por qué no se requiere de un sistema tecnológico específico para cubrir estas preocupaciones. Se han propuesto una serie de reglas a cumplir durante las fases de instalación, desarrollo de agentes y desarrollo de servicios.

## 3.7. Solución integrada

Todos los módulos expuestos en esta sección han sido definidos con la perspectiva de integración entre ellos, orientados a crear una solución ligera y usable, compatible con las directrices de las arquitecturas IoT de nueva generación y preparada para ejecutar los servicios de cálculo y predicción de índices compuestos.

Tal y como se declaró en la sección 3.1.2, esta integración fue llevada a la práctica de forma iterativa en un proceso de generación, testeo y validación de productos intermedios (MVPs). Para ello, se utilizó un entorno de desarrollo contextualizado dentro de la infraestructura disponible del grupo de investigación SATRD, del cual el candidato forma parte.

Esta subsección refleja una visión conjunta de dichos módulos una vez elaborada su composición y funcionamiento internos.

Por un lado, se profundiza sobre la interacción entre ellos. Hasta ahora se ha descrito su funcionamiento aislado. No obstante, la mayoría de las funcionalidades de cara a ejecutar e interpretar los resultados del servicio requieren la intervención de varios de los módulos a la vez. En este sentido, esta sección propone una serie de explicaciones, acompañadas por diagramas de secuencia, de diferentes acciones dentro del sistema. Estos diagramas resumen las interacciones más importantes dentro de la arquitectura, que son las siguientes:

- Flujo completo de un dato útil para el servicio
- Envío de información desde agentes de dispositivos y de carga *batch*
- Consecución de la persistencia de los datos
- Instalación y ejecución de un servicio
- Visualización a través de la interfaz de la solución (incluido en el resto de puntos anteriores)
- Acciones relacionadas con la seguridad:
  - Introducción de credenciales y aplicaciones
  - Proceso general de petición de acceso a recursos

Por otro lado, se expone información sobre los requisitos tanto software como hardware que serían necesarios para desplegarla, en función de varios tipos de configuraciones planteadas (*edge computing...*).

En la siguiente página se muestra la estructura completa de módulos y componentes de la arquitectura propuesta para la solución tecnológica de esta tesis doctoral.

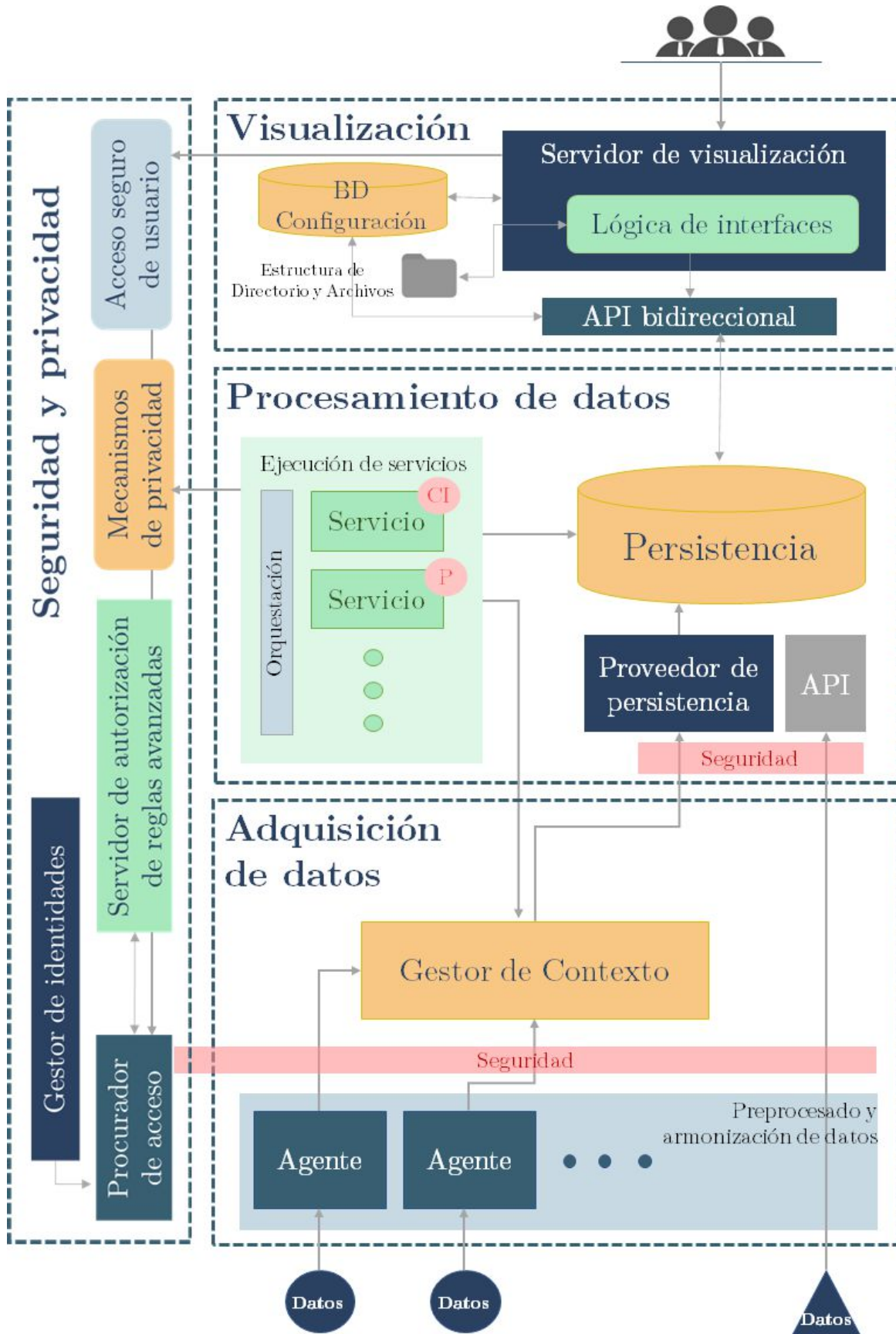


Figura 3.53: Esquema modular detallado de la arquitectura propuesta

El siguiente diagrama trata de ilustrar el ciclo completo de un dato relevante para el cálculo del indicador compuesto objetivo. El dato debe ser capturado e insertado en el sistema a través de un agente. Su ingreso como actualización de una entidad en el gestor de contexto es notificado al elemento de almacenamiento, que se encuentra suscrito a dicha entidad. Este dato ocupa un espacio determinado bajo un índice particular de la base de datos, que es consultada por el servicio. El resultado del procesamiento es registrado en otro índice (también predeterminado) para que pueda ser consultado por la lógica de visualización, que se encargará de presentarlo al usuario final.

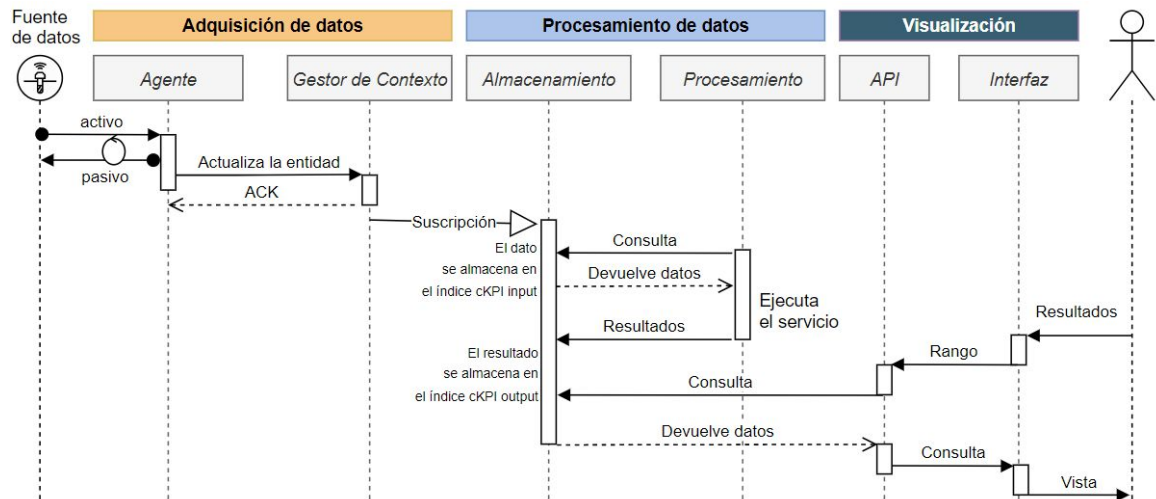


Figura 3.54: Diagrama de secuencia de flujo del dato

Este primer diagrama excluye deliberadamente las interacciones con el sistema de seguridad de la solución, centrándose en el flujo recorrido entre los módulos de procesamiento.

Los siguientes dos diagramas exponen las interacciones entre los módulos de adquisición de datos, seguridad y almacenamiento (secundariamente) para conseguir insertar datos provenientes de sensores (u otras fuentes de datos) dentro del sistema. En primer lugar, se expone cómo agentes conectores de datos “en tiempo real” (con periodicidad variable, pero con un influjo continuo) consiguen insertar información en el sistema a través del gestor de contexto. Es conveniente destacar que, si una entidad no existe la primera vez que se inserta, el software escogido (FIWARE ORION) la crea automáticamente. En el segundo diagrama se muestra una carga de información *batch* (directamente al almacenamiento). Durante la primera interacción (o cuando el *token* de seguridad ha caducado), la comunicación se inicia a través del elemento de aplicación de políticas (PEP proxy) para obtener un *token*, utilizándolo posteriormente en el resto de interacciones. Cabe destacar que, en el caso de agente pasivo, el propio agente NGSI deberá implementar sus mecanismos de seguridad suficientes para garantizar identidad y permiso, haciendo uso (si es necesario) del módulo PEP. Por ejemplo, exponiendo un API que puede ser protegida como una aplicación Wilma.



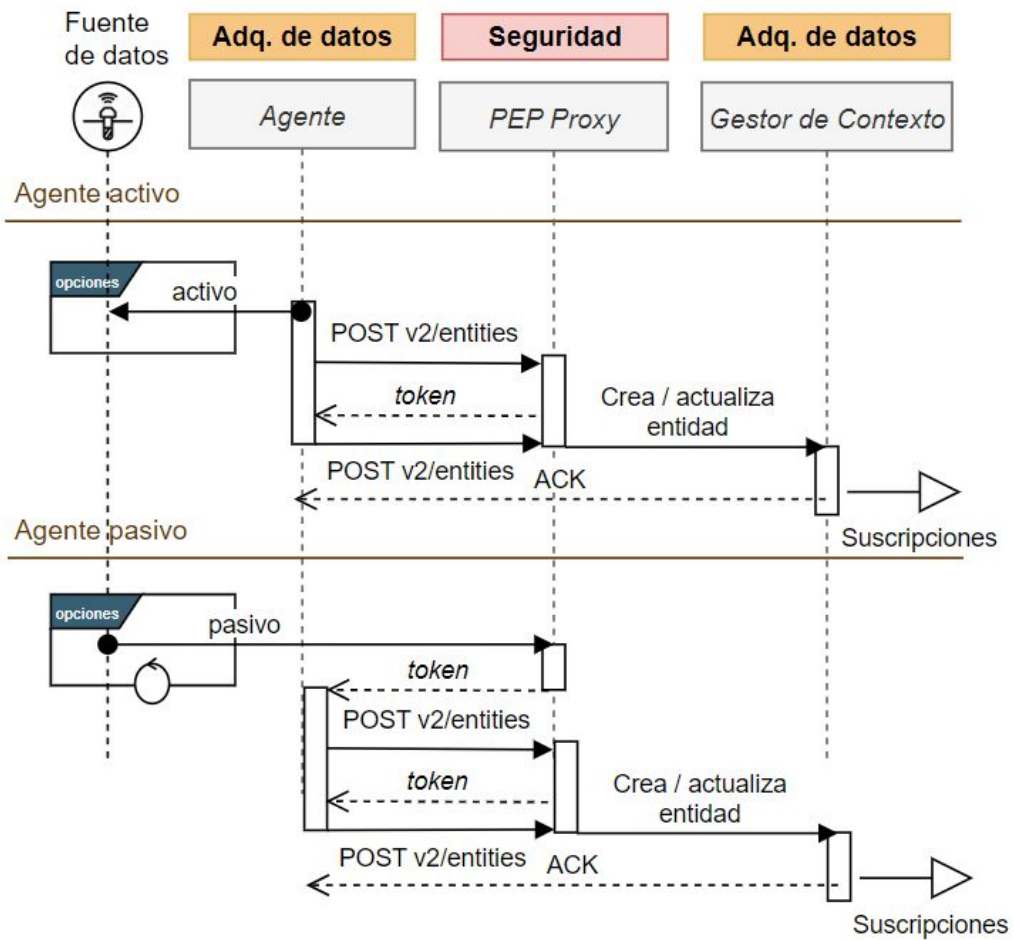


Figura 3.55: Diagrama de secuencia de datos desde gestor de contexto

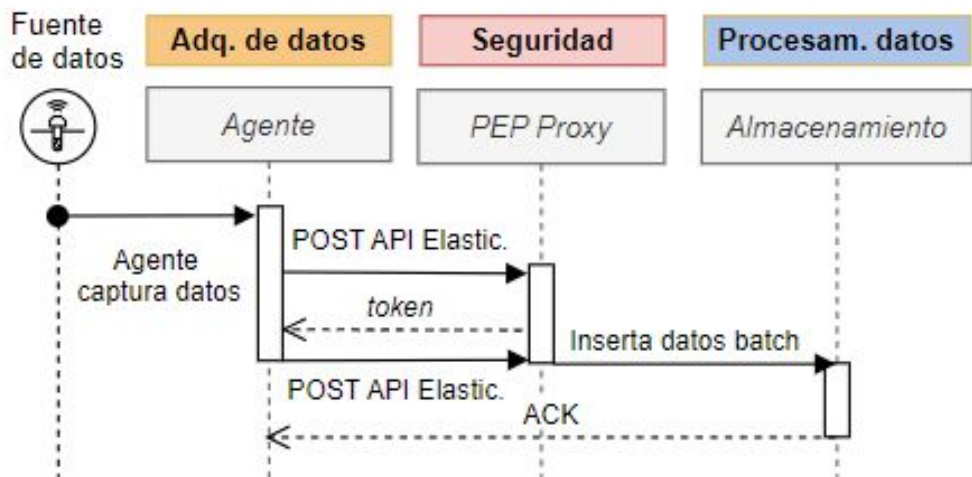


Figura 3.56: Diagrama de secuencia de carga de datos *batch* desde agente

En cuanto a la consecución de la persistencia de los datos, varias acciones son necesarias: (1) se ha de configurar la suscripción del proveedor de persistencia a las entidades correspondientes (todas), (2) análisis del tipo de despliegue (*on-premise*, con almacenamiento de datos centralizado o en entorno *edge*, etc.) y securización, si es necesario, con métodos particulares en cada caso.

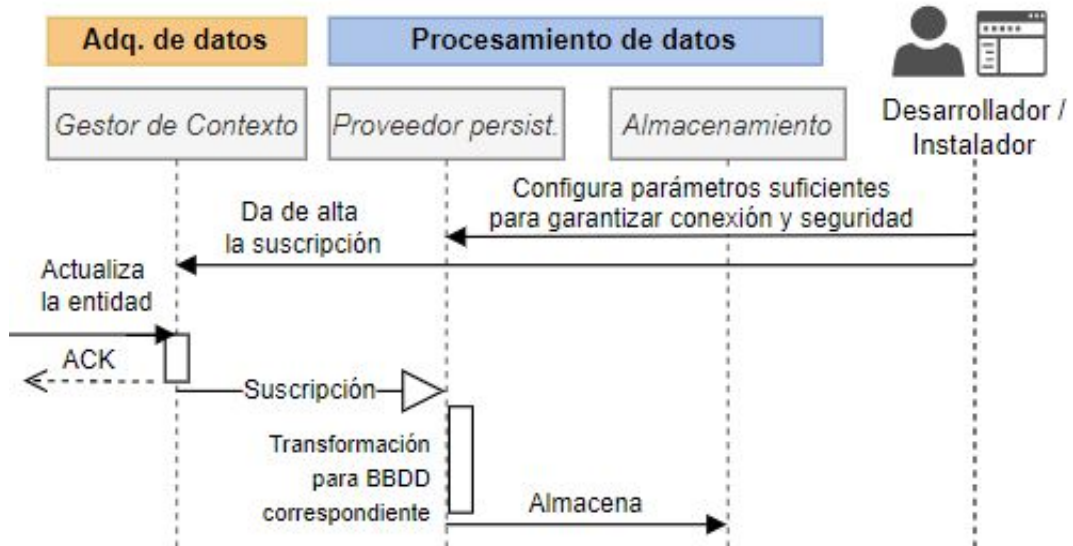


Figura 3.57: Diagrama de secuencia de la persistencia en el sistema

Una vez los datos se encuentran almacenados en los correspondientes índices (o colecciones), llega el momento de instalar y ejecutar servicios. En el siguiente diagrama se muestra el flujo de dichas acciones y qué módulos se ven envueltos:

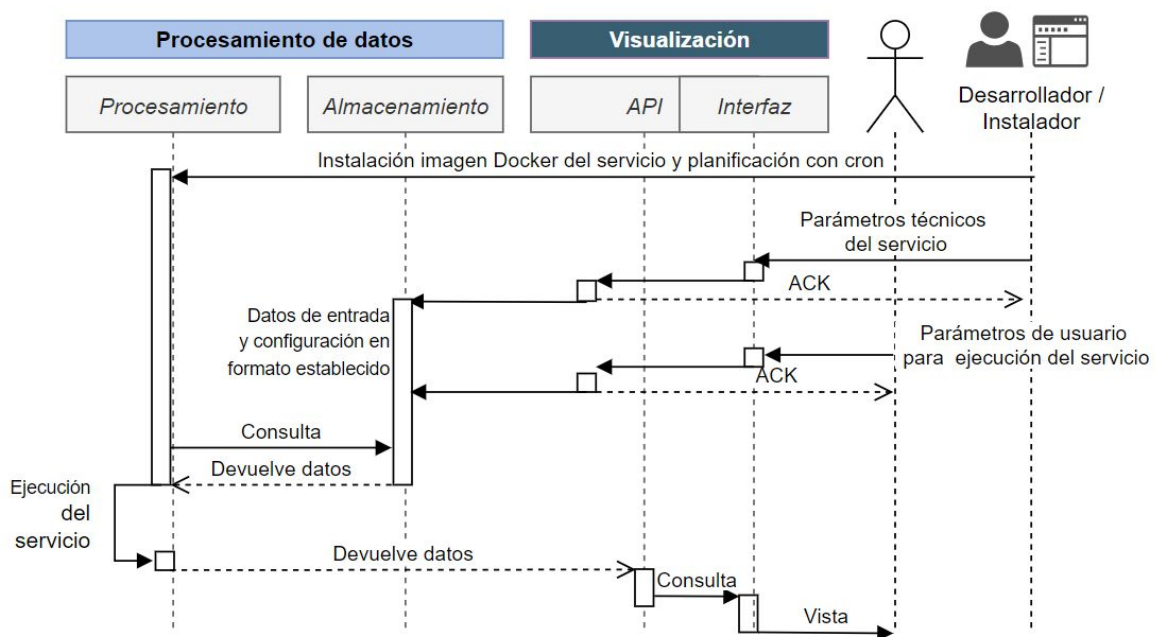


Figura 3.58: Diagrama de secuencia de instalación y ejecución de servicios



En cuanto a la seguridad (complementando las indicaciones de la Figura 3.52: Seguridad en acceso de usuario: roles y permisos), se proporcionan a continuación dos diagramas referidos a (i) la configuración y (ii) la interacción entre los elementos que la componen en el momento que se inicia una petición de acceso a un recurso.

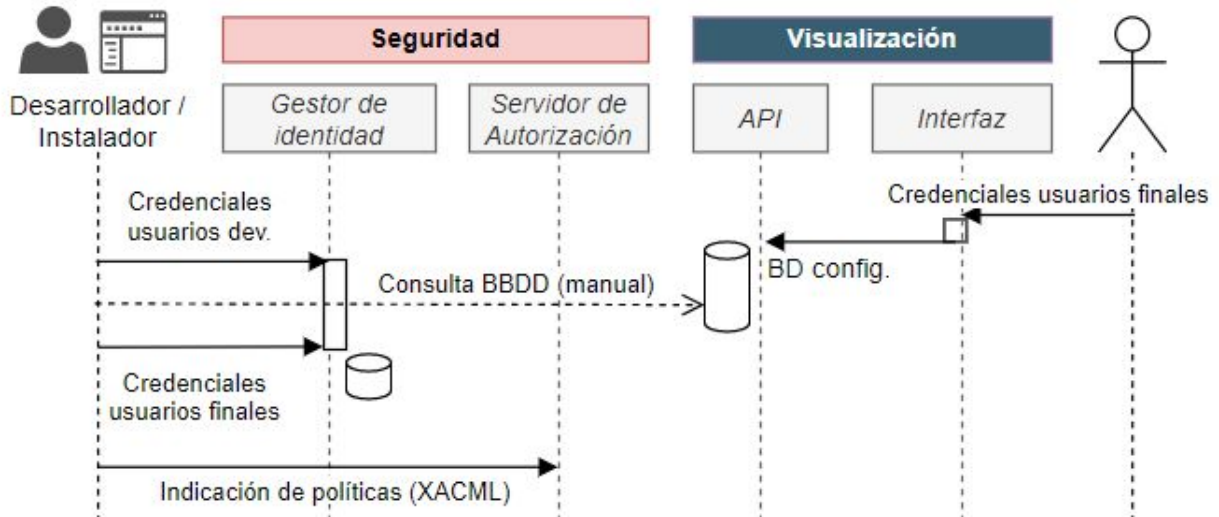


Figura 3.59: Diagrama de secuencia de la configuración de seguridad

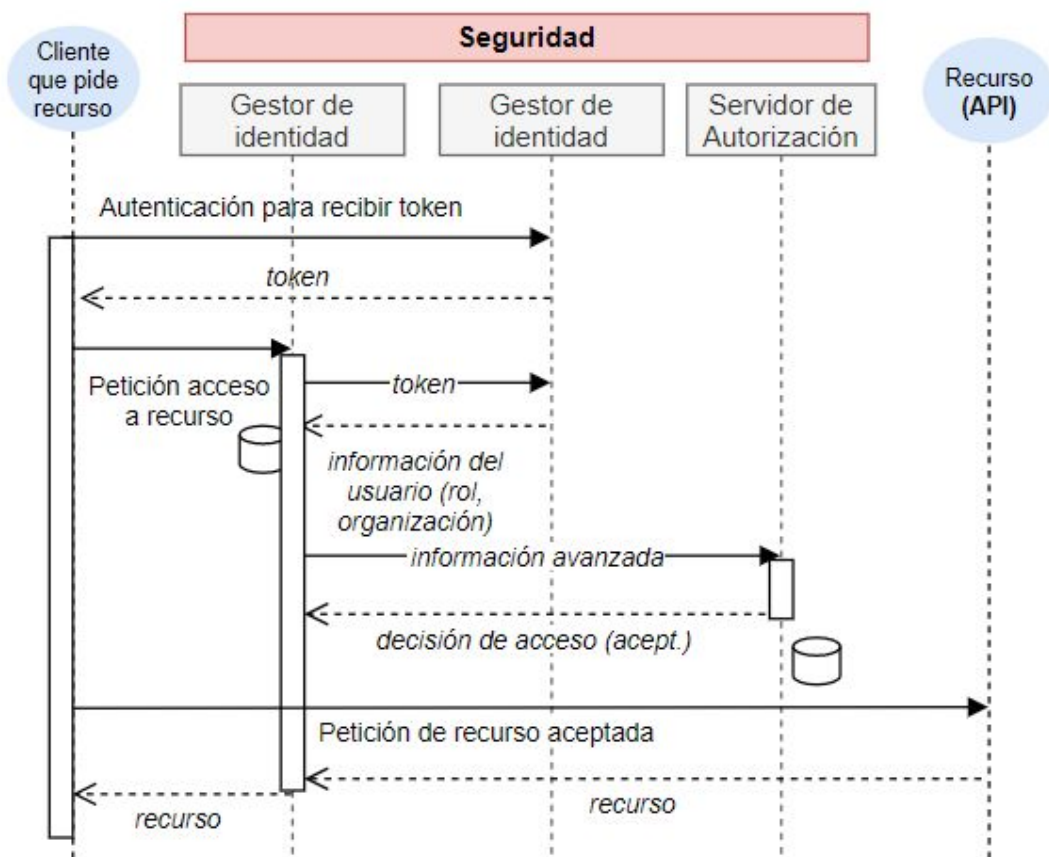


Figura 3.60: Diagrama de secuencia de la petición de acceso a recurso

Durante toda la sección se han hecho varias referencias a la flexibilidad de la arquitectura para desplegarse en entornos *edge computing*. Éstos se caracterizan por una baja disponibilidad de recursos tanto de computación como (generalmente) de red. A la hora de desplegar todos los componentes anteriores y su integración en el *edge computing*, se plantearon varios escenarios posibles.



Entorno central (cloud u on-premises) completo. El caso más habitual. Todos los módulos residirán en el mismo host (o en varios si existe orquestador de contenedores tipo Kubernetes) en un mismo entorno centralizado. El escenario más seguro y con menos complejidades para su despliegue.



Entorno central pero situando la captura y pre-procesamiento de los datos (agentes conectores) más cerca (lo más cerca posible) de la fuente de datos. Se trata del segundo caso más habitual, donde la seguridad se garantiza con las APIs y gestión de identidad.



Entorno más orientado a edge computing, moviendo tanto agentes conectores como gestor de contexto a entornos de computación de borde, permitiendo la deslocalización en diferentes puntos siendo capaces de mantener seguridad y privacidad de muchas fuentes a la vez en cada uno de ellos. Tercer caso de uso en frecuencia.



Un paso más hacia la computación ubicua. En este caso, cada elemento del edge computing poseería una instancia de almacenamiento de datos a largo plazo (persistencia). Escenario ideal para potenciar el procesamiento en el edge, por ejemplo a través de servicios que incluyan aprendizaje federado.



Descentralización total. Una instancia de la solución completa en un entorno edge. Escenario que se ha validado en la tesis doctoral (μPEI). La seguridad descentralizada aplica en este contexto. Compartición de índices, privacidad, nuevas tecnologías y, en definitiva, planteamientos NGIoT alcanzan su mayor grado en este tipo de despliegue.

Figura 3.61: Exposición de diferentes escenarios de despliegue

La siguiente tabla trata de reflejar los requisitos software y hardware mínimos recomendados para desplegar la arquitectura propuesta. A lo largo de las filas, se indica una serie de consideraciones sobre el despliegue *edge* y sobre las particularidades de los servicios.

**Tabla 3.8:** SW y HW recomendado para el despliegue de la arquitectura

| Módulo                 | Componente              | Software seleccionado en la tesis                             | Hardware mínimo recomendado                |
|------------------------|-------------------------|---|--|
| Adquisición de datos   | Agentes                 | <i>pyngsi</i>   | Depende de cada agente. Realmente ligeros. |
|                        | Gestor de Contexto      | FIWARE ORION y MongoDB  | 4GB RAM, 2 CPU cores                       |
| Procesamiento de datos | Persistencia            | Elasticsearch (pila ELK)<br><i>Nota:</i> MongoDB <i>edge:</i> | 16GB RAM, 4 cores<br>1GB RAM               |
|                        | API                     | Node.js   | <1GB RAM                                   |
|                        | Servicios               | Java, Python...   | Depende de cada servicio.                  |
| Visualización          | Servidor                | Apache  | 512MB RAM                                  |
|                        | Base de datos           | MongoDB   | 1GB RAM                                    |
| Seguridad              | PAP, PDP, PEP, PIP, PRP | KeyRock, Wilma, AuthZForce                                    | 8GB RAM,                                   |

Adicionalmente, el *host* seleccionado debe tener instalado Docker y OpenSSH para permitir la ejecución de contenedores y el acceso mediante protocolo SSH. Cabe destacar que, en general, se asume que los equipos de computación a utilizar para desplegar la arquitectura se basan en sistemas operativos Linux [375] y poseen una arquitectura x64 [376]. Aunque la mayoría de los componentes han sido validados en escenarios donde se hace uso de arquitecturas tipo ARM [377], no puede garantizarse un funcionamiento completo de la solución sobre este tipo de máquinas.



# Capítulo 4

## Cálculo y predicción de índices compuestos

### 4.1.Estado de la técnica

Uno de los objetivos principales de la tesis doctoral es profundizar en la elaboración de índices compuestos<sup>1</sup> y plantear una herramienta automatizada (orientada a su uso sobre una arquitectura IoT) para calcularlos y predecirlos.

No obstante, previamente a relatar dicha herramienta, es necesario realizar una introducción a los indicadores compuestos, así como indagar en los planteamientos existentes. Es por ello que se ha introducido esta subsección.

En los subpuntos se expone la siguiente información:

---

<sup>1</sup> Durante toda la tesis doctoral se utiliza tanto el término “indicador compuesto” como “índice compuesto”. Deben entenderse como equivalentes, utilizando uno u otro indiferentemente.

- En el punto 4.1.1 se realiza una introducción a los índices compuestos, en qué consisten, cuáles son sus orígenes y principales usos, así como se hace un repaso del estado del arte de algunas implementaciones relevantes para el caso de la tesis doctoral.
- En el punto 4.1.2 se profundiza en la metodología de construcción de índices compuestos. Acorde a las principales referencias estudiadas, la generación de estos índices se compone de 10 (diez) pasos. Conviene destacar aquí que la solución software planteada en la tesis cubre explícitamente el cálculo de dichos índices, dejando aparte algunos aspectos estadísticos metodológicos para su ideación y validación.
- Por último, el punto 4.1.3 expone (de forma cronológica) la aparición de herramientas software (similares a la propuesta en la tesis) orientadas a la computación de índices compuestos. Se culmina con una comparación con la herramienta desarrollada en este trabajo.

### 4.1.1. Introducción

Un indicador o índice compuesto (de ahora en adelante **CI**, del inglés *Composite Indicator/Index*) es un valor numérico que condensa la información de un fenómeno real complejo a partir de una combinación matemática de indicadores individuales que representan diferentes dimensiones de un concepto. Se entiende como indicador individual un elemento informacional adimensional asociado a un aspecto de la realidad. Según traducción literal de una nota de la Comisión Europea en 2002: “*los indicadores compuestos se basan en subindicadores que no tienen una unidad de medida común significativa y no hay una forma obvia de ponderar estos subindicadores*” [378]

A veces referidos como indicadores LCS (*Lead, Composite, Sentiment*), diferenciando su objetivo (económico, socioeconómico o emocional, respectivamente), en este trabajo se utiliza el término indicador compuesto indiferentemente, ya que la arquitectura y solución desarrolladas aplicarían en todos los casos.

Los indicadores compuestos tienen como objetivo proporcionar una visión clara y fácil de interpretar (y lo más ajustada posible) sobre un aspecto de la realidad. Este tipo de herramientas son consideradas útiles para ayudar en la elaboración de políticas o en la toma de decisiones, aportando un valor único sobre el que se puede actuar, relajando la cantidad de recursos y tiempo que requeriría analizar grandes cantidades de datos estadísticos [379] Debido a su naturaleza estandarizada (deben utilizarse referencias y métodos comunes respaldados por la ciencia de datos) los indicadores compuestos pueden sintetizar e integrar con éxito grandes volúmenes de información, contribuyendo a atraer el interés público al proporcionar una cifra resumida con la que comparar los resultados de distintas entidades y su progreso a lo largo del tiempo [380] En resumen, algunas de las ventajas de utilizar un CI son la facilidad de interpretación, el aporte de una base

para la comparación (ranking) entre entidades asimilables, la idoneidad para reconocer la evolución a lo largo del tiempo de un factor complejo, la provisión de una visión global y su gran capacidad de *benchmarking*.

El hecho de incluir esta técnica de índices compuestos, potencialmente utilizable como elemento comparador/auditor por parte de entidades públicas, se ha antojado clave a la hora de plantear un posible modelo de negocio de la solución (sección 6).

Sin embargo, si un indicador compuesto no está bien construido y no se interpreta correctamente, puede enviar un mensaje engañoso y débil [381]. Puede implicar una pérdida de información (por reducción), desviando la atención de las dimensiones relevantes. Además, requiere de altos niveles de cantidad y calidad de los datos para poblar todos los subindicadores ([382][383]). No obstante, si se aborda y analiza adecuadamente, puede ser una piedra angular tanto para la elaboración de políticas, auditorías o licitaciones desde el punto de vista público como para un seguimiento útil de la evolución en algún aspecto de la realidad para las propias empresas, según el candidato.

Estas ventajas y desventajas han sido audazmente analizadas en la literatura. Puede verse un resumen en la Tabla 4.1.

**Tabla 4.1:** Ventajas y desventajas de los índices compuestos. Fuente: [384]

| Ventajas   | Desventajas   |
|--|---|
| Concentra información compleja, correspondiente a varias dimensiones, en un solo número  | Puede perderse información en la reducción, llevando a una mala interpretación de la realidad   |
| Más fácil y directa interpretación en comparación con muchas líneas de información por separado  | Si no se construye adecuadamente puede ofrecer conclusiones simplistas  |
| Al responder a un proceso estandarizado con referencias comunes, puede compararse la información de muchas entidades que cumplan dicha metodología | La selección de indicadores debe involucrar a suficientes agentes para asegurar que toda la información relevante está siendo contemplada |
| Permite un control de la evolución temporal de un fenómeno   | Puede conducir a asociaciones causa-efecto erróneas   |
| Permite reducir el análisis a una menor cantidad de indicadores  | La calidad y cantidad de datos requeridos de dichos indicadores es alta   |
| Puede ser fácilmente comunicado y utilizado como referencia para licitaciones, márgenes, multas, etc.  | En caso de existir un error, éste se propaga con facilidad.   |

Según representantes de la Comisión Europea, estas desventajas pueden ser mitigadas siguiendo los tres pilares para la construcción de índices compuestos [380]:

- 1) Un marco conceptual sólido.
- 2) Buena calidad de los datos.
- 3) Una metodología clara y transparente.

Si éstos se siguen, un CI puede ser considerado válido para su utilización en la elaboración de políticas y toma de decisiones desde el ámbito público. De entre los anteriores, especialmente relevantes para esta tesis doctoral han sido los pilares 2) y 3). Con respecto al segundo, la calidad global del CI se relaciona tanto con los datos de entrada (indicadores iniciales) utilizados para construir el CI como con la calidad de los procedimientos utilizados para hacerlo. Según el manual de construcción de índices compuestos elaborado por la Comisión Europea en 2008 [385], los requisitos mínimos de dichos datos son: a) que sean relevantes, b) que sean precisos, c) que presenten una aceptable frecuencia de adquisición, d) que sean accesibles, e) que sean interpretables y f) que sean coherentes. Es responsabilidad de los diseñadores de cada despliegue (véase sección 5) asegurarse que estos requisitos se cumplen para poder obtener un valor de CI válido.

En relación al tercer punto, se ha elaborado una sección específica en ese manuscrito (sección 5.2) que, acompañada de las instrucciones de construcción de CIs de este capítulo (4.1.2) evidencian la claridad y transparencia propuesta para la solución de esta tesis doctoral; alineada con dicho pilar.

Volviendo a la revisión de los índices compuestos como herramienta de uso en la literatura, en los últimos diez años el reconocimiento de los indicadores compuestos tanto para la elaboración de políticas como para la comunicación está en auge. Se ha evidenciado que es mucho más fácil entenderlos que analizar muchos indicadores al mismo tiempo [380].

En cuanto a su aplicación, el uso de los CI para definir y analizar fenómenos sociales, económicos, políticos o medioambientales complejos ha sido tradicionalmente competencia casi exclusiva de las oficinas de estadística [386], especialmente de aquellas dedicadas a comparar la actuación/rendimiento entre países (por ejemplo, Producto Interior Bruto – PIB) [387]. Se documentaron al inicio de la década de 2010 investigaciones para evaluar macro áreas específicas como medio ambiente, sostenibilidad o energía [388]. Sin embargo, existe una tendencia a utilizar cada vez más la metodología de índices compuestos. Se han encontrado algunos ejemplos centrados en: el tipo de uso del suelo y otros servicios en ecosistemas y contextos urbanos [389], en la sostenibilidad medioambiental (por ejemplo, el ESI (*Environmental Sustainability Index*) del *World Economic Forum* - WEF [390] o el mismo PEI del proyecto PIXEL [391] (participado por el doctorando en los últimos años). También se han encontrado ejemplos relacionados con la economía urbana (por ejemplo, el IMI (*Internal Market Index*) de la DGMARKT [392] o el Indicador de Sentimiento Económico ESI (*Economic Sentiment Indicator*) de la CE [393]), en los aspectos sociales (el famoso HDI -*Human Development Index*- de las Naciones Unidas [394]) y en los avances de la ciencia, la tecnología y la información (por ejemplo, el TAI -*Technology Achievement Index*, también de las



Naciones Unidas [395] o el índice ICT de Fagerberg [396]). Uno de los más recientes es el NRI, que ha tratado de crear un índice para comparar la resiliencia de 130 economías del mundo frente a los desafíos planteados por la COVID-19 [397]

La Comisión Europea ha publicado recientemente su herramienta *Scoreboard Explorer*, que trata de recopilar todos los índices compuestos disponibles en el contexto mundial. Actualmente cuenta con 117 índices registrados, abarcando más de 375.000 datos iniciales e incluyendo información de 239 países [398].

A continuación se relatan algunos ejemplos específicos encontrados en el estado del arte, relacionados con la perspectiva de la tesis doctoral y que profundizan en cómo se ha calculado el indicador compuesto. En primer lugar, el más antiguo de éstos es el *Living Planet Index* [399]. Este caso resulta interesante ya que se asigna una puntuación a los datos en función de su fuente, metodología y de si se ha incluido una medida de variación. Esta puntuación ha servido de referencia para la generación de los pesos y del índice de fiabilidad en el enfoque general del servicio de la tesis doctoral (4.2).

Resultó también interesante el índice *Smart Campus*, planteado en 2018 para la Universidad de Sapienza (Roma) [388], que diseñó un índice para comparar la idoneidad de campus universitarios dividiendo en subindicadores de medioambiente, vida y servicios y energía. Este caso fue de interés ya que una de las potenciales aplicaciones de la tecnología de la tesis podría ser partir de esta creación conceptual y desplegarlo en el campus de la UPV u otras universidades. Técnicamente, los métodos matemáticos elegidos (véase sección 4.1.2) resultan factibles para un caso general.

Un caso práctico se presenta en el trabajo *MethOSM* [386] (publicado en 2019), que propone el uso de indicadores compuestos para la valoración de propiedades inmobiliarias en Italia haciendo uso de datos georreferenciados disponibles en abierto. Se trata de uno de los trabajos más parecido (en espíritu) al de esta tesis doctoral, ya que además de analizar la metodología de construcción del índice compuesto, plantea un sistema automático de extracción de los datos desde la fuente, este caso *OpenStreetMap* (OSM) [400]. No obstante, no se ajusta a los pasos de la metodología de construcción y cálculo de índices compuestos estudiada en la tesis (sección 4.1.2).

En diciembre de 2020 se propuso la utilización de esta técnica para una gestión cognitiva de las redes de comunicaciones móviles [401]. El caso de uso se centra en el análisis de la equidad de una red 3G/4G en diferentes áreas de cobertura como métrica fundamental en la operación y gestión de las redes. Se realizó la aplicación de índices compuestos mediante software STATA sobre datos generados sintéticamente siguiendo las mediciones que pueden darse en ese tipo de redes.

Por último, afinando la investigación en busca de CIs validadas en entornos similares a los escenarios planteados en esta tesis doctoral (ciudad-puerto), el trabajo más relevante revelado en el estado del arte es [402]. En el mismo se propone un Índice de Puerto Inteligente (*Smart Port Index*) que permita a los puertos

mejorar su resiliencia y sostenibilidad. Sin embargo, en este desarrollo no se incluyó la perspectiva conjunta de la ciudad. Además, se formuló para la comparación entre puertos y los valores de los KPIs se extrajeron de referencias bibliográficas y no se apoyaron en un sistema basado en el IoT, además de no seguir una metodología estandarizada de construcción de índices compuestos.

Como puede observarse, aunque la creación de índices compuestos es un concepto conocido y usado desde hace más de dos décadas [403], se trata de una técnica de gran actualidad y en continua evolución. Esta afirmación se ve refrendada, por ejemplo, con la utilización de índices compuestos desde 2020 por parte de la Comisión Europea para la discusión sobre la estrategia de sostenibilidad conjunta Asia-Europa (*Asia-Europe Meeting – ASEM*) [404]. Además, es interesante mencionar que la mayoría de las citas y ejemplos mencionados, tanto en esta subsección como en la de tecnologías disponibles (sección 4.1.3) datan de 2019 en adelante. Estos plazos coinciden exactamente con el tiempo de desarrollo de esta tesis doctoral por parte del candidato (2019-2022), lo cual hace pensar que es un campo en auge y que requiere del desarrollo de trabajos como este en este momento.

Finalmente, es conveniente mencionar que el principal grueso de referencias y trabajos relacionados con índices compuestos ha orbitado alrededor del grupo de la Comisión Europea denominado *K4P-Knowledge4Policy*, perteneciente a la rama *JRC (Joint Research Centre)* [405]. Desde el año 2004 se apuesta desde este grupo por los indicadores compuestos como una herramienta útil para la toma de decisiones a nivel público. Según traducción literal de las palabras de su *Head of Unit* (Michaela Saisana): “*Conectar los puntos es lo que necesitamos para tener la información correcta en el momento adecuado, ser críticos al respecto y tomar decisiones importantes con sabiduría*”. Esta afirmación fue pronunciada durante la semana de indicadores compuestos y marcadores para el apoyo a la toma de decisiones con evidencia científica [406], organizada por dicha unidad en diciembre de 2021. Estas jornadas son organizadas todos los años desde 2015 y han resultado una fuente de inspiración y aprendizaje continuas para el candidato.

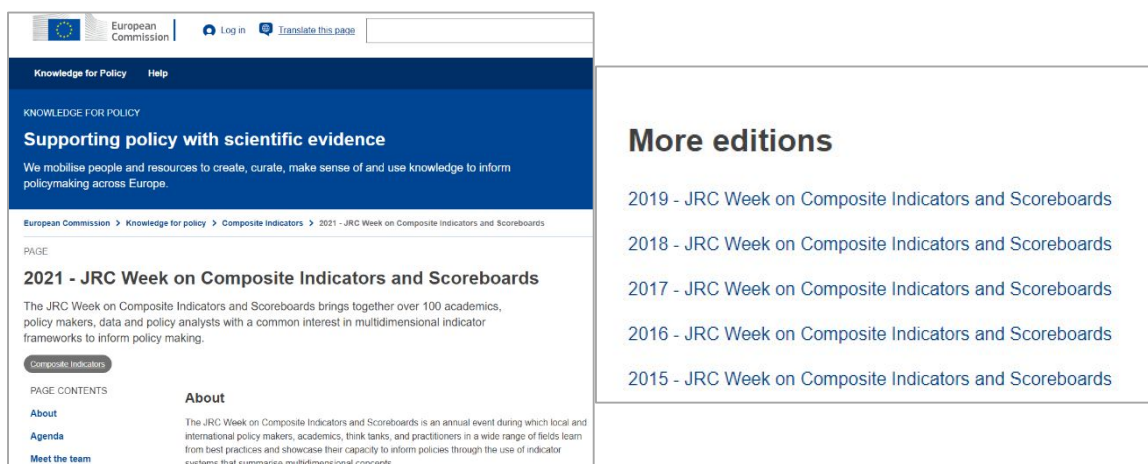


Figura 4.1: Semana de indicadores compuestos 2021 – JRC K4P

## 4.1.2. Construcción de índices compuestos

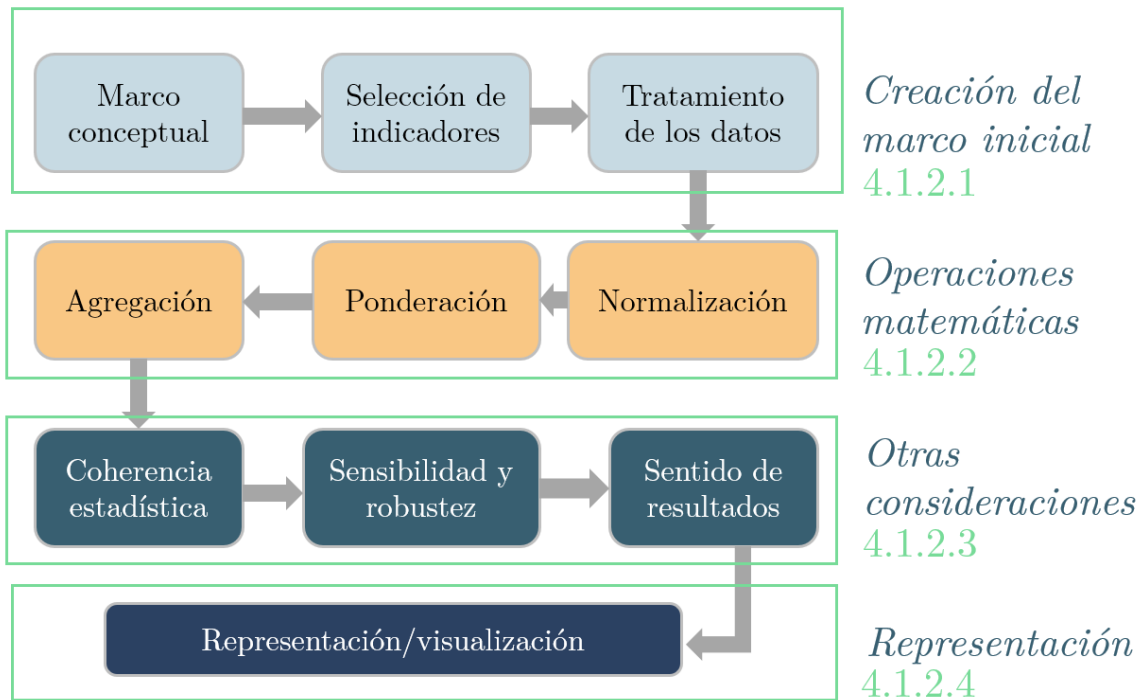
Existen en literatura estudios que describen los fundamentos matemáticos para construir un indicador compuesto, con sus ventajas e inconvenientes.

El primer manual completo exponiendo el marco teórico y las herramientas (matemáticas) para su construcción fue publicado en 2005 por el mencionado grupo de la Comisión Europea [407]. Más tarde, en 2008, el mismo grupo publicó un manual más elaborado, incluyendo un repaso en profundidad de todas las técnicas posibles en cada paso del proceso de construcción de índices compuestos [408]. El candidato considera que esta referencia es, todavía hoy, la más completa. Unos años después, en 2012, la Organización para la Cooperación y el Desarrollo Económicos (OCDE) adoptó esta metodología, produciendo un manual específico para la realización de índices compuestos relacionados con el ámbito económico, principalmente para ejercer una comparación entre países [409]. Las actualizaciones relevantes más recientes son por parte de las Naciones Unidas en 2019 [410] y el manual simplificado del grupo de la Comisión Europea, en 2021 [411]. Las referencias más secundarias son variadas y la publicación de nuevos enfoques metodológicos para la construcción de estos índices está continuamente en marcha.

De hecho, si algo ha permanecido constante es el debate sobre cuál debe ser dicha metodología para la formulación matemática de los índices. Desde los inicios de esta técnica [412] se documentaron voces discordantes: por una parte, la agregación de diferentes indicadores en un único índice era considerado como relevante para capturar la información mientras que, por otra parte, existía la visión de que debería pararse la agregación en los subindicadores, ya que los métodos de ponderación y agregación son (en gran medida) arbitrarios y subjetivos y se estaría introduciendo sesgos innecesarios. Esto se debe, en parte, a la perspectiva desde el punto de vista puramente estadístico de que en dichas metodologías de construcción de indicadores se pone demasiado énfasis, recursos y tiempo para obtener un número con una validez matemáticamente cuestionable. No obstante, la tendencia de uso de estos indicadores parece irreversible (dadas las evidencias expuestas), por tanto, se debe encontrar la mejor forma de obtenerlos. Esta controversia se ha mantenido durante el tiempo, a pesar de la creación de renovados manuales [413]. El acalorado debate en la comunidad científica sobre este aspecto se encuentra activo a fecha de hoy, decretando que, por el momento, no existe una metodología única válida; ésta depende de las áreas de aplicación y de los objetivos estratégicos de cada caso [414].

Para el trabajo de esta tesis doctoral se ha decidido crecer sobre la metodología más establecida, la publicada por el Centro de Competencia mencionado aportando aquellos aspectos técnico-prácticos más aptos para entornos IoT. En particular, el tratamiento y procesamiento de datos, así como la elección de indicadores y la visualización de resultados ha sido evolucionado por el doctorando.

Esta metodología recomienda seguir un proceso iterativo de 10 (diez) pasos, que se ilustran en la siguiente figura:



**Figura 4.2:** Pasos para la construcción de un índice compuesto

Se ha decidido estructurar (en aras de mayor claridad en el documento) los pasos en cuatro bloques diferenciados.

- **Creación del marco inicial:** Incluye los pasos de definición del indicador final, de los subindicadores y de los primeros tratamientos sobre los datos como su análisis o la imputación de valores ausentes.
- **Operaciones matemáticas:** Compone el grueso de las computaciones que realiza el servicio propuesto en la tesis: normalización, ponderación y agregación.
- **Otras consideraciones:** Se trata principalmente de actividades *offline* para la validación del índice tales como análisis de sensibilidad y robustez. Es uno de los pasos que se han cubierto de forma más superficial por la herramienta software planteada en el trabajo.
- **Representación:** Trata sobre la visualización de la información de forma gráfica y atractiva para causar el efecto y la interpretación buscada.

En las siguientes subsecciones se realizará un repaso más detallado a cada uno de los bloques indicados, describiendo los métodos existentes y contextualizando cada paso desde el punto de vista del servicio de esta tesis doctoral.

### 4.1.2.1. Creación del marco inicial

El primer paso en la construcción de un índice compuesto es la **definición del mismo**. Se debe analizar el fenómeno real que quiere representarse a través del indicador. Para ello, se recomienda (1) realizar una revisión de la literatura del sector para encontrar referencias previas, (2) mapear estas iniciativas anteriores con los objetivos que se persigue en el caso particular, (3) diseñar el indicador siempre teniendo en cuenta el campo de aplicación donde va a desplegarse – dividiendo el fenómeno multidimensional a medir y entendiendo qué requiere en términos de entrada y salida y (4) contando con agentes involucrados en dichas áreas (transporte, educación, ciudadanía, economía...) [380]. De este modo, el índice compuesto nacerá con la validación temprana por parte de expertos, que confiarán más en su posible utilización, llevando a una mayor probabilidad de éxito de la solución que implemente dicho indicador (véase sección 6).

En este sentido, se ha de tener en cuenta que un fenómeno real es en la mayor parte de los casos un sistema complejo, por tanto, esta elección requiere suficiente reflexión, considerando qué se desea medir, con qué propósito va a ser usado y empezando a atisbar las potencialidades de medición del mismo [410]. Por ello, es clave la transparencia en este proceso y la clara identificación de todos los componentes involucrados.

Una vez acordado el indicador compuesto, el siguiente paso para la creación de un marco inicial es la **selección de indicadores que lo compondrán**. En este punto se debe realizar una distinción: mientras que para el CI global el foco ha sido un marco teórico y de actores involucrados, para escoger los indicadores se debe considerar (y analizar en la medida de lo posible) de qué datos se va a disponer y cuáles se van a escoger para construir el indicador.

La elección de indicadores es crucial para el éxito del índice compuesto en términos de validez, interpretabilidad y niveles de adopción. Se ha evidenciado que la inclusión de indicadores que no se relacionan adecuadamente con el estado natural del fenómeno objetivo llevan a la confusión en los resultados [415]. Esta identificación no es fácil, ya que (a) requiere un gran conocimiento del problema o fenómeno a representar, (b) no siempre existen datos apropiados, actualizados y/o con la precisión necesaria como para ser incluidos como indicadores en la computación del índice. De hecho, en muchas ocasiones la baja disponibilidad de datos, o el coste del acceso a los mismos, por su escasez o por trabas legales/administrativas, lleva a una toma de decisiones errónea en la selección de indicadores [386].

En la próxima figura se muestra un diagrama que puede ser utilizado para distinguir la validez de un elemento para convertirse en indicador. Tras una revisión de la literatura, preselección de candidatos y análisis de los datos disponibles para calcularlos, debería tenerse en cuenta el flujo de decisión propuesto en dicha figura. De nuevo, registrar estos procesos se antoja clave para demostrar la transparencia en la construcción del índice [416].

El flujo indicado en la figura es una mejora del candidato de la propuesta presentada en [416] (presentada en 2021) a partir de los conocimientos adquiridos durante la realización de la tesis doctoral.

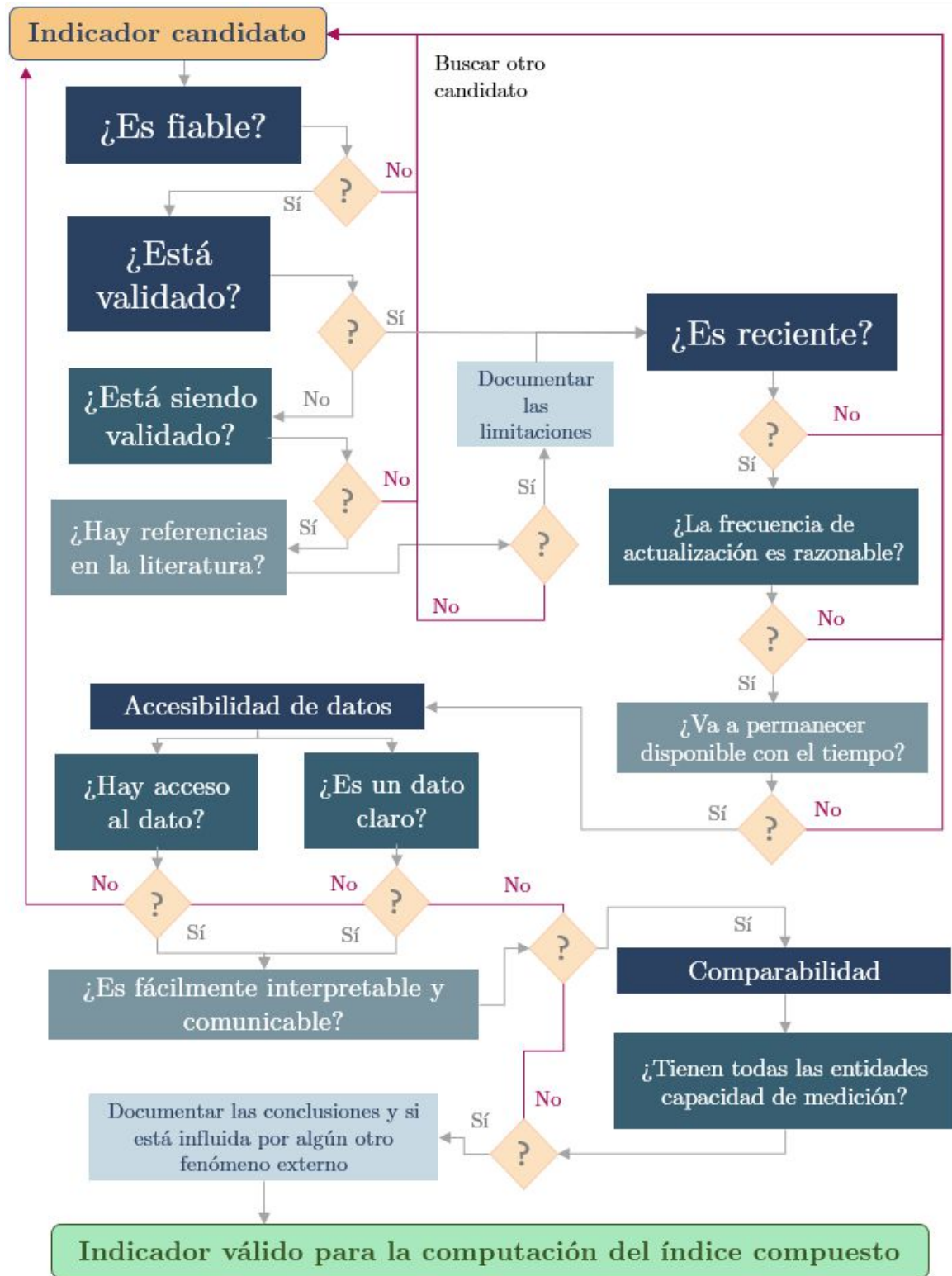


Figura 4.3: Flujo de decisión para la inclusión de indicadores

Otros dos aspectos relevantes en este punto, de alguna forma embebidos dentro del flujo presentado, son la frecuencia de cálculo de los indicadores y su estructuración en directos o indirectos.

En cuanto a la frecuencia de cálculo, debe considerarse también cuál es la periodicidad con la que va a querer ofrecerse el valor resultado del cálculo del CI. Debe entenderse que un indicador compuesto es, por definición, un valor retrospectivo. Lo que significa que, para un momento específico del tiempo, el resultado del CI se calcula para un periodo de tiempo anterior. En este sentido, se debe decidir cuál es dicho periodo. Se debe estudiar qué acciones que pueden tomarse a partir de dicho resultado y la necesidad real de alimentar la computación con datos con una alta tasa de actualización. Así, una vez definida dicha frecuencia, si alguno de los indicadores no va a poder cumplir con la tasa de refresco pactada debe descartarse o aplicar métodos de valores ausentes (véase en las siguientes páginas), siendo conscientes de la disminución de precisión estadística que esto podría suponer. Si este es el escenario que se da, es probable que algunos de los indicadores presenten mayor número de entradas en el periodo elegido, ya que será el elemento más restrictivo aquel con menor frecuencia de muestreo. Asumiendo esta afirmación, la solución planteada en esta tesis propone la posibilidad de establecer resultados en periodos más cortos incluyendo únicamente aquella información más reciente de los indicadores más “frecuentes”.

Con respecto a las visiones aceptadas en la literatura acerca del periodo de computación de un indicador, la OCDE (hay que contextualizar que esta entidad se centra en el análisis de datos globales de índole económica por países) recomienda que la frecuencia sea trimestral [409]. Especifica, además, que los datos deberían tener una tasa de actualización más frecuente que dicha periodicidad, y que la preferencia siempre ha de ser las series temporales largas y sin parones.

Uno de los pasos en el flujo anterior se refiere a la validación y al acceso directo a la información. Es aceptado (y frecuente, según analizado por el candidato en los escenarios de validación – sección 5) la introducción de datos de manera directa (por ejemplo, medición con sensores, información oficial proporcionada por la entidad...) o a través de indicadores indirectos, que requieren la obtención de datos auxiliares “externos” (de la literatura, de informes proporcionados por entidades públicas. Se prefieren, por supuesto, los indicadores directos, no obstante es preferible la inclusión de elementos indirectos (mientras respondan satisfactoriamente las preguntas del flujo anterior) siempre que sean necesarios para ejercer la computación del índice.

Además, en cuanto a la composición estructural del índice (los niveles) se debe realizar un ejercicio similar al de la figura anterior al agrupar los indicadores en conjuntos más reducidos. El objetivo es disgregar en varias “fases” los indicadores iniciales en varias subcategorías, de tal manera que se obtenga un índice compuesto final único. Esto será más habitual cuantos más indicadores iniciales se seleccionen. Dentro de esta tesis doctoral, a dicha estructura se le ha denominado “árbol”, ya que recuerda a dicha descomposición jerárquica en ramas.

Una vez completado dicho flujo, los datos deben recogerse y aplicar sobre ellos métodos estadísticos apropiados para su computación hacia un índice compuesto [385]. Idealmente, los tratamientos expuestos en las siguientes páginas deben partir de la base del análisis del *set* de datos completo. Para el caso de esta tesis doctoral, esto no es posible ya que se ha diseñado una herramienta que procese y calcule indicadores compuestos “en tiempo real”. Es decir, los datos no van a ser fijos/estáticos completos desde el inicio, sobre los que se pueda ejercer operaciones, sino que irá alimentándose de forma dinámica de las series temporales provistas por sensores u otras fuentes del Internet de las Cosas.

Una vez realizada una inspección global de los datos, deberían identificarse los **valores atípicos**. Estos valores son aquellos que se desvían notablemente o se distinguen del resto. En las ocasiones en las que se dispone de datos retrospectivos, se puede analizar gráficamente y seleccionar aquellos que pueden considerarse atípicos. Como se ha comentado, no es el caso de esta tesis, así que tras repasar las principales estrategias de tratamiento de atípicos que existen se indicará cuáles pueden ser de aplicación para el servicio diseñado y desarrollado dentro de la solución de este manuscrito.

La existencia de valores atípicos en las series de datos de los indicadores es un elemento no despreciable a la hora de obtención de un CI. Según fuentes, la anomalía introducida por dichos valores dificulta la aplicación de métodos de estadística descriptiva (media, mediana, varianza), que dejan de ser igual de representativos; igualmente, puede influir en la generación de datos erróneos de normalización (por ejemplo, si se utiliza el método min-max) y, por tanto, dañar la coherencia en el análisis y cálculo del CI [417].

Existen algunos métodos para la corrección de estos valores atípicos. En primer lugar, una inspección visual permitiría su eliminación. En segundo lugar, para conjuntos de datos más grandes, pueden utilizarse las herramientas de análisis de distribución estadística de inclinación (*skewness*) y curtosis [418]. A partir de éstos, existe un método denominado *Winsorisation* [385] que propone el tratamiento de dichos puntos problemáticos (en lugar de eliminarlos) de dos posibles formas:

- (1) “Acercando” los valores extremos (atípicos) al borde del conjunto de datos centrales.
- (2) Utilizando los valores de *skew* anteriores, los atípicos son reemplazados por los puntos de dichas curvas que se encuentran en el margen aceptado.

En tercer lugar, la literatura documenta el método logarítmico (Box-Cox) de corrección de atípicos. Este método se aplica cuando desde la inspección visual se detectan muchos valores atípicos (según referencia [417], a partir de 5) o ninguno, y la solución pasa por una transformación logarítmica de toda la serie de datos del indicador.

Estas operaciones requieren una complicación estadística que se consideró innecesaria para el alcance de esta tesis doctoral. Además, dichos métodos parten



de la base de un conocimiento previo de todo el *set* de datos, de lo que no se dispone dada la naturaleza de la solución. Como se expone en secciones anteriores, se ha propuesto como norma general (y a voluntad del usuario final del servicio) la aplicación la norma  $1.5 \times \text{IQR}$  [419] sobre los datos disponibles de registros anteriores (previos a la puesta en marcha del cálculo dinámico del indicado), lo que supuso (en los escenarios de validación, en aquellos donde se aplicó) un recorte del 5% total de las entradas en una ejecución del CI (por indicador).

Tras la revisión del tratamiento estadístico de valores atípicos, el último ejercicio sobre las series de indicadores previo paso a las operaciones de cálculo del CI fue la gestión de **valores ausentes**. Estas situaciones se dan cuando algunos de los valores de los indicadores no se completan (deliberadamente) o no están disponibles [417]. En el caso de esta tesis, estas situaciones responden a aquellas en las que un nuevo dato que está siendo esperado (debido a la información previa contenida en el campo *nextcalculation* según Tabla 4.5: Campos establecidos en el modelo de datos acordado *cKPI*), no llega a registrarse en el sistema. La existencia de valores ausentes puede tener un impacto en las varianzas y correlaciones del índice compuesto final.

Según la literatura, estos datos ausentes pueden ser catalogados en:

**Tabla 4.2:** Tipos de valores ausentes

| Tipo | Descripción  |
|------|--|
| MCAR | Completamente aleatorios ( <i>Missing completely at random</i> ). La probabilidad es la misma para todas las observaciones. Esta probabilidad sólo depende de los parámetros externos. |
| MAR  | Aleatorios ( <i>Missing at random</i> ). La probabilidad de que falte depende de otras variables en los datos.   |
| NMAR | Sin aleatoriedad ( <i>Not missing at random</i> ) Se trata de los casos en que no se informa de un valor a propósito.  |

A la hora de corregir estos valores ausentes, se debe tener en cuenta que, dentro de la solución planteada en esta tesis, no puede saberse con certeza a qué caso corresponde de los anteriores ya que no se dispone de toda la visión de los datos en un amplio espacio de tiempo. Además, todas las soluciones planteadas incluyen asunciones que pueden entenderse como disminución de la precisión del CI [380]. Según fuentes, si un set de datos contiene más de  $1/3$  (un tercio) de valores ausentes, no puede aplicarse ningún método de los mencionados a continuación, sino que debe descartarse dicha serie temporal completa.

Según la literatura, existen tres mecanismos para la imputación de datos ausentes. En primer lugar, se propone no considerarlos para los cálculos dentro de las series. Esta opción se considera generalmente como válida, ya que en términos prácticos sería equivalente a la imputación de la media de valores de la serie de ese indicador (por tanto, se trata de una forma sutil de imputación) [417].

En segundo lugar, se encuentran los métodos de imputación singulares, que asumen una modelización particular del valor ausente en función de valores anteriores de dicha serie, del valor perdido o de valores adyacentes [420]. Se describen brevemente en la tabla:

**Tabla 4.3:** Métodos singulares de gestión de datos ausentes

| Método                           | Descripción  |
|----------------------------------|--|
| Modelado implícito               | Realizar una asunción e indicar dicho valor por defecto en todas las ocasiones.  |
| Modelado explícito               | Imputación manual de cada dato en base a criterios estadísticos del diseñador del índice o usuario del servicio.   |
| <i>Hot-deck</i>                  | Imputación con un dato similar (con una serie con características lógicas similares).  |
| Sustitución                      | Sustitución por un valor que aún no haya sido imputado.  |
| <i>Cold-deck</i>                 | Reemplazo por un valor fijo por serie basado en referencias externas (por ejemplo, medias obtenidas en la literatura).   |
| Media/mediana/moda incondicional | Se reemplaza el valor por la mediana/media/moda del histórico de su serie (dentro del periodo o a más rango).  |
| Por regresión                    | Se aplica un modelo de predicción por regresión.   |
| Maximización de la expectación   | Se trata de una resolución iterativa de un problema de optimización del error sobre el valor esperado de toda la serie de datos (el más probable). Más detalles en [385] |
| K vecinos más cercanos           | Este método se basa en el cálculo de la distancia relativa a cada uno del resto de indicadores aplicando algoritmos de KNN (existen muchos en la literatura) [417]       |

En tercer lugar, existen métodos de imputación múltiple como las cadenas de Markov, Monte Carlo u otros algoritmos de cálculo numérico [417]. El objetivo de estos métodos es iterar sobre un proceso de asignación aleatorio reduciendo la incertidumbre en cada paso. Según [385], cualquier método de regresión y estimación puede ser utilizado para obtener estimaciones combinadas, las cuales a partir de una análisis de métricas de error estandarizadas, pueden ser iteradas y mejoradas.

Tras un análisis de todos los métodos anteriores (apoyado por las investigaciones realizadas en el proyecto PIXEL [421] [422]), se llegó a las siguientes conclusiones de cara a la inclusión de dichos métodos en el servicio a desarrollar en la tesis doctoral:

- No se dispone de la serie completa de datos (únicamente del periodo de cálculo, y al tratarse de una solución orientada a IoT puede ser muy bajo, por tanto, con pocos registros), así que los métodos que requieren inspección y análisis de dicho *set* global, deben ser descartados.

- Aquellos métodos que requieren grandes recursos computacionales (por regresión, maximización de la expectación, KNN, de imputación múltiple) son descartados para poder proponer una solución ligera factible para escenarios de *edge computing*.
- Aquellos métodos que dependan fuertemente de la correlación entre indicadores (de imputación múltiple, maximización de la expectación, diferentes tipos de modelado) son descartados ya que, siendo la solución orientada a despliegues IoT con fuentes heterogéneas y muy dispares entre sí (sensores, informes, datos de webs externas...) no puede garantizarse esta correlación.
- Según recomendaciones en la literatura [417], se debe evitar utilizar diferentes métodos de imputación de valores ausentes entre los diferentes indicadores, por tanto en el servicio desarrollado se ofrecerá al usuario la posibilidad de seleccionar uno; que será automáticamente aplicado a todos los indicadores.
- En la literatura se recomienda la imputación de valores ausentes después de la normalización de los indicadores. No obstante, esto no es posible en la solución propuesta ya que esta imputación debe realizarse previa conjugación de los datos (véase sección 4.2.2).

Analizando lo anterior, se propone incluir de forma automatizada las posibilidades de imputación *hot-deck*, *cold-deck*, media/mediana/mode incondicional (se seleccionó que la media es el valor apropiado) e imputación explícita manual.

**Resumen de cobertura en la tesis de la creación del marco inicial de índices compuestos:**

El servicio a desarrollar dentro de la solución no proporciona un soporte software para la ideación del índice compuesto. Esto debe ser realizado *offline* por expertos en la materia y agentes involucrados. Se ha aportado un flujo de decisión como soporte a la definición de los indicadores necesarios para el cómputo del CI. La frecuencia de actualización y la imputación directa de datos son cuestiones a considerar, que también quedan fuera del alcance del servicio. En relación a la preparación de los datos para paliar los valores atípicos se opta dentro del servicio por una aproximación del 5% y en cuanto a los valores ausentes, se preparará el software para aceptar las opciones *hot-deck*, *cold-deck*, media/mediana/mode incondicional e imputación explícita manual. Se han seleccionado estos métodos por el beneficio de la usabilidad y flexibilidad de la solución.

### 4.1.2.2. Operaciones matemáticas sobre los indicadores

Una vez los indicadores han sido identificados y las series de datos se encuentran preparadas, puede empezar la serie de operaciones matemáticas para obtener el CI.

En primer lugar, debe realizarse una **normalización**, que transforme los valores de las series a una escala común. Este paso se realiza para poder realizar una posterior agregación, ya que los datos pueden ser muy diversos tanto en unidades (por ejemplo, kg, kWh, ppm...) como en rango de escala. Con esta operación, que se realiza indicador a indicador, se consigue que los diferentes indicadores que pueden utilizarse para construir el indicador compuesto se ajusten a la misma norma [385]. Por tanto, en este paso los indicadores se transforman en números puros y adimensionales.

En este sentido, se ha decidido que para la solución de esta tesis esta normalización debe reescalar los indicadores a valores entre 0 y 1. Aquí es importante realizar una revisión de la estrategia de normalización en cuanto a su direccionalidad; se desea que cuando los valores normalizados se acerquen a 1 signifiquen interpretativamente un mejor rendimiento/impacto de la entidad por/sobre dicho fenómeno, mientras que 0 debería significar lo contrario. Este aspecto es importante ya que marca las posibilidades de normalización a escoger.

Se han encontrado muchos métodos propuestos en la literatura para la estandarización de variables. No obstante, no existe un método único que deba ser escogido para todos los casos. Esta elección debe depender, en gran medida, del caso de estudio particular, del significado del CI. En la literatura se proponen muchos métodos para la estandarización de las variables y aunque no hay un método perfecto, la elección debe orientarse a la “mejor manera” para el caso de estudio, el marco conceptual y las propiedades de los datos [388].

En las siguientes páginas se exponen los principales métodos de normalización encontrados en la literatura, refrendados por varias referencias y ampliamente descritos (principalmente) por el mencionado grupo de la JRC de la Comisión Europea [423][380][384]. Se incluyen, además, comentarios sobre las ventajas e inconvenientes de cada uno y su encaje para los usos generales planteados de la solución de la tesis (enmarcada en un entorno de series temporales IoT “en tiempo real”).

#### **Raking de indicadores:**

El método más sencillo. El objetivo es comparar los valores de todos los indicadores y asignarles como valor normalizado su posición (número ordinal) dentro de dicha clasificación.

$$\text{Valor normalizado}_{ind}^t = \text{Ranking}(\text{Observación}_{ind}^t) \quad (1)$$

Este método es relativamente fácil de aplicar, pero no parece adecuado para el planteamiento de la solución de la tesis. En primer lugar, al tratarse los indicadores de series temporales, esta clasificación debería realizarse en cada punto del tiempo (con cada nueva entrada de un *cKPI* – véase sección 4.2.1). Como se ha comentado, los indicadores pueden provenir con diferentes frecuencias, lo que dificultaría esta clasificación. Además, este método conlleva a una pérdida de información del valor absoluto del indicador, que es algo que no se busca en general en los casos que se han estudiado (véase sección 5). Mientras que esta técnica puede ser interesante para indicadores de ámbito social o de comparación entre (por ejemplo) países, no parece lo más recomendable para datos con cariz IoT.

#### **Estandarización (o Z-scores):**

Este método toma como referencia valores estadísticos de la serie temporal de cada indicador. En particular, considera que se encuentran accesibles tanto la media como la desviación estándar en cada uno. La fórmula de normalización es

$$\text{Valor normalizado} = \frac{[\text{Observación} - \text{Valor\_medio}]}{\text{Desviación\_estándar}} \quad (2)$$

Este método se considera muy interesante y es el más utilizado en el estado del arte por un motivo principal: hace que el valor medio y la desviación estándar de cada serie normalizada sean 0 y 1 respectivamente (*media=0, varianza=1*). Este hecho hace que no se introduzcan distorsiones durante la agregación, lo cual es muy conveniente. No obstante, la escala de rango de cada indicador normalizado no es fija (no se obtiene un *cKPI* entre 0 y 1). Aunque esto puede corregirse más adelante durante los pasos de agregación y ponderación, la interpretabilidad de cada indicador podría verse afectada, necesitando una contextualización en cada caso. Se decidió mantener esta opción como válida para la solución de la tesis ya que puede haber casos en los que se desee no propagar posibles errores en la agregación debido a la diferencia de medias entre indicadores.

Si los datos dependen del tiempo, la media y la desviación estándar se calculan para el tiempo de referencia deseado (es decir, sobre la periodicidad acordada para el cálculo del CI, según la subsección anterior). Por tanto, dentro del servicio de la solución, a la hora de aplicar esta normalización, deberán efectuarse llamadas adicionales a la base de datos indicando la necesidad del cálculo de la media y desviación estándar de cada serie temporal (indicador) en dicho periodo. La fórmula particularizada a datos con carácter temporal es la siguiente:

$$\text{Valor normalizado}_{ind}^t = \frac{x_{qc}^t - x_{qc=\bar{c}}^t}{\sigma_{qc=\bar{c}}^t} \quad (3)$$

#### **Re-escalado (mín-máx):**

Este método es uno de los más utilizados ya que consigue, por su naturaleza, expresar cada ocurrencia de un indicador en un valor normalizado entre 0 y 1. Para conseguirlo, hace uso de los valores mínimo y máximo de ese indicador en el periodo

acordado para el cálculo. Además, es especialmente interesante en el caso de esta tesis ya que tiene en cuenta la direccionalidad de la normalización mencionada en la página anterior (0 peor rendimiento y 1 mejor rendimiento).

$$\text{Valor normalizado } (t) = \frac{[\text{Observación } (t) - \text{mínimo}(t)]}{[\text{máximo} - \text{mínimo}]} \quad (4)$$

Aquí, puede tomarse una decisión. Los valores máximo y mínimo del denominador pueden corresponder a un solo indicador (el que busca ser normalizado) o para todos los indicadores de la entidad adoptante. En el caso de este trabajo, se ha decidido optar por el primer caso, ya que como se ha comentado los indicadores (*cKPI*) se corresponderán a fuentes de datos heterogéneas sin necesidad de estar en escalas similares.

Como inconvenientes de este método, se destaca que es altamente sensible a valores atípicos (ya que tanto el valor máximo como el mínimo considerados podrían ser atípicos y por tanto distorsionar la normalización). En este sentido, se ha incorporado en la solución la regla 1.5xIQR o 5%, como se ha mencionado anteriormente. No obstante, sigue siendo un riesgo que debe asumirse de manera consciente si se decide seleccionar este método.

$$\text{Valor normalizado}_{ind}^t = \frac{x_{ind}^t - \min_c(x_{ind}^{t_0})}{\max_c(x_{ind}^{t_0}) - \min_c(x_{ind}^{t_0})} \quad (5)$$

Además, su utilización en entornos de series temporales (como es el caso, véase la fórmula anterior) puede conllevar a errores al comparar la evolución en el tiempo ya que nuevos valores serán introducidos en la serie que podrían reemplazar a antiguos máximos y mínimos, haciendo las normalizaciones previas parcialmente inconsistentes con las actuales. De nuevo, si se es consciente de esta situación a la hora de interpretar el CI tras usar reescalado en normalización, debería poder ser considerado un método viable.

#### **Distancia a una referencia:**

Este método consiste en normalizar cada indicador contra un valor de referencia del mismo. En el contexto de la solución de este trabajo consistiría en utilizar una referencia externa (connotaciones de privacidad en caso de comparación entre varias entidades – véase sección 4.2.4). Conceptualmente, puede asumirse que dicha referencia sería el “valor ideal” a alcanzar por parte de ese indicador, de tal manera que la normalización en este caso consistiría en cuantificar “cuán lejos” de dicha referencia ideal está dicho indicador.

$$\text{Valor normalizado}_{ind}^t = \frac{\text{Obsevación}_{ind}^t}{\text{Valor de referencia}_{ind=\bar{c}}^{t_0}} \quad (6)$$

Si se utiliza este tipo de normalización debe considerarse la inversión de los valores dentro de los agentes de conexión de datos (3.3.1) o la introducción cuidadosa de los valores de referencia, ya que con el método de distancia la

direccionalidad de la normalización cambia. Si se trata de un indicador que debe ser cuanto más bajo posible para considerarse “mejor” (por ejemplo, emisiones de CO<sub>2</sub>), un valor normalizado cercano a 0 sería mejor que cercano a 1. Además, este tipo de normalización (sin inversión en el agente) haría que los valores normalizados pudieran ser mayores que 1, indicando un peor rendimiento.

Aquí, existen varias opciones. Puede considerarse que la referencia, es el valor inicial de cada indicador de la entidad, siendo actualizada a medida que nuevas (mejores) ocurrencias sean registradas. Esto permitiría una buena monitorización de la evolución en el tiempo de cada entidad, no obstante no resulta útil en casos que se requiera comparación entre los índices de diferentes entidades adoptantes. Para la solución de esta tesis, se ha considerado el método de distancia a una entidad de referencia como uno de los más adecuados, en especial cuando se requiere realizar dicha comparación (por ejemplo, escenario de validación PEI – sección 5.4). Además, se ha documentado que este tipo de método se utiliza normalmente en indicadores medioambientales, lo cual, por cercanía a los casos de aplicación de la tesis, resulta altamente conveniente [421].

#### Escalas categóricas:

Este método consiste en asignar una puntuación (valor normalizado) en función de una categorización de los valores de entrada de los indicadores. Requiere una clara justificación de las categorías y sus puntuaciones (según la literatura pueden ser numéricas o cualitativas, aunque dentro de la solución de este trabajo solo podrían ser cuantitativas). Estas puntuaciones y la selección de rangos por categoría puede ser desde arbitraria a estar basada en distribuciones estadísticas, pasando por una asignación por porcentaje de ocurrencia de valores de cada indicador. Normalmente se utilizan valores normalizados entre 0 y 100 [423] aunque para el caso de esta tesis podrían normalizarse entre 0 y 1 realizando una simple división.

$$\begin{aligned}
 &\text{Si } \textit{Observación}_{ind}^t \text{ está en el 5\% superior, entonces } \textit{Valor normalizado}_{ind}^t = 100 \\
 &\text{Si } \textit{Observación}_{ind}^t \text{ está en el 15\% superior, entonces } \textit{Valor normalizado}_{ind}^t = 80 \\
 &\text{Si } \textit{Observación}_{ind}^t \text{ está en el 35\% superior, entonces } \textit{Valor normalizado}_{ind}^t = 60 \\
 &\dots
 \end{aligned}
 \tag{7}$$

Este método tiene varios inconvenientes: (1) debe justificarse con antelación la selección de las categorías, lo que para un *set* de datos total desconocido resulta complicado, (2) se pierde información sobre el valor absoluto del indicador, lo que, en el caso de mediciones con sensores, valores de tráfico, valores de consumo, etc. resulta inconveniente, (3) se pierde información sobre la media y la varianza de cada indicador y (4) puede dificultar la interpretabilidad de los valores normalizados de los indicadores de cara a un informe final tras la ejecución del CI (requisito expresado en la sección 3.1.1.3).

Por tanto, se decidió no incorporar este tipo de normalización a las opciones elegibles dentro del servicio de esta tesis.

Por último, se documentan otros tres métodos de normalización que son menos utilizados en la literatura y que han sido descartados de las opciones a incluir en la solución ya que presentan algunos inconvenientes:

**Transformación logarítmica:**

Consiste en la normalización de cada valor del indicador realizando la operación matemática de su logaritmo neperiano. Si bien es cierto que este método ayuda a relajar los valores extremos y atípicos, y requiere pocos recursos de computación, hace que los valores normalizados sean difícilmente interpretables e introduce incertidumbre previa agregación de los indicadores.

$$\text{Valor normalizado}_{ind}^t = \ln(x_{ind}^t) \tag{8}$$

**Indicadores por encima o por debajo de la media:**

Similar al método de escalas categóricas, el objetivo de este método es otorgar un valor de normalización 0, 1 o -1 función de si el valor del indicador es superior o inferior a unos umbrales arbitrarios sobre/bajo la media de su serie temporal.

$$\begin{aligned} \text{Si } \frac{x_{qc}^t}{x_{qc=\bar{c}}^{t_0}} > (1 + p), & \quad \text{entonces } I_{qc}^t = 1 \\ \text{Si } \frac{x_{qc}^t}{x_{qc=\bar{c}}^{t_0}} < (1 - p), & \quad \text{entonces } I_{qc}^t = -1 \\ \text{Si } (1 - p) < \frac{x_{qc}^t}{x_{qc=\bar{c}}^{t_0}} < (1 + p), & \quad \text{entonces } I_{qc}^t = 0 \end{aligned} \tag{9}$$

Este método no es apropiado para los casos de la tesis ya que pierde casi absolutamente la información de los valores, haciendo imposible un ajuste fino y una monitorización de la evolución temporal del impacto de la entidad sobre el fenómeno real bajo estudio.

**Porcentaje de diferencias anuales sobre años consecutivos:**

Este método parte de la base que únicamente quiere realizarse una comparación de los valores de los indicadores con aquellos obtenidos en periodos anteriores, por tanto la normalización toma como referencia los registros de periodos antiguos (el periodo de cálculo anterior, el último mes, año...).

$$I_{qc}^t = \frac{x_{qc}^t - x_{qc}^{t-1}}{x_{qc}^t} \tag{10}$$

Este método asume que estos datos están disponibles y que cada serie temporal solo tiene sentido con la perspectiva de su diferencia a la anterior. No obstante, en el caso de la solución, cada ocurrencia de un indicador debe poder ser interpretada de manera singular, sin necesidad de recurrir a registros anteriores.

Como conclusión, se han catalogado como aptos para la solución en esta tesis los métodos de estandarización, reescalado y distancia a una referencia.



Tras la normalización, una vez los datos de indicadores (*cKPIs*) se encuentran en una escala común, pueden ser agregados. No obstante, previo a la agregación se introduce un paso extremadamente relevante: la **ponderación** de cada indicador. En el sentido lógico de la interpretación, se trata de la importancia que cada indicador tiene con respecto a su contribución al índice compuesto (CI) final.

Esta es una acción crucial que ha sido ampliamente estudiada en la composición de CIs [424]. Se trata de un proceso complejo para el que no hay una solución única (*one-size-fits-all*), introduciendo desafíos a la hora de elegir el mejor método. De manera similar a lo expuesto para la selección de indicadores, resulta imprescindible exponer de manera clara el método escogido en cada caso para que el CI resultante sea válido y creíble [425][426].

En las siguientes páginas se exponen los métodos existentes según la revisión de la literatura, sus características, ventajas e inconvenientes y su encaje para la inclusión de los mismos como opciones válidas en la solución de la tesis.

Según las referencias consultadas, no existe un método ideal único para la asignación de factores de ponderación (pesos) a cada indicador a la hora de construir un CI y probablemente nunca exista; siendo ésta más bien una decisión subjetiva (similar al caso de normalización) [427][425][385][428]. En la tabla que sigue se realiza una clasificación general, aportando un detalle de los métodos más relevantes inmediatamente después.

**Tabla 4.4:** Tipos de métodos de ponderación

| Tipo                             | Descripción y métodos incluidos   |
|----------------------------------|---|
| Ponderación equitativa           | Único método en los que todos los indicadores de una agrupación son asignados el mismo factor de ponderación. Se trata del método más utilizado con diferencia. Según [428], utilizado el 46,88% de los casos de los CIs identificados hasta 2017.  |
| Basados en métodos estadísticos. | Se trata de métodos que utilizan herramientas estadísticas sobre los valores de las series temporales (análisis de los componentes principales, regresión y beneficio de la duda). Según el mismo estudio, el 30,21% se decantaron por alguno de estos métodos.   |
| Basados en opinión externa       | Métodos que parten de una consulta (a partir de preguntas de varios tipos) a externos en la materia o la opinión pública general para obtener información sobre la relevancia absoluta o relativa de cada indicador. Los métodos principales son BAM, AHP y opinión pública. Su uso es menor, 15,62% según el estudio mencionado. |

**Ponderación equitativa:**

Este método asigna un factor de ponderación igual para cada indicador dentro de su agrupación o dimensión. Como se ha mencionado, este método es el más utilizado ya que es el que se recomienda en caso de no tener suficiente conocimiento de la estructura interna de todo el *set* de datos del indicador. De hecho, también se

recomienda cuando no hay motivos estadísticos o empíricos para elegir un esquema diferente.

$$\text{Factor de ponderación (peso)}_{\text{indicador}} = \frac{1}{\text{Indicadores en grupo de agregación}} \quad (11)$$

Es de destacar también que este método no significa no aplicar ponderación, sino que se realiza sutilmente una ponderación por la media de los indicadores incluidos en dicho grupo de agregación, asumiendo que todas las variables en dicho grupo tienen valor de cara a una interpretación hacia el CI final [401].

No obstante, este método no está carente de desventajas. Asumir una distribución igualitaria de la relevancia hacia la agregación puede introducir el efecto de arrastre de imprecisiones o desequilibrios más adelante en niveles del cálculo del CI. Además, este tipo de ponderación igual puede llevar a combinar variables con un alto grado de correlación introduciendo un elemento de doble contabilidad. Según el grupo mencionado de la Comisión Europea, este enfoque es válido si todas las dimensiones (agrupaciones de indicadores) están representadas en el indicador compuesto con un número de subindicadores similar [385]. Al ser el método más sencillo y utilizado, además de su componente de baja carga computacional y estadística, se antoja como una de las opciones más recomendables de cara al servicio de esta tesis.

Dentro de los métodos basados en análisis estadístico, se describen brevemente los dos más importantes (PCA y BoD). Cabe mencionar, sin embargo, que, en línea con lo expresado para la normalización, se ha preferido relajar la dependencia a estos esquemas debido a los recursos computacionales necesarios, la disponibilidad de *set* de datos y a su baja practicidad en muchos casos en escenarios de series temporales en tiempo real.

#### **Análisis de factores o PCA** (*Principal Component Analysis*)

El objetivo de este método es reducir las dimensiones de los datos (ponderación y agregación) minimizando la pérdida de información en el proceso. Se trata de encontrar una distribución de pesos que, en combinación con el paso de adición posterior, mantenga similares niveles de varianza que los que muestran los indicadores iniciales. Para ello, establece que la agrupación de los indicadores no debería realizarse en función de su significado sino en función de su colinealidad. De este modo, pueden resolver el problema de doble contabilidad, pero requieren altos niveles de correlación entre los indicadores y desde luego dificulta la comunicación del proceso al público en general. Debido a que, una vez agrupados, algunos indicadores son “embebidos” en otros con los que mantienen alta correlación, se estaría perdiendo de nuevo información sobre los valores absolutos de dichos indicadores. Este método se recomienda para aquellos CI que contengan una gran cantidad de indicadores iniciales y que estén menos orientados a una comunicación para la toma de decisiones sobre los mismos. Por tanto, se consideró que este método no es adecuado para la solución planteada en la tesis.

**Aproximación del beneficio de la duda** (*Benefit of Doubt – BoD*):

La referencia principal de este método es [429], que consiste en el estudio de la relevancia de diferentes pesos a cada indicador en cada uno de los *sets* de datos (es decir, de cada entidad) para observar cuál es su impacto en el índice final por entidad. De esta manera, éstos pueden ser ajustados para mostrar un aspecto u otro del mismo, definidos por la observación de su comportamiento. Se trata de un método interesante para los casos de única entidad adoptante, pero impide que se realice una comparación, al no poder ser estandarizado entre diversas instancias del índice compuesto. Además, requiere una gran cantidad de datos y un análisis particular por separado, lo cual no permitió plantear este método como válido para el caso de la tesis.

En cuanto al tercer grupo de métodos, se hace un repaso de los tres más utilizados.

**Asignación de presupuesto** (*Budget Allocation Method – BAM*):

En este método se cuenta con la participación de expertos en el campo de aplicación del CI (se recomienda que no sean expertos únicamente en algún indicador particular, sino para un dominio o dimensión -agrupación de indicadores-, como mucho [425]). Consiste en plantear a dichos expertos la distribución de N puntos sobre la lista de los indicadores de una dimensión. La idea es que, a partir de esta distribución, expresen implícitamente qué indicadores son más relevantes dentro del grupo (los que obtengan, de media, mayor proporción del presupuesto). Una vez realizado, se obtiene una ponderación acorde a dicha distribución. Este método es recomendado debido a su transparencia, pero posee como desventajas la potencial subjetividad o sesgo de dichos expertos, el no estar basado en datos empíricos y que, ante una gran cantidad de indicadores, los expertos podrían indicar puntuación en condiciones de estrés debido al riesgo de pensamiento circular. En varias referencias se menciona 10-12 como el número máximo de indicadores por agrupación sobre la que solicitar a los expertos realizar la asignación. Este método resulta interesante para el caso de esta tesis doctoral ya que es importante considerar, desde el comienzo de la construcción del índice, la participación de expertos relacionados con la materia. Además, no supone un coste computacional adicional y puede ayudar en la comunicación e interpretabilidad del CI final y de los subíndices calculados.

**Proceso analítico jerárquico** (*Analytical Hierarchy Process – AHP*):

Se trata de un método análogo al anterior con la diferencia que se solicita a los expertos realizar comparaciones uno a uno (*pair-wise*) entre los indicadores. Así, dentro de un mismo grupo, los expertos deberán indicar, como si de un enfrentamiento se tratase, cuán importante es cada indicador en comparación con cada uno de los demás (normalmente usando una escala del 1 al 9 – propuesto en [430]). A partir de estos análisis, se obtiene una matriz NxN (siendo N el número de indicadores del subgrupo) que ha de convertirse en pesos asociados, pudiendo elegir varios métodos para ello [425]. Es considerado ligeramente mejor que BAM,

al poder realizar una mejor trazabilidad en las asignaciones y una más fácil corrección de errores, pero impone una indudablemente mayor carga de trabajo a los expertos externos, resultando impráctico para entornos con una razonable cantidad de indicadores. Considerando que, en el caso general, los expertos externos no siempre podrán involucrarse lo suficiente como para ejercer un AHP robusto, se prefiere la inclusión del método BAM como opción factible dentro del servicio de la tesis.

### **Opinión pública:**

Por último, el método de opinión pública (denominado también análisis conjunto – *Conjoint Analysis*, CA) es equivalente a los dos anteriores pero en lugar de contar con expertos en el campo, se expone la consulta al público en general, convirtiendo la ponderación de indicadores del CI en un proceso participativo. Dentro de este grupo puede utilizarse tanto la técnica BAM como AHP. Suelen ser utilizados en entornos de marketing o de investigaciones de consumo, no obstante no parece la mejor opción a incluir para el caso general de esta tesis, ya que existe un alto riesgo de sesgo orientado hacia las preocupaciones de la opinión pública en lugar de aquellas propias de un análisis riguroso del campo de estudio. No obstante, se ha decidido abrir la posibilidad a introducir pesos provenientes de este tipo de análisis dentro de la herramienta de la solución (a través de la introducción manual).

Para finalizar con la ponderación, cabe destacar que pueden tomarse decisiones fuera de estos métodos con la intención de alinear el proceso global a la realidad de los datos del caso en particular. Por ejemplo, podría elegirse asignar un peso inferior (manualmente) a propósito a un indicador que proporcione datos poco fiables estadísticamente o con una gran cantidad de valores ausentes, de tal manera que éste no tenga una gran influencia en el valor final del CI.

El paso de la **agregación** se encuentra íntimamente ligado a la ponderación que acaba de ser relatada. Se trata del momento en el que los indicadores correspondientes a una dimensión (agrupación), cuyo valor está en una misma escala (elegida, 0 a 1) y usando sus pesos correspondientes, se unen, avanzando hacia el próximo nivel dentro del cálculo del CI.

Según la literatura, existen tres métodos de agregación ampliamente utilizados [424]: agregación aditiva, la agregación geométrica y agregación no compensatoria, siendo la primera la más popular con mucha diferencia (86% según [428]).

### **Agregación aditiva:**

En el método de agregación aditiva más utilizado se realiza una unión de los indicadores a través de una suma aritmética ponderada. Los valores normalizados son sumados tras ser multiplicado (cada uno de ellos) por su factor correspondiente.

$$\text{Valor agregado} = p_1 I_1 + p_2 I_2 + \dots + p_m I_m = \sum_{i=1}^m p_i I_i \quad (12)$$

La combinación de esta fórmula de agregación aditiva con una ponderación equitativa resulta en lo que se conoce como media aritmética, que es el método más

obvio y utilizado. Existen otros métodos de agregación aditiva usados muy infrecuentemente [431]. Se trata de un método simple, computacionalmente ligero y muy transparente, por lo que al análisis de incertidumbre y la comunicación de los resultados serán más directos. Sin embargo, este método no está exento de inconvenientes. Se ha demostrado que, para indicadores con una alta correlación, estos métodos llevan a una interpretación sesgada del CI en la cual la dirección del error y la detección de en qué dimensión está sucediendo serán realmente complicadas [385]. Una ventaja de este método es la compensación aportada entre indicadores de bajo factor de ponderación y aquellos con alto factor de ponderación.

Por tanto, para aplicar este método con garantías, es conveniente que los indicadores sean lo más independientes posible entre sí. Afortunadamente, este es el caso que se espera para la mayor parte de aplicaciones de la solución de la tesis, ya que, al provenir de fuentes diversas, heterogéneas, del IoT, no debe esperarse una alta correlación entre los datos de estas. Se ha considerado, pues, este método, como el más idóneo para incluir dentro del servicio a desarrollar (sección 4.2.2).

#### **Agregación geométrica:**

Los métodos de agregación geométrica utilizan funciones multiplicativas en lugar de sumatorias para componer los indicadores de una dimensión. En combinación con una ponderación equitativa se obtiene lo denominado como media geométrica, el tipo de agregación más usada dentro de este método.

$$\text{Valor agregado} = I_1^{p_1} x I_2^{p_2} x \dots x I_m^{p_m} = \prod_{i=1}^m I_i^{p_i} \quad (13)$$

Las principales características de estos métodos es que los valores deben estar en la misma escala (0 a 1 en el caso de esta tesis), por tanto, debe aplicarse el mismo criterio de valores de normalización objetivo en pasos anteriores. Además, la capacidad de compensación entre indicadores con factores bajos y altos de ponderación es menor que con la agregación aditiva. Poseen, igualmente, problemas a la hora de realizar análisis de incertidumbre. En la literatura se indica que una media aritmética es siempre igual o mejor que una media geométrica, aunque en algunos casos pueda llegar a ser interesante ésta última [431]. Con relación al servicio dentro de la solución propuesta, se ha permitido la selección de agregación geométrica debido a su bajo coste computacional y por si pudiera resultar relevante en algún despliegue del sistema.

#### **Métodos no compensatorios:**

Con los dos métodos anteriores existe el efecto estadístico de compensación de valores de indicadores con bajo peso en la agregación, que son de facto sustituidos por otros con mayor factor de ponderación. Este hecho puede introducir sesgos relevantes en la interpretación del CI [428], por tanto, en los casos en que este hecho exista debe permitirse la opción de aplicar métodos no compensatorios, que impiden que este efecto ocurra. El principio de estos métodos es que no hay necesidad de obtener un valor cuantitativo resultante de la agregación entre indicadores, sino que el único objetivo es establecer un ranking entre las entidades

adoptantes. De este modo, se puede realizar una comparación de diferentes aproximaciones y determinar cuál es el ranking de dichas entidades observando el conjunto de resultados.

Se trata de generar una matriz basada en un análisis multicriterio del valor normalizado de cada indicador en el mismo subgrupo de todas las entidades. A partir de esta matriz (*outranking matrix*), y siempre con la necesaria intervención de los usuarios finales del CI que desean realizar la comparación, pueden interpretar cómo los indicadores se comportan y establecer su clasificación acorde a dicho comportamiento. Matemáticamente, se trata de realizar “enfrentamientos” uno a uno (*pair-wise*) entre dichos indicadores y realizar una evaluación sobre qué indicador es más importante en función de cuántas veces haga “vencer” a dichas entidades. Este método asume una ponderación equitativa y sustituye el cálculo matemático de la agregación por una valoración comparativa que se traduce en una matriz de clasificación, utilizando, por ejemplo, el método Copeland [431]. Una descripción completa de este método puede consultarse en [432].

Está ideado para casos en los que prima la comparación entre entidades y donde se sabe cuál es el *set* de datos completo. De esta forma, puede analizarse la “victoria” de una entidad contra otra comparando los diferentes métodos y elegir así la agregación correspondiente. Todo lo anterior, junto con la consideración que las decisiones de este método deben realizarse *offline* por parte de los responsables de la comparación y que, como norma general, los datos a incluir presentarán baja correlación, ha decidido no incluirse esta opción de manera automática dentro del servicio de la solución. Además, según algunas fuentes [433], en términos de comparación entre entidades, estos métodos (si se realizan de forma frugal) proveen resultados similares que los métodos de agregación compensatoria (aditiva y geométrica).

#### **Resumen de cobertura en la tesis de métodos matemáticos para el cálculo del CI:**

En esta subsección se han expuesto los diferentes métodos de normalización, ponderación y agregación más utilizados en la literatura para la construcción de CIs y se han realizado reflexiones sobre su adecuación al caso de la tesis. Como conclusión, se han catalogado como aptos para la solución los métodos de estandarización, reescalado y distancia a una referencia para ejercer la normalización. En cuanto a la asignación de factores de ponderación, el método equitativo, la asignación de presupuesto BAM y la imputación manual (en caso de requerir opinión pública u otra aproximación a medida), deben ser proporcionadas por la solución. A nivel de agregación, los métodos aditivos y geométricos se incorporan automáticamente dentro del servicio para ser elegidos. En cuanto a aquellos no compensatorios, su automatización completa dentro de la herramienta no es posible y requerirían un análisis pormenorizado en cada caso (a realizar *offline*).

### 4.1.2.3. Otras consideraciones

En el estado del arte relacionado con la construcción de índices compuestos se detalla una exhaustiva lista de acciones para la validación y reevaluación de los métodos utilizados, asunciones realizadas e indicadores identificados. Como se ha expuesto en la sección 4.1.2.1, estas acciones se orientan a garantizar una robustez de todo el proceso, dar sentido a los resultados (estadísticamente) y a analizar si debiera modificarse el marco inicial de alguna manera para ajustar las conclusiones.

En esta subsección se aportan unas pinceladas de las herramientas estadísticas que podrían utilizarse, principalmente según las fuentes [55] [56] y [434]. No obstante, conviene destacar que ninguna de estas operaciones ha sido incluida en el servicio automático desarrollado dentro de esta tesis doctoral. Los motivos detrás de esta decisión son los siguientes:

- (1) El servicio desarrollado dentro del objetivo de la tesis tiene como alcance previsto la especificación de la estructura y el cálculo de un indicador compuesto. Los métodos y técnicas expuestos aquí corresponden a la evaluación y análisis de un índice compuesto, acciones que pertenecen al ámbito de la estadística y de las ciencias sociales, lejos de la finalidad de esta tesis doctoral.
- (2) Deben realizarse *offline* y bajo la supervisión y decisión de personal que no está previsto que utilice una herramienta automatizada como la planteada en esta tesis.
- (3) Añadirían una complejidad y profundidad al diseño del servicio indicado que debería plantearse una tesis independiente para abordar todos los puntos relevantes. En lugar de incluir asunciones y dejar una cobertura estadística a medio camino, se ha preferido omitir estas operaciones del trabajo recogido en este manuscrito.

Las acciones recomendadas para estas consideraciones son:

- **Estudio de correlación** entre todos los indicadores, subindicadores y el índice compuesto final para encontrar sobre- o infrarrepresentaciones, sesgos, relaciones negativas, implicaciones en las ponderaciones, etc. Existen multitud de herramientas que requieren un análisis y reflexión *offline*.
- **Análisis del componente principal** (PCA) para resumir la información de todos los indicadores reduciéndola a unos pocos componentes que representen una varianza similar. Existen varios métodos (porcentajes, Kaiser-Guttman Eigenvalues, test de Cattell...) y se pueden utilizar en este contexto para analizar la consistencia del CI y verificar su multidimensionalidad. En cierto modo, son una herramienta para la validación de la división de los indicadores en agrupaciones (o dimensiones).

- **Análisis de fiabilidad** a través de la aplicación de la fórmula Cronbach's alfa. Se trata de un valor para medir la consistencia del CI. Su aplicación y escalas de interpretación están disponibles en [435].
- Como se ha visto, todos los pasos dentro de la construcción y cálculo de un CI añaden un componente de incertidumbre al valor final. Se propone realizar en este punto análisis de sensibilidad y de incertidumbre sobre todos los indicadores y sobre el CI final para averiguar cómo de robustos son los cálculos y los métodos escogidos.
- Por último, se hace hincapié en mantener alineación con los principios TRACE (*Transparency, Accuracy, Credibility, Engagement*): manteniendo la transparencia en los procesos, haciendo públicos los datos en la medida de lo posible, tratando de hacer el trabajo replicable, realizar las menores asunciones posibles y basarse en métodos empíricos, seguir los manuales disponibles [385], ofrecer una correcta visualización y comunicación e involucrar a los agentes relacionados durante todo el proceso.

#### 4.1.2.4. Representación de resultados

El último paso incluido en la construcción de un índice compuesto está orientado a transmitir correctamente (a la audiencia objetivo) tanto los resultados como los métodos utilizados. Aunque estos aspectos son subjetivos, los principios de comunicación, visualización y representación de la información deben ser seguidos.

Según referencias en el estado del arte, una buena comunicación de los resultados de un índice compuesto se basa en los pilares [436]:

- Debe contar una historia (referencias temporales)
- Ser atractivo para el público
- Captar la atención en los aspectos importantes
- No sobrecargar la información visual
- Mostrar transparencia con los datos y métodos escogidos

En este sentido, la herramienta visual para comunicar los resultados de los cálculos de CIs ha sido diseñada en esta tesis con las siguientes características.

- a) **Comparación histórica de valores para contar una historia** Permite comparar el CI, los indicadores de diferentes niveles entre ejecuciones y periodos de tiempo, de forma individual o superpuesta. Esto facilita el análisis y la comparación de los resultados.
- b) **Carga de un valor anterior:** Es una opción interesante porque permite las opciones de visualización interactiva con los valores de una ejecución anterior del CI.



- c) **Índice de fiabilidad e información de métodos matemáticos seleccionados:** tabla completa con las fuentes de datos de entrada, comparando la fuente ideal con la utilizada. Este punto, junto con la información de los métodos contribuye a mejorar la transparencia y no esconder posibles defectos de la ejecución del CI.
- d) **Creación de un escenario hipotético:** La herramienta proporciona funcionalidad de introducción de datos de forma manual, así como la solución (a través de los agentes) permite la simulación de datos de entrada, dando la posibilidad al usuario de crear escenarios hipotéticos para poder centrarse en la evolución de algunos indicadores de interés.
- e) **Informe con recomendaciones:** Además de la información incluida en los gráficos de la pantalla principal, se puede descargar un informe con los resultados de un modo más formal, incluyendo recomendaciones asociadas, orientadas a la toma de decisiones relacionadas con el fenómeno a representar.

Como puede verse en las capturas aportadas en la sección 3.5, esta herramienta visual (*layout* diseñado por el candidato) cumple con los principios establecidos en la teoría de construcción de CIs. Se aporta una vez más en las imágenes 4.4 y 4.5 dos ejemplos visuales de los resultados de una ejecución.

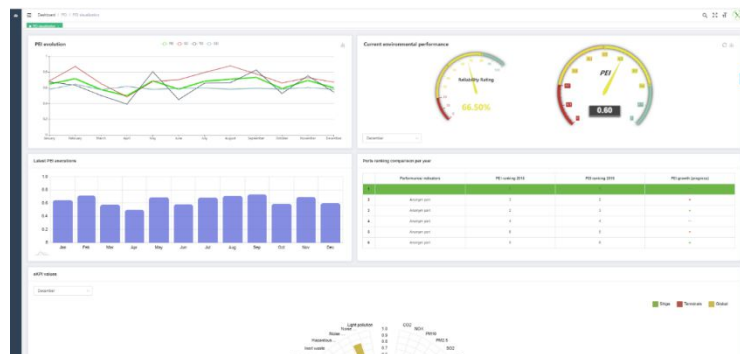


Figura 4.4: Representación del CI – Ejemplo I

Dashboard / PEI / Reliability Ratings

| Data origin    | Subindex                            | Piece of data                         | Optimal retrieval way | Current retrieval way         | Reliability Rating | Aggregated RR (subindex) | Aggregated RR (origin) |
|----------------|-------------------------------------|---------------------------------------|-----------------------|-------------------------------|--------------------|--------------------------|------------------------|
| Ships          | Air Emission                        | IMO number and gross tonnage of ships | Real time API         | Periodic API                  | 80.56 %            | 83.47 %                  |                        |
|                |                                     | Main and auxiliary engine of ships    | Periodic API          | Average value from literature | 36.84 %            |                          |                        |
|                | Berth and maneuvering time of ships | Sensors                               | Periodic API          | 85.40 %                       |                    |                          |                        |
|                | Waste                               | MARPOL annexes about waste            | Pixel proxy tool      | Periodic API                  | 97.14 %            |                          |                        |
|                | Wastewater                          | MARPOL annexes about wastewater       | Pixel proxy tool      | Periodic API                  | 85.29 %            |                          |                        |
| Terminal       | Wastewater                          | Wastewater produced by the terminals  | Pixel proxy tool      | Handtyped forms               | 21.95 %            | 21.95 %                  | 21.95 %                |
| Port authority | Wastewater                          | Wastewater produced by the PA         | Pixel proxy tool      | Handtyped forms               | 9.76 %             | 9.76 %                   | 9.76 %                 |

2022 | January

Figura 4.5: Representación del CI – Ejemplo II

### 4.1.3. Herramientas software disponibles

Uno de los objetivos de la tesis doctoral es plasmar los pasos indicados en la subsección anterior en un software utilizable dentro de una arquitectura de IoT. En los diferentes puntos anteriores se ha hecho un repaso de aquellas acciones que no se consideró necesario cubrir en el software (servicio) a desarrollar. En esta subsección se hace un repaso cronológico a las herramientas que han surgido y que están disponibles que realizan (todas o algunas de) las funcionalidades expuestas. Se trata de un punto realmente relevante en la tesis doctoral ya que se ha generado una solución que no existía en el estado del arte. Aunque es cierto que hay varias tecnologías consolidadas (por ejemplo, sistemas SCADA [437]) que podrían entenderse como representantes de KPIs, tras un sondeo inicial al comienzo de la tesis doctoral, se encontraron únicamente unas pocas referencias a herramientas que cubrieran específicamente índices compuestos:

El producto **CACIS**, de la OCDE, existe desde el año 2010 [438]. Se trata de un paquete de software para ejecución en entornos Windows centrado en indicadores de ciclos de negocio (*business cycles*).

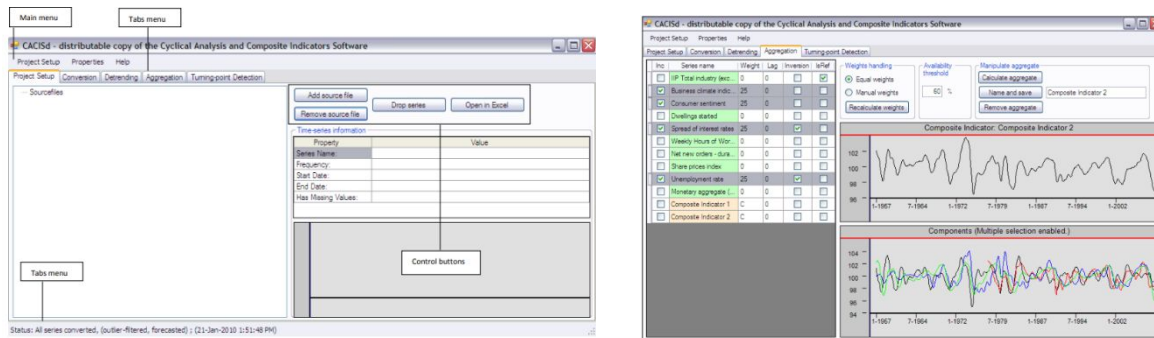
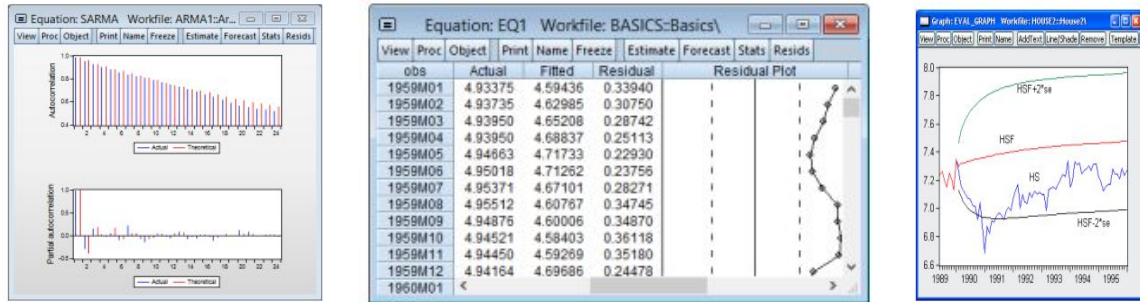


Figura 4.6: Herramienta CACIS de la OCDE. Fuente: [438]

Se trata de un software descargable desde la página de OCDE, pero tras su análisis se comprendió que es una solución inflexible desde el punto de vista de datos aceptados (únicamente preparada para datos concretos brindados por la propia OCDE, con indicación precargada de las series a analizar y que requiere carga manual en formato Excel o CSV), personalización e interfaz de usuario (sin cambios desde 2010). Además, el hecho de ser una aplicación únicamente ejecutable en entorno Windows reduce sus posibilidades de integración con datos reales en entornos distribuidos o de computación ubicua/IoT. Como puntos positivos, es una aplicación muy enfocada a la computación de indicadores cíclicos, por tanto, las herramientas matemáticas que incluye son muy variadas y completas (por ejemplo, detección de puntos de inflexión – *turning-point*-, detección de valores atípicos o la introducción de filtros de series de datos como Christiano-Fitzgerald). Además, incluye predicción de todos los indicadores a un periodo cíclico vista usando un modelo ARIMA. Como punto a la vez positivo y negativo, requiere mucha intervención (por tanto, conocimiento técnico) del usuario en todos los pasos. Está

enfocado en series económicas, lo cual, junto con todo lo anterior, hizo que se descartara como una solución válida para los objetivos de la tesis doctoral.

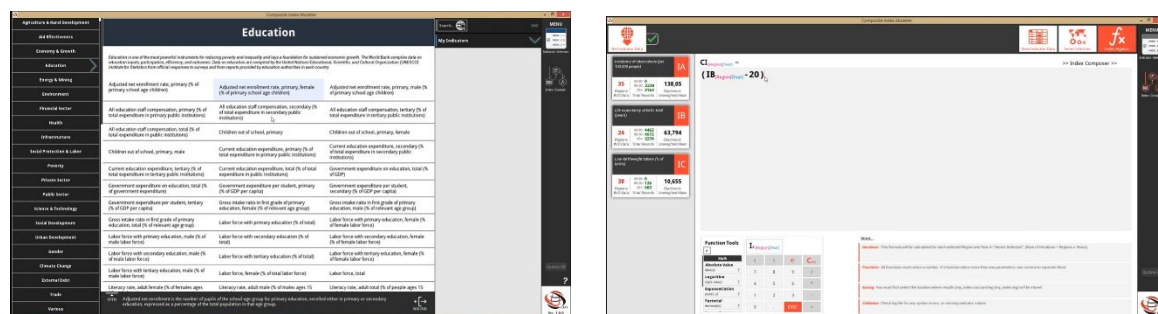
Algo más tarde, en 2014, la compañía IHS Global Inc. ofreció un producto llamado **EViews** [439]. Se trata de un producto comercial (por tanto, no disponible de manera abierta) centrado en el análisis y predicción de series temporales relacionadas con ciclos de negocio.



**Figura 4.7:** Herramienta EViews de IHS Global Inxc. Fuente: [440]

Se trata de una solución muy completa para el análisis de series temporales. Es una aplicación para entornos Windows con bajas características de personalización y que requiere un alto conocimiento de su lenguaje. Como punto positivo, esta herramienta permite cargar información desde una base de datos determinada, además desde archivos CSV o Excel. La última versión data de noviembre de 2020 [440] e incluye una gran cantidad de funciones interesantes como predicción de regresión lineal sobre los datos, agrupaciones, correlaciones, gráficas, ecuaciones y modelos avanzados sobre los datos... No obstante, se identificó como una solución inadecuada para el trabajo de la tesis ya que: (i) es privativa y conlleva un coste, (ii) no permite la interacción con los datos para la construcción de índices compuestos, (iii) es altamente compleja y requeriría un gran esfuerzo de aprendizaje, fuera del alcance de este trabajo.

Un año más tarde apareció la primera herramienta más flexible en términos de despliegue y de creación por parte de la comunidad de software libre. El *Composite Index Modeler (CIM)* [441], publicado por Geogouz Xenakis se trata de una herramienta de creación y visualización de índices a partir de los indicadores de desarrollo del banco mundial (*WorldBank's World Development Indicators database - WDI*).



**Figura 4.8:** Herramienta CIM. Fuente: [441]

A partir de los datos generados por esta entidad (altamente estructurados y con los índices muy identificados), un usuario puede seleccionar indicadores, cargar los valores de dichos indicadores en un periodo determinado y aplicar sobre ellos funciones algebraicas (limitadas) para construir índices compuestos. Los resultados son descargados en un archivo y pueden ser visualizados sobre un mapa (tipo mapa de calor). Como principales inconvenientes de esta herramienta, no dispone de una capacidad de conexión con APIs externas y, principalmente, está muy supeditado a la información de esta fuente de datos (WDI). Se trata de información contextualizada a indicadores de países, sin responder a las necesidades de tiempo real y de datos heterogéneos que se desean en esta tesis doctoral. Además, las combinaciones de las series se basan en operaciones algebraicas y no en los pasos de construcción de índices compuestos indicados en la sección 4.1.2.

Observando en el ámbito de la investigación, en 2017 apareció otra propuesta interesante centrada en la creación de indicadores compuestos para *Smart Cities*, resultado de un proyecto europeo iniciado en 2015. El proyecto **CITYkeys** (financiado por la UE) [442] se centró en el descubrimiento y la aplicación de CIs para *Smart Cities*, definiendo un vasto conjunto de indicadores procesables, que incluían la ciberseguridad, la producción local de alimentos, el estilo de vida saludable y el uso de la planta baja, entre otros; observando claramente a los actores que intervienen, diferentes ejemplos de despliegue y descripciones exhaustivas [443].

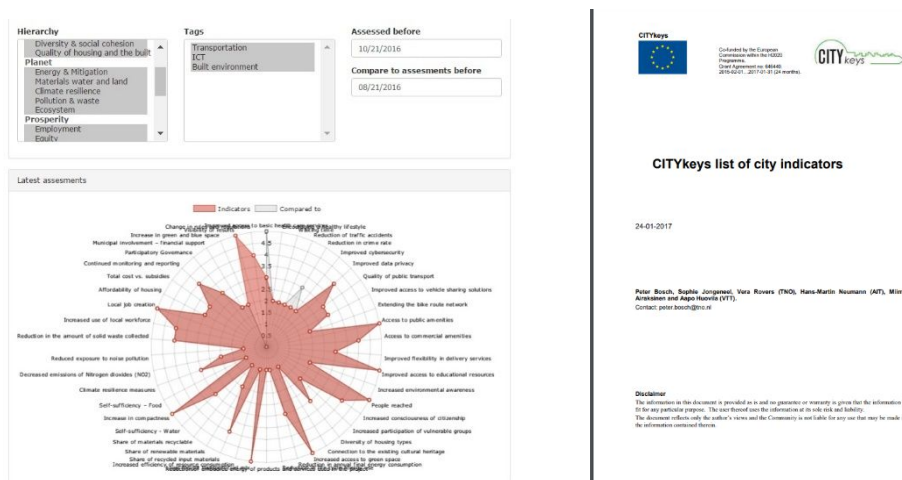
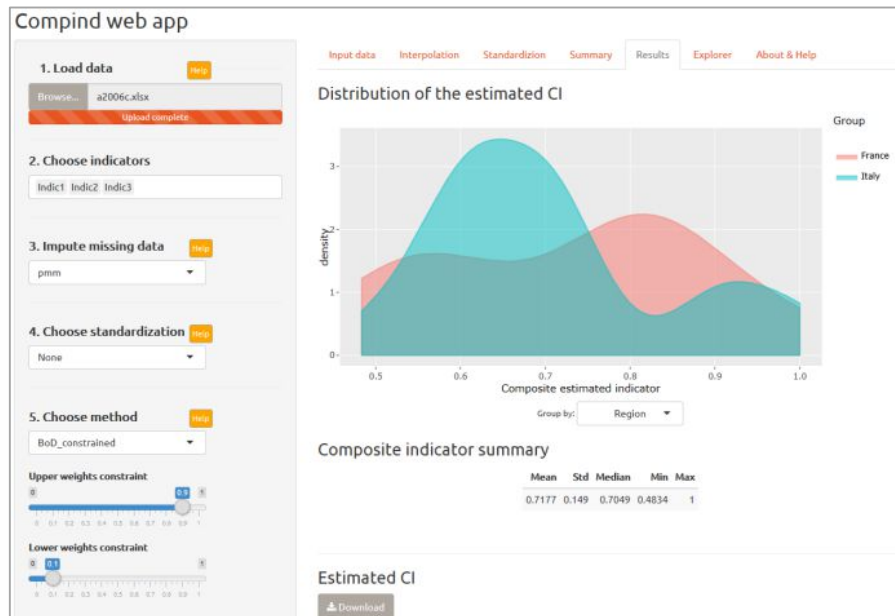


Figura 4.9: Herramienta del proyecto CITYKeys.Fuente: [442]

Sin embargo, el cálculo se basó en su mayoría en evaluaciones tipo Likert, en ponderaciones simplistas (por ejemplo, [444]) y en el uso de datos estáticos procedentes de bases de datos de informes o de literatura (por tanto no se ajusta a los requisitos expuestos en la sección 4.1.2). Aunque se propuso un sistema de evaluación de indicadores (KPIs) [445], el trabajo en CITYkeys no permite importar datos desde fuentes en tiempo real ni la construcción de índices compuestos, así como no expone la metodología para el cálculo de los mismos. Por tanto, aunque interesante y más cercano (en términos de contexto y sector) este trabajo se asumió como incompleto a efectos de esta tesis doctoral.

La siguiente referencia en términos cronológicos es una de las más principales a nivel de usabilidad y constituyó una enorme inspiración para el desarrollo del servicio. Se trata de la herramienta **Compind** [427], disponible en GitHub en el año 2018, acompañada de un artículo publicado en el que se describió su funcionalidad y objetivo.



**Figura 4.10:** Herramienta Compind. Fuente: [427]

Se trata de una solución muy visual desarrollada en lenguaje R que se enfoca en la validación diferentes métodos de ponderación. Según sus afirmaciones, ya que no existe un método oficial única para ponderar indicadores de cada a un índice compuesto, proponen la utilización de su herramienta para validar un método alternativo (poco usado) como el de valores de frontera (conocido como beneficio de la duda – *Benefit of the Doubt* – BoD). La herramienta permite cargar datos (en un formato CSV determinado) y ejercer análisis de sensibilidad tras elegir métodos de agregación y ponderación entre los que se encuentra el mencionado BoD. De cara a su interés para la tesis doctoral, el hecho de que sea público y con una clara documentación y ejemplo ha ayudado para el diseño de interfaz y del desarrollo del servicio. No obstante, esta herramienta no permite la conexión con bases de datos que incluyan datos heterogéneos, ni está preparada para definir una estructura de varios niveles de construcción del índice compuesto (árbol). Por lo demás, aporta una buena perspectiva sobre la selección de métodos de imputación de valores ausentes, agregación y ponderación. Además, un demostrador de esta herramienta se encuentra disponible *online* [446]. Por último, es conveniente apuntar que esta herramienta (en contraste con el resto de las mencionadas) sí está en continua actualización [447].

Al final de ese mismo año (2018), apenas unos meses antes del inicio de esta tesis doctoral, una nueva herramienta fue publicada y puesta a disposición de la comunidad como un paquete software descargable. Se trata de la herramienta



desarrollada por KTH y ETH Zürich llamada **CIAO Tool** [448], cuya publicación asociada se encuentra, además, participada por los miembros del grupo JRC de la Comisión Europea mencionados.

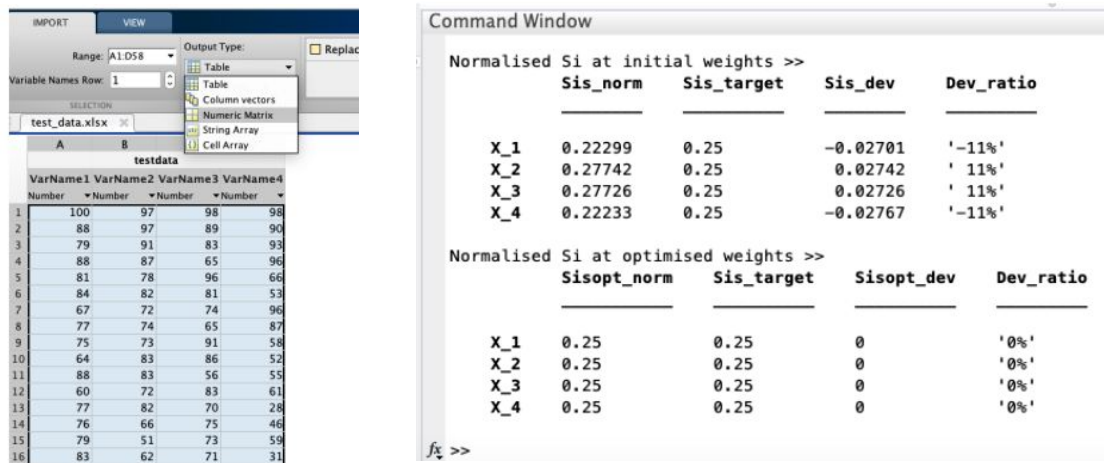


Figura 4.11: Herramienta CIAO Tool.Fuente: [448]

Se trata de una herramienta centrada en el análisis de métodos de agregación y ponderación, aportando información sobre si su selección es equilibrada; además, propone un ajuste de las ponderaciones (optimización) para que el resultado final se ajuste a la supuesta influencia de cada indicador. Así, no permite la construcción de índices compuestos únicos a partir de indicadores, como se propone en la sección 4.1.2.1 ni profundiza sobre la normalización de los datos. Por último, la documentación aportada se basa en la utilización de la herramienta sobre datos sintéticos de test importados como *.m* desde una hoja Excel. Conviene destacar que se trata de una extensión sobre MatLab probada en entornos Apple, por tanto su flexibilidad a la hora de integración con un sistema IoT en tiempo real es limitada.

Llegados a este punto, una vez comenzada la tesis, doctoral, se analizó la herramienta que sirvió como base para el resto del trabajo del servicio de cálculo de CI de este manuscrito. En el año 2019, el mencionado centro de Competencia en Indicadores Compuestos y Marcadores (grupo K4P) de la Comisión Europea publicó su herramienta **COIN** [449], durante la semana de índices compuestos celebrada el mismo año.

Se trata de un archivo Excel muy completo compuesto por varias hojas y macros que permite introducir (manualmente) información sobre países o entidades a comparar y realizar una selección de los métodos matemáticos a aplicar. A través de sus funcionalidades, se puede observar cómo los datos son procesados, incluyendo un análisis de sensibilidad. Esta hoja Excel se acompañó de un manual de uso para la construcción de índices compuestos robustos. Como principal valor, cabe destacar que esta herramienta está completamente alineada con todos los pasos propuestos por este grupo de trabajo del JRC y, por tanto, incluye todos los pasos documentados en la sección 4.1.2.

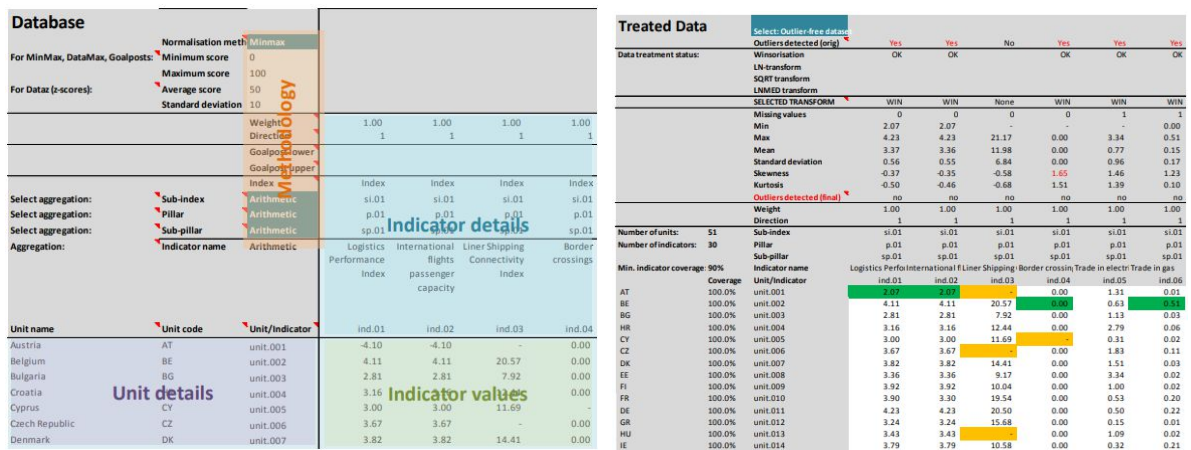


Figura 4.12: Herramienta COIN (Excel). Fuente: [449]

Similarmente a la herramienta CIM, COIN se centra en rankings entre países, pero es un claro ejemplo en el avance hacia la digitalización de todos los procesos de construcción, gestión y validación de un índice compuesto y sus resultados. Es interesante cómo aplican las fórmulas de su metodología utilizando fórmulas de MS Excel e interfaz de interacción. Sin embargo, COIN no procesa los datos en tiempo real ni está preparado para una integración con otras herramientas. Según el candidato, es la herramienta más completa que se ha hallado durante los 3 años de estudio del problema (mención aparte de su versión en R, como se ve más adelante). No obstante, requiere una imputación manual de los métodos, una inclusión manual de la información base sobre la que crear los índices y una navegación impráctica entre sus diferentes pestañas y funcionalidades. Además, no permite la creación de un índice único final desde múltiples indicadores iniciales. Por todo lo anterior, su integración en un sistema automatizado no resulta factible, quedando por tanto excluido su uso en un entorno IoT como el planteado en esta tesis doctoral. Según la opinión del candidato, COIN es la herramienta ideal para instruirse y formarse sobre la construcción de índices compuestos, entendiendo las implicaciones de cada paso y operación, desde el punto de vista lógico y estadístico, pero no es una solución flexible y escalable para su utilización en un entorno dinámico.

Durante el mismo año se publicó la tesis doctoral de Lenka Vrana [450]. El trabajo realizado por esta doctora resultó una gran fuente de consulta y aprendizaje para el candidato. Además, dentro de su análisis y trabajo se incluye la creación y publicación en abierto de una herramienta automatizada para la construcción de índices compuestos a partir de series temporales de datos económicos: el **CIF** [451]. Similarmente a lo planteado en CACIS y en EViews, esta solución aborda el problema de la captura de datos económicos. En este caso, la herramienta permite cargar tanto archivos Excel o CSV como desde una base de datos incluso accediendo directamente a la API de la OCDE seleccionando el país deseado a analizar. El software fue creado en Python y permite la selección de indicadores de la lista de los datos proporcionados por esta entidad económica. Incluye grandes funcionalidades de las expuestas en la sección 4.1.2 tales como la identificación de

puntos de inflexión, diferentes métodos de agregación y visualización. Una de las grandes potencias de esta herramienta es su flexibilidad. Al estar disponible en abierto y en Python (acompañada por útiles cuadernos de Jupyter), puede ser modificada y extendida a medida para el caso de uso requerido.

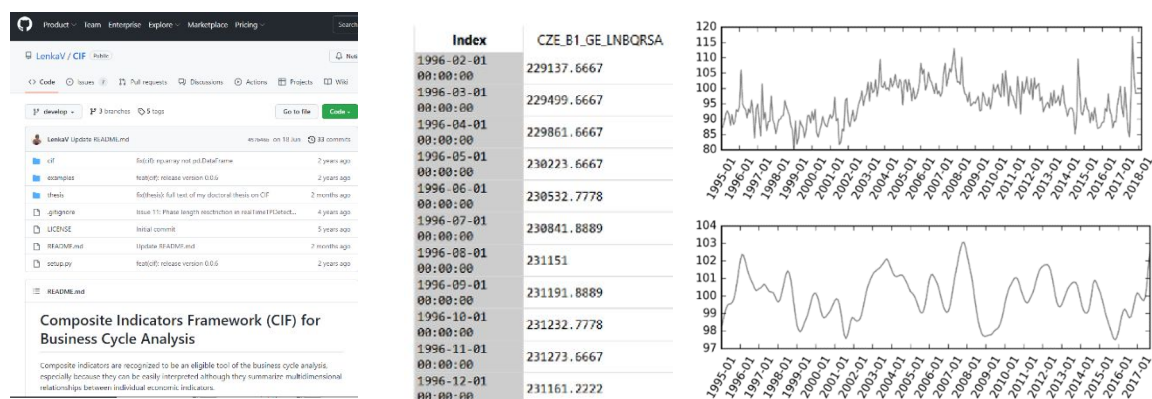


Figura 4.13: Herramienta CIF. Fuente: [451]

Como notas negativas, se trata de una herramienta que no cubre el ciclo completo de aspectos relacionados con la creación de índices, especialmente la selección de parámetros fuera de un *data set* determinado. Tampoco permite la combinación de indicadores a varios niveles para la construcción de un árbol para el cálculo de los mismos. Se trata de una solución que superaba las herramientas existentes de análisis de ciclos económicos, fuertemente marcada por las opciones de estadística computacional que podían elegirse. Su flexibilidad es un punto destacable, lo que podría haberla convertido en una candidata para integrar como un servicio “dockerizado” en la solución de esta tesis. No obstante, este ejercicio no se planteó ya que hubiera supuesto una reformulación de la herramienta, cuyo objetivo inicial era diferente al de este manuscrito. Por último, las opciones de visualización son prácticamente relegadas a la ejecución particular por el usuario de gráficas simples de Python.

Aunque algo más tangencial, otra implementación analizada por el candidato fue el “*Dashboard thinking cities*”, creado en año 2020 y promovido por Telefónica. Esta solución, que propone la utilización de productos de FIWARE en todo el procedimiento [452], lo cual coincide en gran medida con la propuesta de esta tesis doctoral. Esta solución fue cuidadosamente estudiada teniendo en cuenta que (i) se basa en la utilización de FIWARE Context Broker (ORION) como gestor de contexto en entorno IoT y (ii) otros componentes que se consideraron para ser incluidos en este trabajo, (iii) consta de módulo software llamado explícitamente “KPIs” diseñado para crear Indicadores Clave de Rendimiento para *Smart Cities* y (iv) permite un enfoque multidimensional multiservicio construido sobre datos abiertos y tecnologías de código abierto. Es una solución orientada a cubrir las necesidades de creación de parámetros para evaluar el rendimiento de una ciudad en diferentes áreas para construir políticas sobre los mismos (de nuevo, coincidiendo con el espíritu de esta tesis doctoral).



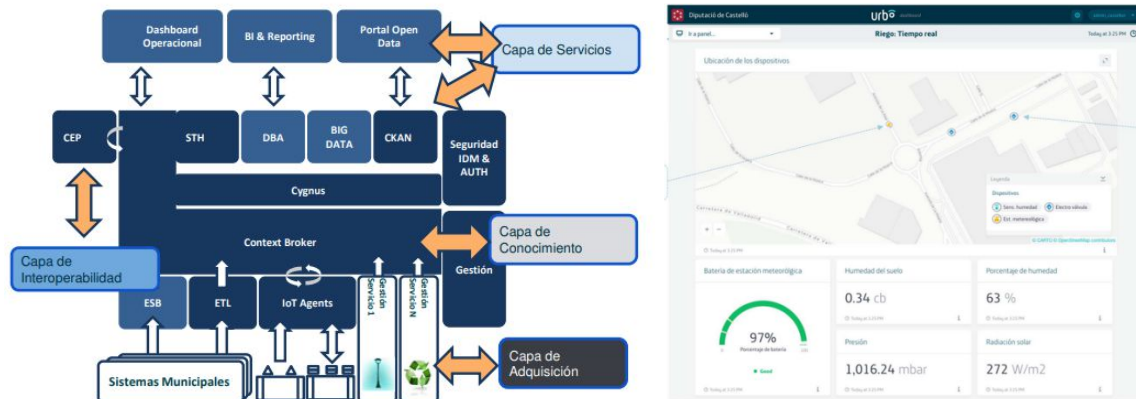


Figura 4.14: Plataforma y dashboard Thinking Cities. Fuente: [452]

Sin embargo, se descartó al no encontrar ninguna aplicación real relevante del concepto, pareciendo por el momento una propuesta interesante pendiente de validar en casos prácticos. Además, no contemplaba la inclusión de opciones para adaptar la estructura del cálculo de CIs, no incluye opciones ajustadas a las operaciones matemáticas indicadas en la sección 4.1.2.2 ni describe si es posible realizar predicciones sobre dichos indicadores.

Por último, en los momentos de finalización de esta tesis doctoral (julio 2022), el mencionado grupo dentro del JRC de la Comisión Europea ha publicado (tras su anuncio en la semana de los indicadores compuestos en diciembre de 2021) la versión en software ejecutable (en R) de la herramienta COIN [453].

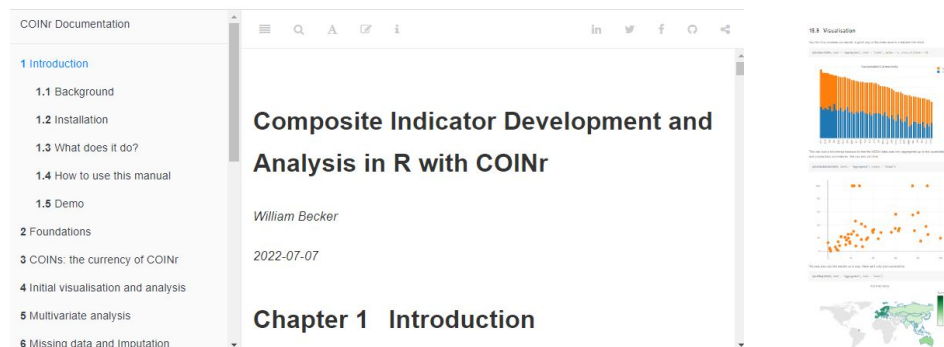


Figura 4.15: Herramienta COIN-R. Fuente: [452]

Según el candidato, esta publicación es un gran avance hacia la automatización de estos cálculos, que se encuentra alineado con la propuesta de esta tesis doctoral. Esta solución de código abierto abre las puertas a nuevas mejoras e integraciones. Mientras que COIN-R está centrado en la validación y aplicación de todos los métodos matemáticos y estadísticos precisos de la construcción de CIs, el servicio propuesto en la tesis aporta una visión más práctica centrada en la integración con datos en tiempo real proveniente de fuentes heterogéneas.

Diferenciándose de todo lo expuesto anteriormente, el objetivo principal de esta tesis, en este sentido, es diseñar un sistema que aproveche los datos y las técnicas del IoT para representar un fenómeno complejo mediante indicadores compuestos. Se trata de crear una herramienta para el análisis retrospectivo de dicho fenómeno, pero, aprovechando su base de Internet de las Cosas, poder realizar una monitorización en tiempo real. Una de las principales propuestas de valor del CIC – *Composite Index Calculator* (así se ha denominado el servicio desarrollado en la tesis) es que se basa en información medida directamente de las actividades/impactos de una entidad sobre un fenómeno particular. A diferencia de otros enfoques, el CIC se nutre de información recogida de fuentes de datos heterogéneas. En la Figura 4.16 se exponen las diferencias funcionales y de planteamiento entre la solución propuesta y las herramientas disponibles:

|  | CACIS | Eviews | CIM | Comphind | CLAO | COIN Excel | CIF | COIN-R | CIC |
|--|-------|--------|-----|----------|------|------------|-----|--------|-----|
| Flexibilidad y nivel de personalización  | ●     | ●      | ●   | ●        | ●    | ●          | ●   | ●      | ●   |
| Centrado en los ciclos económicos, alimentado por datos de la OCDE o del Banco Mundial           | SI    | NO     | SI  | NO       | NO   | NO         | SI  | NO     | NO  |
| Centrado en la comparación entre países  | NO    | NO     | SI  | NO       | SI   | SI         | SI  | SI     | NO  |
| Posibilidad de utilización por una misma entidad para observar evolución temporal de un fenómeno | NO    | SI     | NO  | NO       | NO   | NO         | SI  | NO     | SI  |
| Captura de datos de fuentes heterogéneas   | ●     | ●      | ●   | ●        | ●    | ●          | ●   | ●      | ●   |
| Cobertura completa de los métodos matemáticos  | ●     | ●      | ●   | ●        | ●    | ●          | ●   | ●      | ●   |
| Cobertura completa de los métodos estadísticos   | ●     | ●      | ●   | ●        | ●    | ●          | ●   | ●      | ●   |
| Visualización atractiva de resultados de CI  | ●     | ●      | ●   | ●        | ●    | ●          | ●   | ●      | ●   |
| Nivel de automatización  | ●     | ●      | ●   | ●        | ●    | ●          | ●   | ●      | ●   |
| Posibilidad de creación de indicadores a medida  | ●     | ●      | ●   | ●        | ●    | ●          | ●   | ●      | ●   |
| Posibilidad de extensión de las funcionalidades para la comunidad científica                     | ●     | ●      | ●   | ●        | ●    | ●          | ●   | ●      | ●   |
| Predicción de índices  | ●     | ●      | ●   | ●        | ●    | ●          | ●   | ●      | ●   |
| Árbol, estructura de varios niveles indicadores hasta el índice final                            | ●     | ●      | ●   | ●        | ●    | ●          | ●   | ●      | ●   |
| Posibilidad de integración con sistema IoT   | ●     | ●      | ●   | ●        | ●    | ●          | ●   | ●      | ●   |

● Funcionalidad no incluida     
 ● Funcionalidad incluida parcialmente     
 ● Funcionalidad incluida

Figura 4.16: Comparación del CIC con herramientas disponibles

## 4.2. Implementación de los servicios

El objetivo principal de la tesis ha sido el diseño y desarrollo de la arquitectura (sección 3) para la ejecución de dos servicios: (1) el cálculo y (2) la predicción de índices compuestos utilizando las ventajas de un sistema de IoT de nueva generación.

Esta subsección trata de evidenciar los planteamientos, diseños y acciones de desarrollo llevadas a cabo para implementar dichos servicios siguiendo la estructura de la arquitectura propuesta.

### 4.2.1. Consideraciones generales

Acorde a los requisitos definidos, el cálculo la arquitectura debe demostrar su capacidad para permitir entrenamiento e inferencia de modelos de *machine learning* partiendo de la base de un cálculo de indicador compuesto. Además, se especificó que estos servicios deben de ser capaces de aceptar diferentes orígenes de la información (fuentes de datos) y proveer de flexibilidad en su cálculo (número de datos necesarios, indicadores iniciales variados, etc.).

Para satisfacer estos requisitos, y como antesala del detalle de su funcionamiento (e implementación), dos consideraciones generales resultaron necesarias para complementar su entendimiento.

1. Debido a su marcada orientación a servicios y la voluntad de separación de funciones (microservicios), se permitió que tanto el cálculo como la predicción de los índices fuera llevada a cabo mediante varios servicios cada una. Así, mientras que la funcionalidad principal compondría un único servicio, podrían hacer uso de otros componentes (contenedores Docker) para algunas acciones auxiliares. De esta forma, además de validar la arquitectura, se proporciona de una mayor flexibilidad ante la evolución de dichos servicios y su posibilidad de incorporar funcionalidades externas sin necesidad de modificación de los programas.
2. Tanto el servicio de cálculo del CI como de predicción de índices requieren una preparación de los datos. Debido a que son herramientas diseñadas para ser introducidas de forma aislada y sustituible, su programación ha sido realizada de la forma más genérica posible; esto es: asumiendo un modelo de datos de entrada invariable y proporcionando su funcionalidad en base a dicho acuerdo.

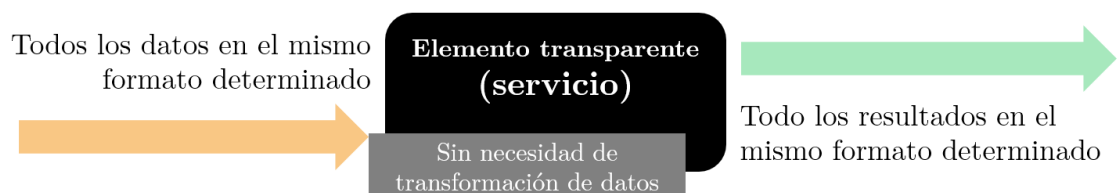


Figura 4.17: Acuerdo de datos de entrada y salida

Este hecho se ha introducido en la sección 3.3.1, donde se ha explicado que la conversión de los datos originales a dicho modelo de datos de entrada acordado es, de hecho, una atribución de los agentes de preprocesamiento. No obstante, sería un error pensar que estos dos puntos (agentes y servicio) no están conectados. En realidad, aunque existe separación (tecnológica) entre ambos módulos, el desarrollo de uno (agentes) debe realizarse considerando el otro (servicio). De manera análoga a lo que se expuso en la sección 3.3.2 ilustrado en la Figura 3.21: Agentes instalados funcionando según indicaciones, una relación 1:1 entre agente y servicio no sería óptima ni escalable. Por tanto, esta “conexión” entre ambas partes se tradujo en el acuerdo sobre un modelo de datos útil para la ejecución del servicio y factible por parte de los agentes, encargados de transformar la información a dicho esquema previa inserción en el sistema propuesto. Por tanto, podría decirse que el servicio está compuesto por dos elementos, aunque uno de ellos es delegado en los agentes de adquisición de datos. El elemento unificador es el esquema definido.

De forma simplificada, los datos brutos se preprocesan (*data curation*) se convierten a un formato común acordado a través de los agentes de interoperabilidad, se almacenan y luego se recuperan de los cálculos matemáticos del servicio de CIs para obtener la métrica única final (cálculo o predicción). En los escenarios de validación (sección 5), el desarrollo de los agentes depende de los datos a capturar y el esquema de datos particular acordado.

Esta estructura fue decidida por el candidato para permitir la flexibilidad de cambiar/corregir/adaptar/añadir más fuentes de datos en una parte restringida de la arquitectura (agentes), sin afectar a la estructura del servicio, dejando el cálculo desde los datos de entrada hasta el indicador compuesto (y los subíndices) como un elemento fijo (contenedor) a lo largo del tiempo. Además, de esta manera, el cálculo más exigente (a nivel cantidad, frecuencia de ejecución, etc.) podría ser soportado conjuntamente por diferentes equipos (aproximación *edge computing*).

Por tanto, la primera consideración antes de la descripción de los servicios es elaborar el modelo de datos que fue escogido en estos acuerdos. Esta acción ya fue esbozada en la sección 3.3.1. El modelo de partida elegido fue *KeyPerformanceIndicator* [454] de FIWARE. Los motivos fueron:

- El servicio de cálculo de indicadores compuestos (como se ha expuesto en las secciones inmediatamente anteriores) aprovecha la definición de múltiples indicadores individuales iniciales para convertirlos, tras varios procesos, en un valor único. Como su propio nombre indica, estas entidades iniciales pueden ser identificadas como “indicadores” o KPIs. Por tanto, su expresión usando un modelo de datos que se ha diseñado con esta intención pareció apropiado.
- La flexibilidad del modelo y su cumplimiento no el estándar NGSI.
- La cercanía del doctorando a los modelos de datos FIWARE, así como a los componentes tecnológicos (*enablers*) de la iniciativa, que han compuesto gran parte de la arquitectura propuesta.

Por tanto, se procedió a extender este modelo de datos para definir el propio de la tesis doctoral: *ComposedKeyPerformanceIndicator*. El esquema se muestra en la Figura 4.18: Esquema del modelo de datos acordado de entrada (desde agentes), mientras que los campos son explicados en la siguiente tabla:

**Tabla 4.5:** Campos establecidos en el modelo de datos acordado *cKPI*

| Campo/atributo              | Significado  | Relevancia para el servicio  |
|-----------------------------|--|--|
| <i>id, type</i>             | Identificación del indicador y su tipo.              | Campos requeridos por el estándar NGSI.  |
| <i>properties</i>           | Modelo de datos                                      | Invariable, se informa del modelo de datos <i>cKPI</i> .   |
| <i>location</i>             | Localización del sensor.                             | No tiene mucho valor para el servicio. Es útil en caso de representación en un mapa de las fuentes de datos.   |
| <i>source</i>               | Descripción de la fuente según significado original. | Se utilizó este campo para indicar qué tipo de indicador se trata en relación a su profundidad en el árbol: en caso de provenir directamente de una fuente de datos: <i>ckpi-input</i> .                 |
| <i>name</i>                 | Nombre y descripción del indicador.                  | Descriptivo. Útil en el servicio para la propagación de información  |
| <i>description</i>          |  |  |
| <i>category</i>             | Categoría de entidad de indicador.                   | Este atributo trata de expresar de qué tipo era la fuente original del indicador: cuantitativa o cualitativa (información desde informes o literatura que ha sido convertida a número dentro del agente) |
| <i>organization</i>         | Nombre de la organización.                           | Útil en multi-tenancia. No es el caso del servicio ni arquitectura escogidas así que no es un valor relevante. Meramente informativo con el nombre de la entidad adoptante.                              |
| <i>process</i>              | Proceso de cálculo del indicador.                    | Para los servicios de la tesis, este atributo informa (de forma explicativa) sobre las fuentes utilizadas para generar el indicador.   |
| <i>calculationFrequency</i> | Frecuencia con la que se calcula el valor.           | Muy relevante para el servicio, a la hora de realizar la normalización y   |

|                            |  |   |
|----------------------------|--|---|
|                            |  | para inspeccionar la frecuencia más restrictiva.  |
| <i>calculationMethod</i>   | Método de cálculo del indicador.                       | En el servicio, usado para indicar si se trata de un agente automático (hacia el gestor de contexto, etc.) o manual (introducción de datos desde formulario o carga <i>batch</i> ). |
| <i>calculationPeriod</i>   | Periodo cubierto por el valor del indicador (from-to). | Muy útil para el servicio, tanto para normalización como para el cálculo del CI. Formato: <i>YYYY-MM-DD</i>   |
| <i>dateObservedFrom</i>    | Detalle del periodo cubierto.                          | Mismo significado que el anterior pero con formato detallado hasta el milisegundo. Útil para escenarios cercanos a tiempo real.   |
| <i>dateObservedTo</i>      |  |   |
| <i>kpiValue</i>            | Valor del indicador.                                   | Valor principal que se aporta de cara al servicio.  |
| <i>dateNextCalculation</i> | Siguiente fecha en que se recibirá un valor.           | Útil para el servicio en temas de filtrado. Formato: UTC ISO-8601. “YYYY-MM-DDTHH:mm:ss.sTZD”   |
| <i>dateModified</i>        | Fecha en que se ha realizado el registro.              |   |
| <i>ciccategory</i>         | Categoría de cara al servicio de CI.                   | Familia a la que pertenece (indicador de nivel 2).  |
| <i>ciclevel</i>            | Nivel dentro del árbol.                                | Cuando proviene de agente: “Indicator”.   |
| <i>sourceCIC</i>           | Categoría de cara al servicio de CI.                   | Subíndice principal al que pertenece (indicador de nivel 1).  |
| <i>unit</i>                | Unidades de medida.                                    | Útil para normalización.  |
| <i>dataPieces</i>          | Información sobre las funetes de datos originales.     | Útil para la contextualización del índice. “ <i>Sensors</i> ”, “ <i>APP</i> ”, “ <i>average-literature</i> ”...   |

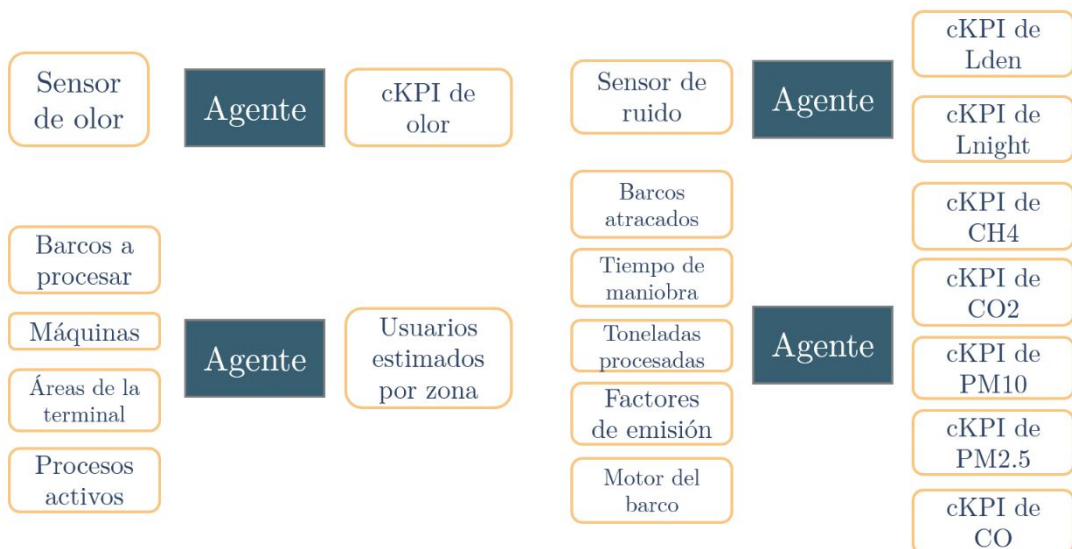
```

{
  "$schema": "http://json-schema.org/schema#",
  "$id": "https://[...]/KPI/ComposedKeyPerformanceIndicator/schema.json",
  "title": "Adaptation of the FIWARE - KPI / KeyPerformanceIndicator data model for the purpose of PhD of Ignacio Lacalle",
  "description": "ComposedKeyPerformanceIndicator",
  "type": "object",
  "allOf": [
    {
      "properties": {
        "type": {
          "type": "string",
          "enum": [
            "ComposedKeyPerformanceIndicator"
          ],
          "description": "NGSI Entity type"
        },
        "location": {
          "$ref": "http://geojson.org/schema/Geometry.json#",
          "description": "The geo:json location of the data"
        },
        "source": {
          "type": "string",
          "description": "The ID of the data source"
        },
        "name": {
          "type": "string"
        },
        "description": {
          "type": "string"
        },
        "category": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "organization": {
          "type": "object",
          "properties": {
            "name": {
              "type": "string"
            }
          }
        }
      },
      "required": [
        "id",
        "type",
        "name",
        "category",
        "calculationFrequency",
        "calculationMethod",
        "calculationPeriod",
        "kpiValue",
        "dateNextCalculation",
        "dateModified",
        "source",
        "ciccategory",
        "ciclevel",
        "sourceCIC",
        "unit",
        "dataPieces"
      ]
    },
    {
      "process": {
        "type": "string"
      },
      "calculationFrequency": {
        "type": "string"
      },
      "calculationMethod": {
        "type": "string"
      },
      "dateObservedFrom": {
        "type": "string",
        "format": "date-time"
      },
      "dateObservedTo": {
        "type": "string",
        "format": "date-time"
      },
      "kpiValue": {
        "type": "number"
      },
      "dateNextCalculation": {
        "type": "string",
        "format": "date-time"
      },
      "dateModified": {
        "type": "string",
        "format": "date-time"
      },
      "peicategory": {
        "type": "string"
      },
      "peilevel": {
        "type": "string"
      },
      "sourcePort": {
        "type": "string"
      },
      "unit": {
        "type": "string"
      },
      "dataPieces": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "name": {
              "type": "string"
            },
            "retrievalWay": {
              "type": "string"
            }
          }
        }
      }
    }
  ]
}

```

Figura 4.18: Esquema del modelo de datos acordado de entrada (desde agentes)

El trabajo de interpretación, conversión y preparación para el servicio por parte de los agentes constituye la base del cálculo del CI. Es conveniente mencionar que el desarrollo de los agentes NGSi es una acción que divergirá en gran medida entre un escenario de aplicación y otro. Como los datos serán diferentes, el tratamiento a este nivel variará y será necesario un análisis cuidadoso y a medida para plantear el desarrollo de los agentes. Se tratará de la parte más personalizada y costosa a la hora de desplegar la arquitectura y el servicio planteado, ya que dependiendo del número de datos necesarios para alimentar los indicadores esperados, estos desarrollos llevarán más o menos tiempo. Además, la práctica ha enseñado que se trata de procesos que requieren una razonable cantidad de test e iteraciones para conseguir popular todos los campos necesarios en el esquema. Pueden ir desde una simple conversión de formato hasta una compleja relación, combinación y construcción de datos. Además, cada agente puede tener un número diferente de entradas y salidas. A pesar de que el caso habitual es realizar una relación 1:1 (una fuente de datos genera un indicador), es también muy común el caso N:1 y N:N, siendo 1:N el menos frecuente. La arquitectura y el diseño del servicio se han preparado para aceptar cualquier casuística en este sentido. Se expone un ejemplo de cada tipo en la siguiente figura (basado en los escenarios de validación de la tesis – sección 5).



**Figura 4.19:** Variabilidad en relaciones número de entradas:salidas en agentes

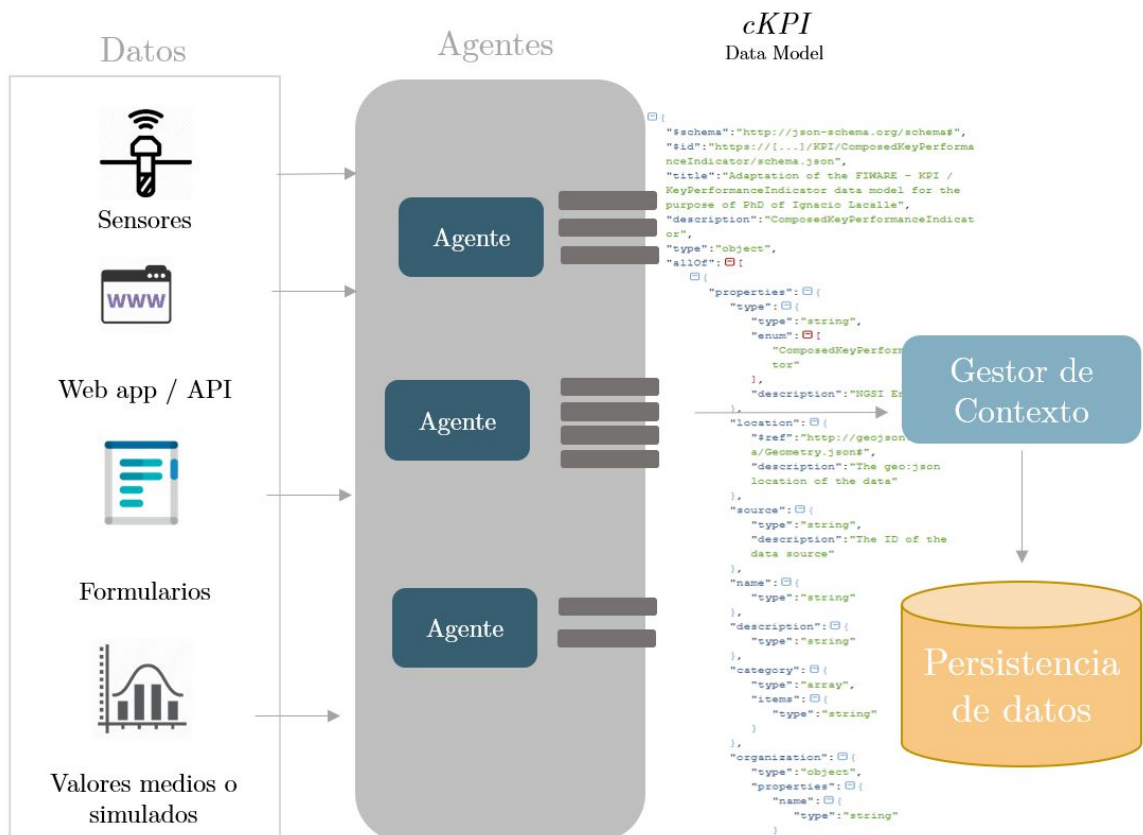
Será responsabilidad de cada entidad adoptante el diseño y despliegue de estos agentes. A pesar de no entrañar una dificultad técnica desmedida, los desarrolladores deberán conocer el servicio de cálculo de CI para conseguir una integración adecuada. Este punto es también de alto interés de cara a una explotación de la solución (sección 6) ya que este servicio de desarrollo podría ser ofrecido por la entidad comercializadora del producto como canal de ingresos adicional. Por tanto, estos agentes ejercerán una función clave en la ejecución de los servicios. Para complementar lo expuesto en la sección 3.3.1, desde el punto de



vista del cálculo y predicción de índices compuestos, las funciones de un agente deben ser las siguientes:

- Recuperar los datos de la fuente original. Esta conexión puede ser: a) el agente consulta activamente el origen de la fuente de datos o b) el agente implementa un servicio de *broker* de datos para que la entidad de origen de datos activa apunte al agente publicando nuevos datos.
- Realizar un filtrado, primera agregación e interpretación (hacia el servicio) de los datos. Esto incluye obviar partes irrelevantes de los datos e identificar posibles valores espurios o valores atípicos.
- Contextualizar los datos: proceso (automático, manual), tipos de fuentes de datos, información operativa sobre su funcionamiento (frecuencia, periodicidad de refresco, última actualización...).
- Construcción de la entidad para actualizar el valor en el gestor de contexto siguiendo el modelo de datos acordado.
- En perspectiva de servicios más complejos orientados a *edge computing*, estos agentes podrían también alojar funciones de predicción, entrenamiento, almacenamiento temporal de información de indicadores iniciales, etc.

En la siguiente figura se resume el papel de estos agentes incluyendo el formato de envío de la información hacia el sistema.



**Figura 4.20:** Resumen de consideraciones en agentes para la ejecución del servicio

## 4.2.2. Cálculo de indicadores compuestos

Esta subsección relata el componente central de la tesis, donde el resto de secciones han convergido para dotar de un valor añadido al usuario: el servicio de cálculo de indicadores compuestos. La arquitectura propuesta ha sido orientada para facilitar este servicio, y el resto de desarrollos expuestos en esta sección aporta un valor sobre la base que éste ofrece.

En esencia, se trata de un software que recoge información disponible en una base de datos - que llega con una frecuencia variable y un formato de datos (fijo) determinado-, la procesa y provee un valor numérico que condensa la información de un fenómeno real en una escala de 0 a 1. Conceptualmente, debe recopilarse una serie de KPIs (indicadores) iniciales (numéricos – *kpiValues*), aplicar sobre ellos las operaciones matemáticas estudiadas en esta sección y devolver una estructura de valores por indicador anidada y jerárquica para poder mostrarlos al usuario.

En los párrafos siguientes (y en la Figura 4.21: Mecanismos de extracción e inserción de datos por/desde el servicio) se expone la relación de este servicio con los módulos de la arquitectura, especialmente con el componente de almacenamiento de información.

Los indicadores son enviados desde agentes conectores (a través -o no- de un gestor de contexto) a un elemento de persistencia (base de datos) de donde extraerá la información el servicio y donde deberá depositar, igualmente los resultados. Durante el diseño del servicio, además de acordar el modelo de datos que se espera por parte del mismo, convino concretar mecanismos específicos para realizar las consultas e inserciones adecuadas. Los mecanismos establecidos para que el servicio funcione correctamente son los siguientes:

- El proveedor de servicio almacena los *cKPIs* provenientes de los agentes, con formato *ComposedKeyPerformanceIndicator* en el índice (o colección en el caso de MongoDB): “*ckpi-input*”.
- El usuario introduce una serie de configuraciones (métodos matemáticos, estructura de los índices, etc.) a través de la GUI del módulo de visualización. La lógica de interfaces (a través del API) almacena dichas configuraciones en un formato determinado en el índice (o colección *ckpi-config*). El desarrollador/instalador informa de otro tipo de configuraciones (técnicas) mediante otros espacios en la misma interfaz. La información se almacena en el mismo índice.
- El servicio se ejecuta y almacena los resultados (formato *cKPI*) en el índice/colección *output-ckpi*. Estos resultados son utilizados por otros servicios para proveer de utilidades auxiliares/adicionales (se explican más adelante en esta sección).

De este modo, en vez de establecer una gran cantidad de índices para cada

envío de información, todos los valores de indicadores (*cKPIs*) son almacenados en el mismo índice, pudiendo utilizarse para filtrarlos los valores dentro de su modelo de datos *dateObservedTo* y *dateObservedFrom*. Mientras que en el índice de entradas (*input*) solo existirán indicadores iniciales, en el índice de salida (*output*) existirá un *cKPI* para cada elemento del árbol de cálculo (subíndices, índice final...).

La siguiente figura ilustra los mecanismos explicados:

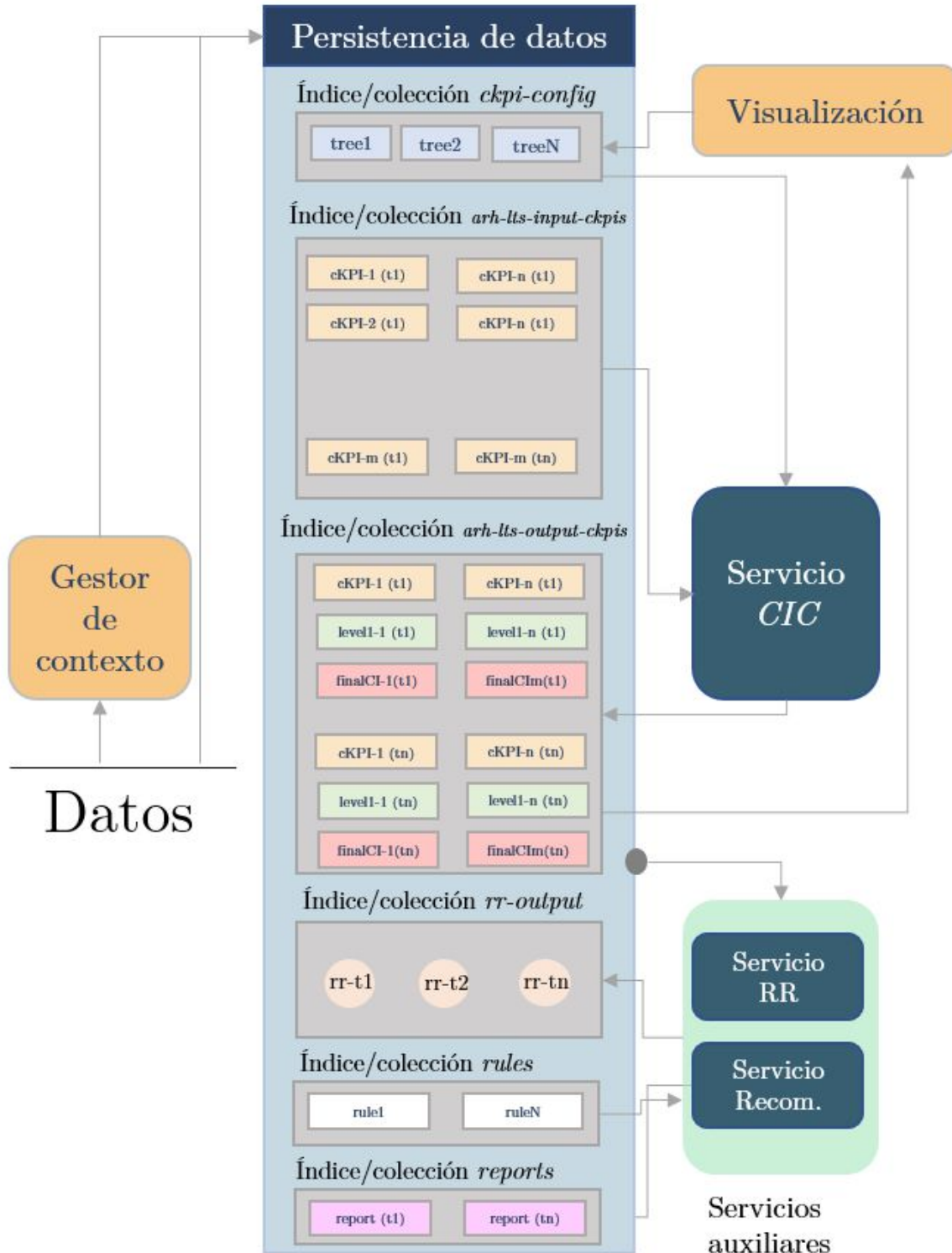


Figura 4.21: Mecanismos de extracción e inserción de datos por/desde el servicio

Por un lado, el usuario final deberá realizar las configuraciones propias de las características del servicio. En las subsecciones de 4.1.2 se han expuesto detalladamente las operaciones matemáticas a incluir en el servicio y los diferentes métodos que pueden utilizarse. Como se desea crear una herramienta genérica para la aplicación de este servicio en varios escenarios, el usuario debe poder seleccionar entre dichos métodos.

Para ello, desde el módulo de visualización se incluye una interfaz de interacción donde el usuario final puede escoger los diferentes métodos de:

- **Normalización:** Tras el estudio realizado en secciones anteriores, se acordó ofrecer la posibilidad de elegir entre: (i) distancia a un valor de referencia, (ii) re-escalado – *min-max* y (iii) estandarización – *z-scores*.
- **Ponderación:** Se permite la selección entre (i) igualitario, (ii) manual y (iii) por asignación de presupuesto.
- **Estrategia de actualización:** (i) preguntar cada vez o (ii) último valor.
- **Imputación de datos ausentes:** (i) *hot deck*, (ii) *cold deck* o (iii) media incondicional.

Además, se le permite al usuario personalizar la estructura del árbol de cálculo de índices (nodos). Esta es una de las funcionalidades de escalabilidad y flexibilidad más interesantes del servicio. A través de la GUI, el usuario puede configurar el árbol, bien desde cero (véase la siguiente figura), indicando para cada nodo sus predecesores, valor de ponderación (si manual) y método de agregación o bien a partir de un esquema precargado (configurable en cada despliegue).

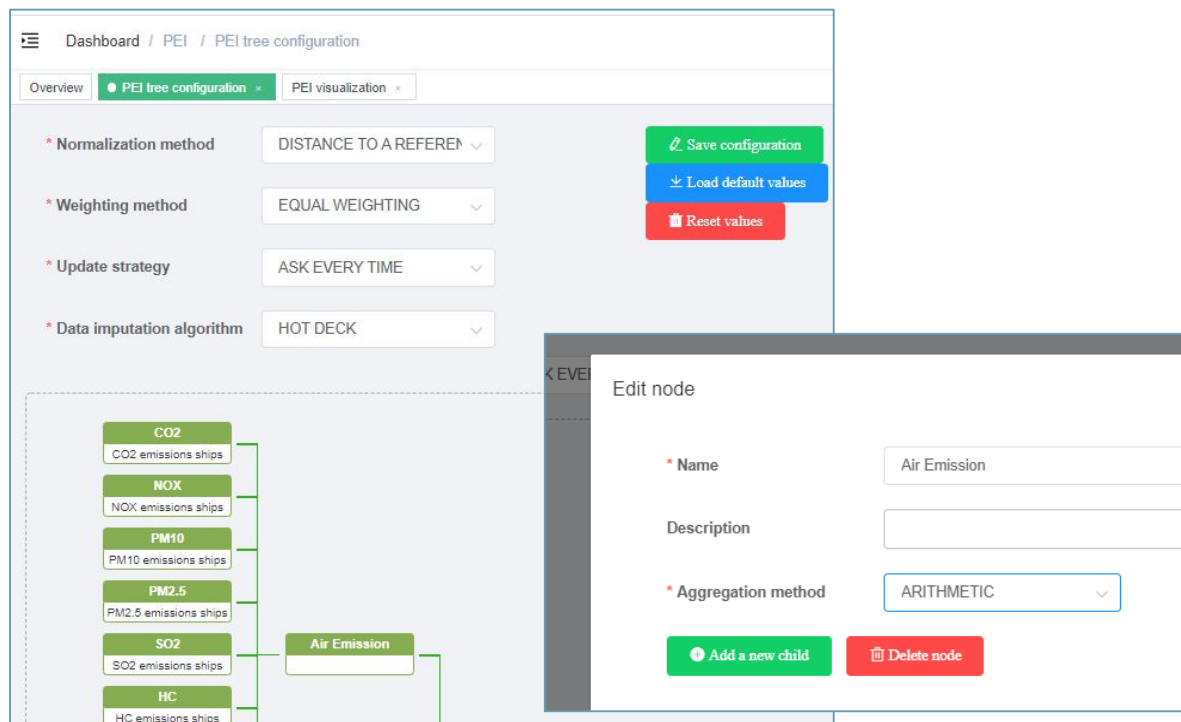


Figura 4.22: Personalización estructura del árbol cálculo del CI

Por otro lado, el servicio requiere de una información técnica que no puede esperar ser completada por parte de un usuario final. Esta información se deriva directamente de la naturaleza de la arquitectura (sección 3.2.1). La flexibilidad del sistema impone que un servicio sea considerado una pieza aislada del sistema, pudiendo ser sustituida y reemplazada por otra. Como tal, no puede conocer (internamente, dentro de su código) ninguna información sobre el resto de la arquitectura. En este caso, este hecho se traduce en la imposibilidad de incluir la dirección del API del módulo de almacenamiento dentro de la lógica del servicio. Por tanto, ha de crearse un método alternativo para que el servicio pueda conocer cuál es dicha dirección, para introducir dentro de sus consultas (código) a la base de datos, dicha URL del API donde realizar las consultas.

Para ello, se establece un espacio dentro del módulo de visualización a través del cual se proporciona la información de dirección, puerto e índice o colección de la base de datos que contiene (véase Figura 4.23: Espacio de configuración de datos técnicos para ejecutar el servicio):

- Los datos de configuración del usuario (página anterior).
- Los indicadores iniciales *cKPIs*.
- Los datos para normalización (en caso de requerir instancia externa).
- Dónde se debe depositar el resultado del cálculo del servicio.
- Dónde se debe depositar el resultado de cara a la normalización (en caso de requerir instancia externa).

```

1  [
2  {
3    "input": [
4      {
5        "element": "tree",
6        "URL": "http://172.1.18.65:9200",
7        "index": "arh-lts-tree",
8        "collection":""
9      },
10     {
11      "element": "indicators",
12      "URL": "http://172.1.18.65:9200",
13      "index": "arh-lts-input-kpis",
14      "collection":""
15     },
16     {
17      "element": "normalisation-values",
18      "URL": "http://172.1.18.65:9200",
19      "index": "arh-lts-normalisation",
20      "collection":""
21     }
22   ],
23   "output": [
24     {
25       "element": "calculated-indicators",
26       "URL": "http://172.1.18.65:9200",
27       "index": "arh-lts-output-kpis",
28       "collection":""
29     },
30     {
31       "element": "normalisation-sink",
32       "URL": "http://172.1.18.65:9200",
33       "index": "arh-lts-norm-sink",
34       "collection":""
35     }
36   ]
37 }
38 ]

```

**Figura 4.23:** Espacio de configuración de datos técnicos para ejecutar el servicio

Una vez establecidas las configuraciones técnicas y de usuario, puede ejecutarse el servicio acorde a las instrucciones de la arquitectura (sección 3.4.2). En la misma se especifica que la instalación de los servicios debe materializarse en la indicación del comando de ejecución de contenedor Docker (**docker run**) dentro del archivo *cron* (**crontab -e**).

Por tanto, el desarrollador/instalador procede con este paso para la instalación. En este punto cabe destacar que las configuraciones establecidas en las páginas anteriores residen en un índice específico de la base de datos de persistencia de la arquitectura, que a su vez vive dentro de otro contenedor (arquitectura de microservicios) que el servicio, inicialmente, no conoce.

Con la intención de que sean visibles y el servicio pueda alcanzar estas configuraciones para desarrollar su funcionalidad, debe instalarse como sigue:

```
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
* * * * 0 sudo docker run cic_service -e DB_URL='http://172.1.18.65:9200' DB_INDEX="ckpi-config"
```

**Figura 4.24:** Instalación del servicio por comando

- Paso por parámetro de entorno de la dirección donde se encuentran los parámetros de configuración. Esto puede hacerse de dos maneras: (i) que esta configuración se encuentre en el sistema de archivos del equipo (*host*), y establecer un volumen Docker para que el contenedor obtenga una copia de la información del directorio correspondiente o (ii) indicar por parámetro la URL del sistema de persistencia. Se recomienda la utilización del segundo método (más alineado con la lógica de la arquitectura de separación modular). Otros aspectos como las contraseñas de acceso para la base de datos pueden incluirse también como parámetros en esta llamada o ser acordadas previamente (por ejemplo, usuario por defecto “*elastic*” y contraseña conocida)
- Las direcciones IP indicadas en la figura Figura 4.23: Espacio de configuración de datos técnicos para ejecutar el servicio se corresponden con la IP fija del contenedor de la BBDD (puede haber varias BBDD por tanto varias IPs fijas). Esto se consigue asegurando que tanto el servicio como la BBDD están dentro de la misma red de Docker.

Una vez el servicio ha sido instalado de esta forma, se ejecutará según la periodicidad marcada.

En las siguientes páginas se expone la composición interna del servicio en sí, explicando su funcionamiento y ciertos detalles de su desarrollo. Por último, se proporciona información técnica sobre el programa creado y se incluyen una serie de diagramas de secuencia para ilustrar el orden de interacción con el resto de módulos de la arquitectura.



Con respecto a la estructura del servicio, el procesamiento a realizar consiste en una serie de cálculos invariables (servicio reutilizable en diversos escenarios), siguiendo un flujo determinado. Estos cálculos son transparentes para los actores que intervienen en la definición y uso del índice compuesto, resultando en un elemento transparente que toma unas entradas, las procesa y proporciona una salida (Figura 4.17: Acuerdo de datos de entrada y salida).

El candidato decidió abordar el CIC (*Composite Index Calculator*) a partir de un flujo de reducción de indicadores en cascada descendente (véase la siguiente figura). Por lo tanto, se adoptó el enfoque de considerar cada pieza de datos de entrada (*cKPI*) como las hojas de un árbol, que van juntándose hacia la copa, constituyendo una red de nodos cuyo valor se calcula iterativamente.

Para evolucionar en el cálculo de dichos valores, se realizan las operaciones matemáticas correspondientes (normalización, ponderación, agregación) según el nivel de profundidad del árbol en que se encuentre el algoritmo. Además, todo lo anterior se encuentra gobernado por los parámetros de configuración mencionados.

La siguiente figura ilustra la esencia del servicio:

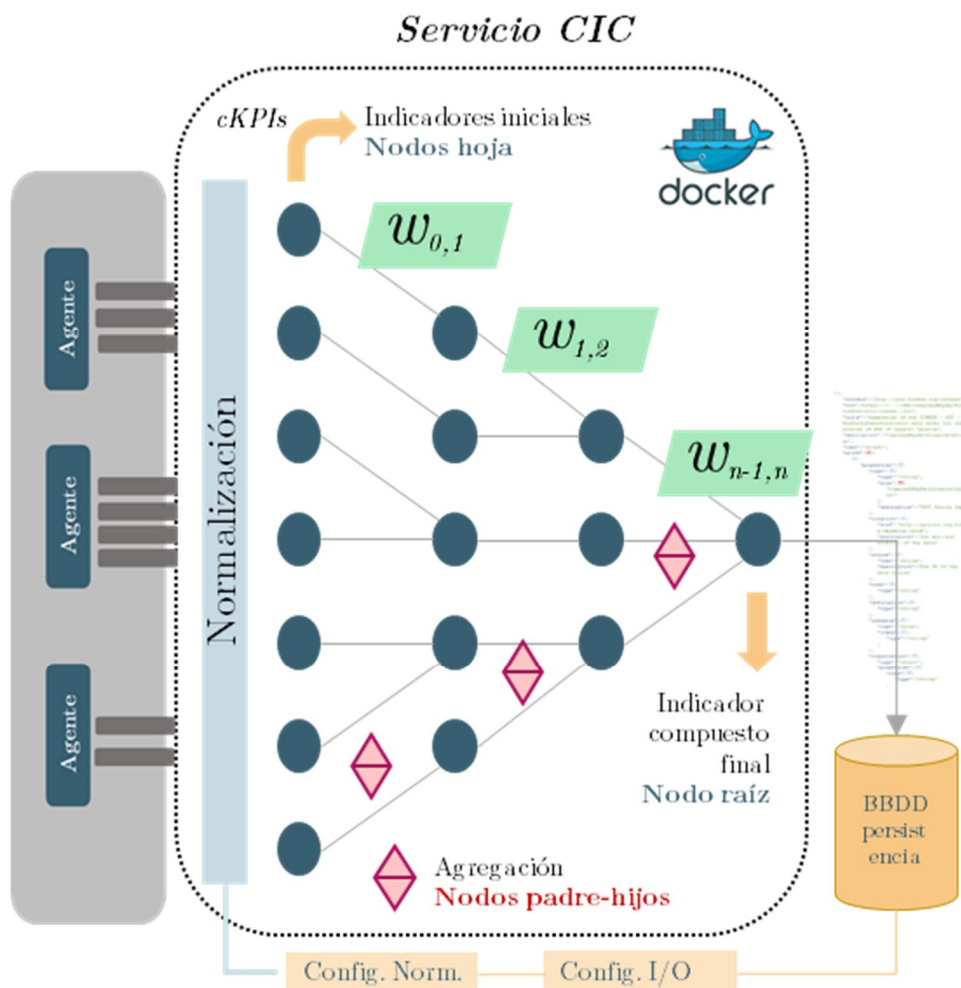


Figura 4.25: Estructura del servicio de cálculo de indicadores compuestos (CIC)

De acuerdo con este marco diseñado (véase la figura anterior), el servicio CIC realiza un procedimiento de operación de 8 pasos serializados:

### **Paso 1: Lectura de los valores de configuración**

A partir de los valores pasados por parámetro, el servicio obtiene información sobre la URL de configuración donde se alojan los datos sobre dónde recuperar la estructura del árbol a calcular, de dónde extraer los *cKPIs* iniciales y dónde enviar los resultados una vez calculados.

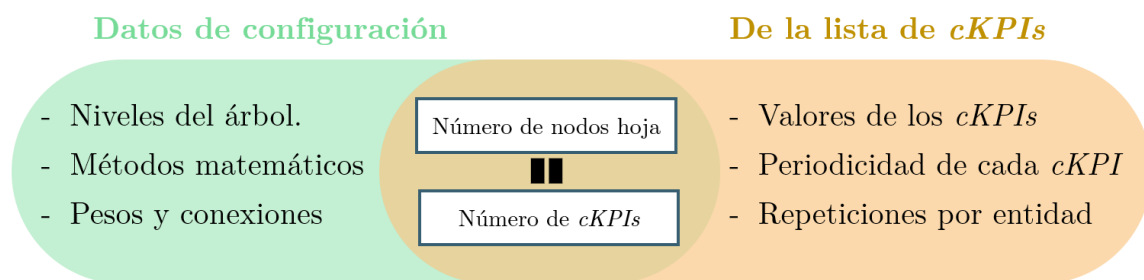
Esta información se transforma en las expresiones vectoriales/matriciales del lenguaje de programación escogido para su utilización a lo largo del programa.

Asimismo, es en este momento cuando se procesa (si existe) la información sobre el servidor de normalización donde consultar (en caso de requerirse una consulta externa o de una fuente diferente). Esto suele estar informado cuando la opción escogida ha sido normalización por referencia (como se explica posteriormente).

### **Paso 2: Validación de la estructura del árbol**

Se trata de un paso de comprobación, previo al cual el servicio no puede ser ejecutado. El objetivo es comprobar que la configuración leída de los parámetros coincide con los datos provenientes de fuentes de datos en tiempo real. Es decir, que el árbol diseñado con la herramienta gráfica de la interfaz (Figura 4.22: ) es coherente con los agentes de datos desarrollados que están enviando información al módulo de persistencia.

Si esta comprobación no es correcta, el servicio se detiene y no procesa ningún resultado.



**Figura 4.26:** Comprobación de la estructura de nodos e indicadores

### **Paso 3: Construcción de los elementos programáticos de la estructura del árbol**

Con la información recuperada de la estructura del árbol, el programa utiliza clases propias (Node y Edge – explicadas más adelante), para construir el árbol. Esto se hace siguiendo el enfoque clásico de generación descendente, partiendo del nodo raíz.

Se lee la información de dicho nodo (el indicador compuesto final – véanse ejemplos en los escenarios de validación de la sección 5), se crean los nodos del



siguiente nivel y se guarda la información de los mismos (su identificador) en un tipo de dato vectorial para poder procesar los cálculos iterativos.

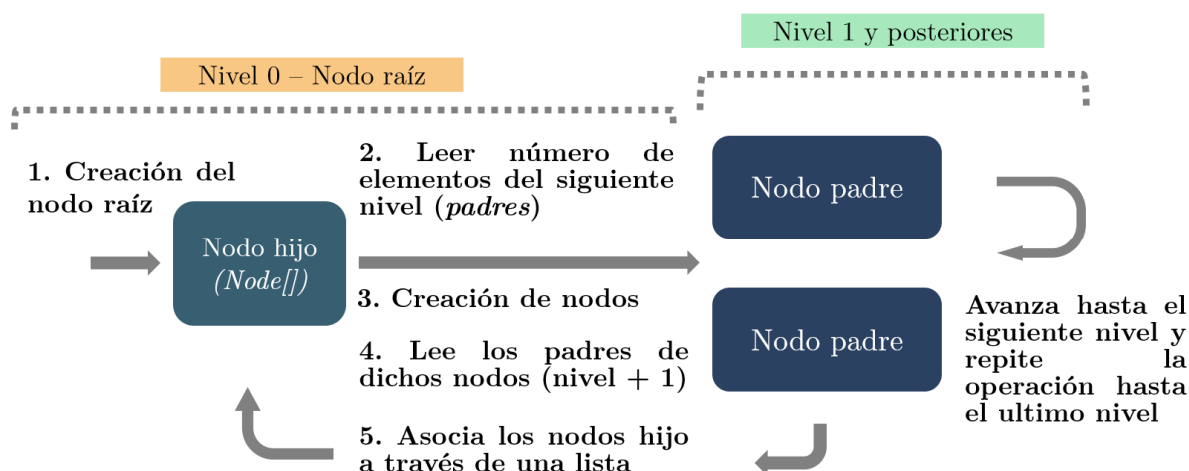


Figura 4.27: Construcción de la estructura del árbol en la lógica del servicio

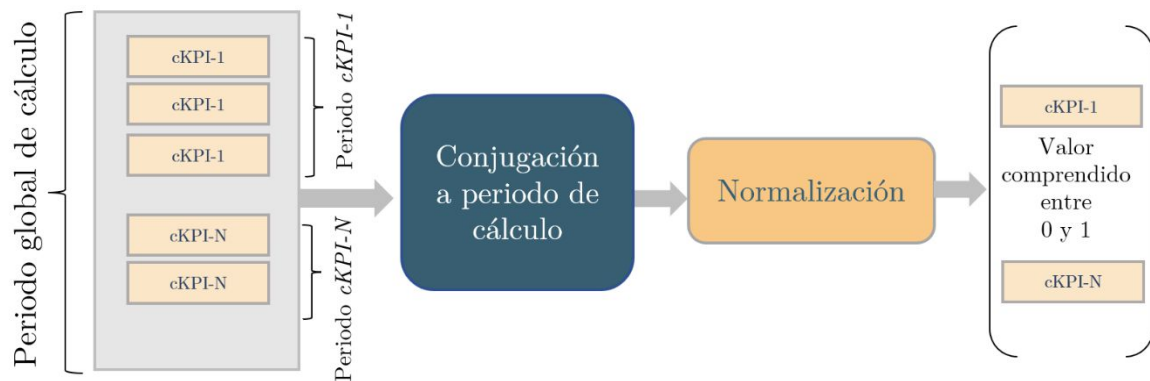
#### Paso 4: Adquisición y adición de los indicadores iniciales (*cKPIs*) del periodo correspondiente

Este paso consiste en consultar el módulo de almacenamiento de datos de la solución y recuperar la información global de *cKPIs* del periodo y en leer y filtrar los JSONs obtenidos. El periodo, en este caso (coincidente con la frecuencia de ejecución del servicio), puede ser obtenido de dos formas: (1) automáticamente por el servicio interpretando cuál es la periodicidad más restrictiva de todos los *cKPIs* a considerar (campo *calculationFrequency* en la Tabla 4.5). o (2) a partir de la configuración del usuario final. Este caso no es habitual, pero pudiera ser que quisiera calcularse un CI con menos periodicidad que la más baja posible.

Este paso del proceso recoge esta información, siendo consciente que cada *cKPI* puede tener una frecuencia de actualización diferente, referidas a un determinado periodo de tiempo sobre el que se han calculado (por ejemplo, residuos de plásticos solicitados por los barcos para ser procesados por el puerto durante el mes de abril). A continuación, realiza una adición de aquellos *cKPIs* de la misma entidad que hayan llegado durante dicho periodo. Por ejemplo, si el *cKPI* más restrictivo se actualiza cada semana, el periodo seleccionado para el cálculo será desde el momento de la ejecución hasta una semana anterior (recordatorio que el índice compuesto está orientado a ser un valor de cálculo retrospectivo). Asumiendo esta periodicidad, es de esperar que otros *cKPIs* se actualicen más frecuentemente. De este modo, si un *cKPI* proporcionara un nuevo dato (por ejemplo) cada día, en el periodo de una semana existirían siete (7) registros de dicho *cKPI*. A efectos prácticos, estos registros son JSONs en el índice *ckpi-input* con *kpiValue* diferentes, uno correspondiente a cada día. Esta fase del proceso se dedica a agregarlos a la periodicidad que se va a utilizar para el cálculo de índice compuesto. El dato conjugado de cada *cKPI* se mantiene en los registros de la lógica del servicio.

**Paso 5: Normalización de indicadores iniciales (*cKPIs*)**

El siguiente paso es la normalización de los indicadores iniciales. Como se ha detallado en la sección 4.1, valores de indicadores no pueden agregarse ni combinarse si no están expresado en un marco de unidades y escala común. *Los cKPIs* que se reciben de los agentes conectores de datos vienen expresados en determinadas unidades (*units* en Tabla 4.5) correspondientes a la magnitud que expresan. Por ello, es necesario realizar una conversión a una escala común (elegida: de 0 a 1) en la que expresar todos los *cKPIs* obtenidos. De acuerdo, también, con las reflexiones en dicha sección, se debe aplicar el mismo método de normalización a todos los *cKPIs*.



**Figura 4.28:** Avance iterativo de ponderación y agregación en niveles del árbol

A continuación se explica cómo se aplican los diferentes métodos de normalización según la lógica de la herramienta:

- *Método de re-escalado min-max.* El servicio analiza un histórico (de hasta un año de registros) a través de una consulta con filtro temporal y de mínimos (soportada por la API de Elasticsearch) y obtiene el valor mínimo y máximo de cada *cKPI* en dicho periodo. Se descarta el 5% de los valores superiores e inferiores para evitar valores atípicos [455].
- *Método de estandarización (z-scores):* Similar al funcionamiento anterior. El programa realiza las consultas con la sintaxis adecuada para obtener el valor medio y la desviación estándar de los registros de cada *cKPI* del periodo de un año y, a partir del valor conjugado del periodo bajo estudio, se realiza la normalización.
- *Método de distancia a una referencia:* Este caso requiere una explicación más concreta. Se trata de un método de normalización donde el valor conjugado se contrasta con un valor de referencia (*cKPI* a *cKPI*), calculándose la distancia al mismo. Pueden darse dos escenarios, que están ligados al escenario de despliegue escogido en el caso de utilización de la solución:

A) Escenario que no requiere comparación entre entidades: en este caso, la normalización se hace contra unos valores que están

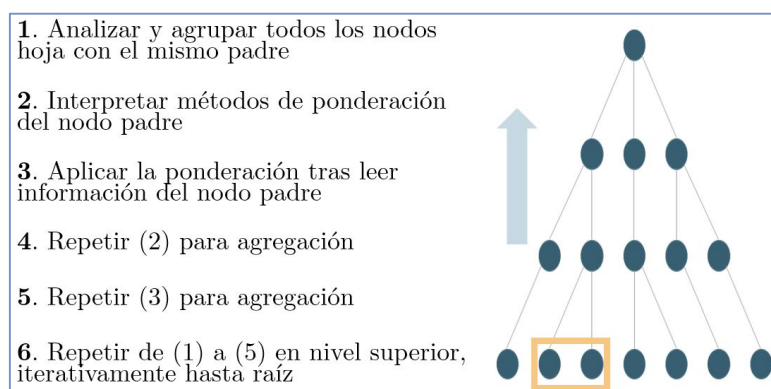
alojados en el mismo espacio de persistencia que se utiliza para el resto de las operaciones (dentro de la solución). Se consulta al índice *norm-ckpi* que posee los datos de referencia de cada *ckPI* en el formato que se expone en la sección 4.2.4.

B) Escenario en que se desea comparar entre varias entidades que ejecutan el servicio y que deben ser contribuyentes a los valores de normalización. Se explica con detenimiento en el apartado de servicios auxiliares; baste decirse que requiere la utilización de un servicio adicional y de una base de datos externa con la que se interactúa de manera anonimizada.

Para finalizar este paso, se realizan las imputaciones de valores ausentes y la gestión de valores atípicos indicada en secciones anteriores.

**Paso 6: Avance iterativo de cálculos en los niveles del árbol, incluyendo ponderación y agregación**

Una vez que se dispone de lo anterior, el siguiente paso es actualizar los valores de cada nodo considerando todas las operaciones necesarias: ponderación y agregación, siempre respetando la dirección de las flechas de las aristas en el árbol y las relaciones de herencia (padres-hijos). La Figura 4.29 resume el proceso. En este sentido, todos los nodos (excepto las hojas) deben tener un método para agregar los valores ponderados de los nodos hijos. En este caso, al contrario que en la normalización, diferentes métodos de agregación pueden utilizarse en diferentes niveles de la jerarquía del árbol (el mismo dentro de un nivel). No obstante, para el cálculo de CI de esta tesis se recomendó utilizar el mismo método para todas las operaciones (como se manifiesta en la inclusión de un único desplegable de agregación y ponderación para selección por parte del usuario). Para llevar a cabo correctamente este “cumplimiento” de los valores, el enfoque a seguir es la propagación ascendente, empezando por los nodos hoja (*ckPIs*):



**Figura 4.29:** Avance iterativo de ponderación y agregación en niveles del árbol

En este paso se realizan dos de las operaciones matemáticas estudiadas en la sección 4.1.2. El objetivo es avanzar desde los nodos hoja (*ckPI* inicial) hasta el nivel superior, iterativamente hasta el nodo raíz. Cada *ckPI* tiene asignado un nodo padre, que puede ser considerado ya en sí mismo como un indicador compuesto. En

este sentido, en cada nuevo nivel hay que ponderar y agregar diferentes nodos de nivel inferior.

A diferencia de lo que se hace en el paso anterior, en este punto se trabaja sobre valores (normalizados) que representan diferentes indicadores. Con respecto a la ponderación, el programa desarrollado ha sido preparado para aceptar varios métodos de asignación de pesos, que son seleccionables por el usuario a través de la interfaz de configuración. A continuación se expone cómo se comporta el servicio ante la selección de cada uno de los métodos:

- Ponderación igualitaria: Este método asume que se supone que cada subíndice tiene el mismo impacto en el índice de nivel  $n$  que el resto de elementos que lo componen. La lógica del servicio cuenta los nodos hijo que cuelgan del mismo nodo padre y realiza una división de cada valor en el nivel  $(n-1)$  por dicho número.

$$\text{Factor de ponderación} = \frac{\text{valor}_{\text{nodo (nivel } n-1)}}{\text{Nodos en nivel } n - 1} \quad (14)$$

- Ponderación manual: Dentro de cada nodo se puede establecer un valor manualmente a partir de la interfaz gráfica del servicio. Estos valores se almacenan dentro del registro de configuración mencionado anteriormente.
- Ponderación por asignación de presupuesto: Si se utiliza la asignación presupuestaria (basada en el conocimiento de los expertos o incluso en la opinión pública), se supone que algunos subíndices tienen un impacto mucho mayor (basado en el conocimiento de los expertos) o se consideran mucho más importantes (basado en la opinión pública) que otros. Tecnológicamente, muy similar al caso anterior, no obstante, esta vez no es el usuario quien lo introduce de forma manual, sino que es el instalador de la solución quien, en la fase de despliegue, consulta con el personal involucrado en el diseño del índice compuesto de dicho escenario, y aplica la configuración directamente sobre la base de datos. Se almacena en el mismo entorno de configuración que el punto anterior.

En cuanto a la agregación, el servicio desarrollado contempla ambos casos seleccionados en la sección 4.1.2.2: aditiva y geométrica. La agregación aditiva no requiere de una intervención adicional con la base de datos ni de una lógica compleja, se trata de sumar, para cada nodo hijo (nivel  $n-1$ ) correspondiente al indicador a calcular (nivel  $n$ ), sus valores actuales.

$$\text{Nodo (nivel } n) = \sum \frac{\text{valor}_{\text{nodo (nivel } n-1)}}{\text{Nodos en nivel } n - 1} \quad (15)$$

La adición geométrica (mucho menos utilizada) sí requiere la consulta (de nuevo) del índice de configuración, donde deben encontrarse almacenados los factores de multiplicación, acorde a la fórmula comentada en la sección 4.1.2.2.

Estos factores deberán ser informados al instalador/desarrollador de la solución para su inserción en la base de datos (sin intervención del usuario final).

En resumen, el CI final se crea como un árbol inverso (de ahí el nombre de la configuración), desde los nodos hoja (*cKPIs*) hasta la rama final (CI), asignando varios pesos y realizando operaciones de agregación

### **Paso 7: Construcción del JSON con los resultados obtenidos de los cálculos**

Una vez realizado el cálculo del valor resultado de todos los nodos del árbol, el servicio debe construir un elemento de información en formato JSON siguiendo un modelo de datos determinado. Este modelo de datos es muy relevante ya que el módulo de visualización deberá basarse en la interpretación de su estructura para representar los datos en los gráficos indicados.

Para este modelo de datos, se eligió utilizar el mismo formato definido como entrada de los indicadores iniciales: *ComposedKeyPerformanceIndicator*. No obstante, varias particularidades aplican a la construcción de este resultado.

- Primeramente, en lugar de ser una entidad JSON única (como cada *cKPI*), se tratará de varias entidades JSON, cada una representando un nodo del árbol de cálculo.
- Esta salida, que incluye todos los nodos del árbol de cálculo, expresa cada uno de ellos como un indicador (o *cKPI*), diferenciados con el campo “*type*” y “*level*”, que indican la posición dentro del árbol de ese JSON.
- Debido a que no se realizan las mismas operaciones sobre todos los nodos del árbol (el nivel inicial no posee nodos hijo y el final no posee nodo padre, por tanto, no se combina), se acordó que para cada uno de estos niveles diferentes se realizarían unas asunciones sobre el formato. Por tanto, el programa crea diferentes JSON individuales para los nodos de diferentes niveles dependiendo del nivel del árbol en el que se encuentren.
- Como complemento de los dos anteriores, el JSON del nodo raíz (CI) sólo indicará el nivel (“*level*”) cero (0) el método de agregación y el valor. Los JSON de los indicadores de niveles inferiores (1 al  $n-1$ ) incluirán los métodos de agregación y ponderación utilizados, así como el valor final de cada uno. Por último, el JSON más extendido, correspondiente al nivel inicial u hoja ( $n$ ), contendrá la información de normalización (método y valores utilizados), el origen, el peso, las unidades y la categoría de cada *cKPI*.
- Parte de la información que se incluye en el modelo de datos final se extrae directamente de los *cKPI* de entrada (por ejemplo, los intervalos temporales de cálculo) y otra parte (principalmente, el valor de cada nodo) de los resultados obtenidos.

- Se realizaron asunciones adicionales para facilitar la interpretación de estos resultados por parte de la lógica de interfaces en el módulo de visualización: (i) el campo “id” se completa indicando el nivel (primera cifra) y el elemento dentro del nivel (se establecen 5 cifras, por tanto se aceptarían hasta 100.000 nodos en cada nivel, aunque esto sería configurable). Se agregan los campos “valueNorm”, “normalization” y “normalization\_values” en el JSON del nodo hoja para indicar el resultado y procesos usados en la normalización.

La siguiente figura proporciona ejemplos de los tres tipos de JSON individuales que compondrán el vector de resultados del servicio.

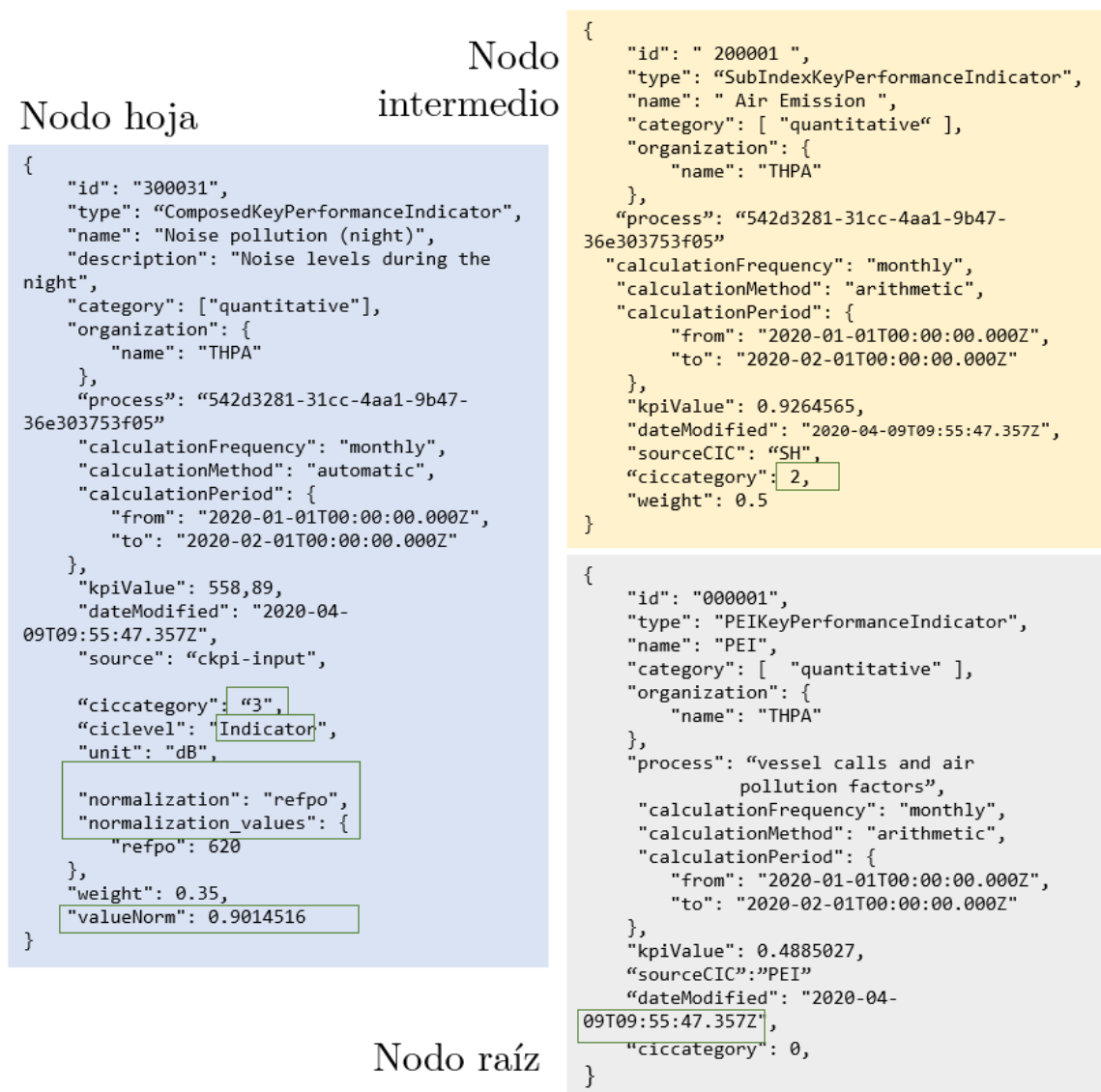


Figura 4.30: Formato JSON de resultado de cálculo en cada nivel de nodo

### **Paso 8: Almacenamiento en persistencia de los valores calculados en el servicio**

El último paso es la inserción de dichos resultados en el sistema de persistencia de la solución. De forma similar a los puntos iniciales donde se consultaba la información de configuración y los datos de los *cKPIs*, este paso requiere la conexión con el API del módulo de persistencia de la arquitectura.

Como se ha comentado en 3.4.2, se diseñaron dos servicios paralelos diferentes (habitual y alternativo) que difieren, principalmente, en este punto. Mientras que el primero utilizará el API de la base de datos Elasticsearch [456], el segundo hará lo propio con el API de la tecnología base de datos MongoDB [457]. Se trata de almacenar los resultados en el índice acordado, *output-ckpi*.

#### *Implementación del servicio:*

Acorde a los principios de la arquitectura definida (sección 3.2.1), todos los servicios a ejecutar deben ser encapsulados en una imagen Docker para poder ser orquestados por el módulo de ejecución de servicios. Con esto en mente, la selección de tecnología para la implementación del servicio quedó en un segundo plano aun manteniéndose relevante. Las últimas páginas han proporcionado toda la información valiosa suficiente para convertir la metodología de cálculo del PEI en una herramienta práctica. Es decir, convertir el diseño teórico en un código ejecutable.

En este sentido, las opciones para la implementación cálculo del índice compuesto fueron variadas. Aunque Python prometía ser una opción interesante, el candidato decidió desarrollar el código en Java debido a (i) los trabajos anteriores sobre árboles y árboles de decisión binarios (similares al funcionamiento de una red) [458], (ii) los conocimientos y la experiencia del doctorando y (iii) las ventajas del paradigma orientado a objetos para este caso en particular. Python también posee capacidades de programación basada en clases, no obstante Java presentaba mejor eficiencia y rendimiento y su facilidad de depuración.

La estructura técnica de cálculo que se ha expuesto anteriormente está, en términos de código, gobernada por una clase principal (práctica habitual en los programas Java) que ha sido denominada “*CIC Calculator*”. Además, se han utilizado clases para definir al árbol (*Tree*) a los nodos (*Node*), a las relaciones entre ellos (*Edge*) y a los elementos de salida. Se ha hecho uso de herencia y otras cualidades de la programación orientada a objetos.

Un recurso útil para entender cómo se traducen las instrucciones en código ejecutable es incluir un diagrama de clases. La siguiente figura representa una versión resumida de los diagramas de clase presentes en el código desarrollado. El Apéndice A contiene todos los diagramas de clase. Dichos diagramas han sido generados con la herramienta ObjectAid UML (plugin del IDE Eclipse) [459].

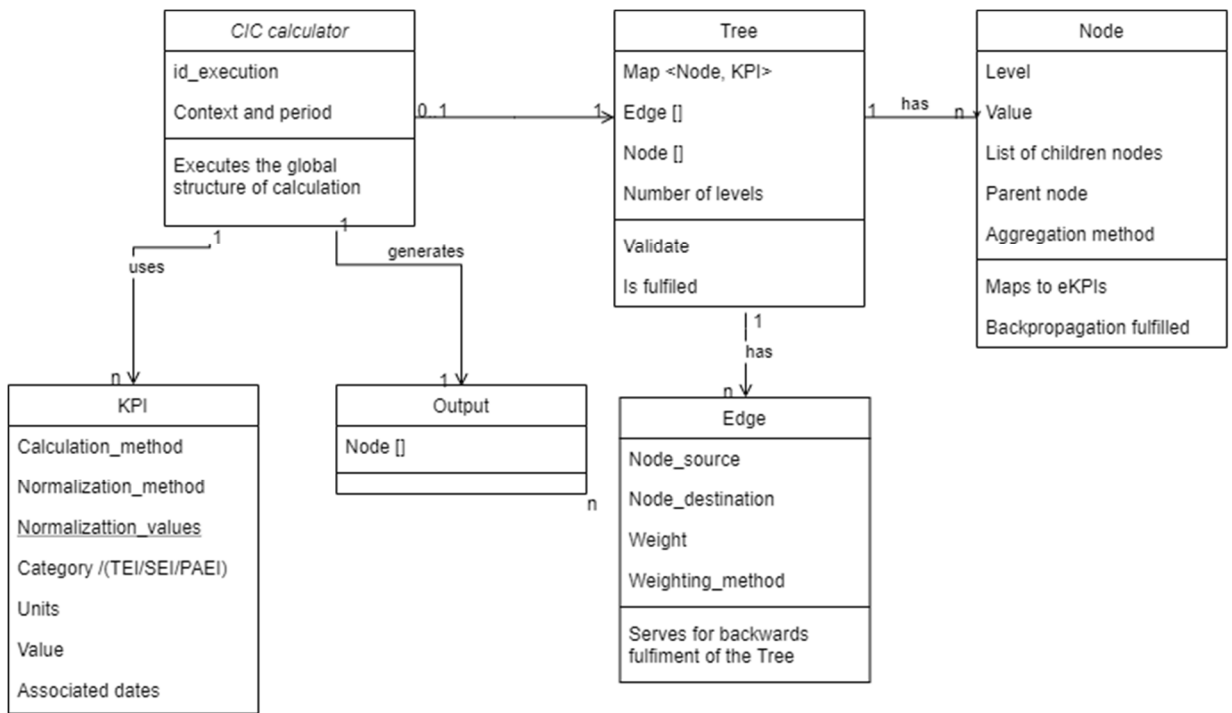


Figura 4.31: Diagrama de clases (simplificado) del servicio desarrollado

Otro recurso interesante para entender el funcionamiento del servicio es clarificar el flujo de interacción entre los módulos y dichos pasos de la aplicación. Se muestra en las dos siguientes figuras.

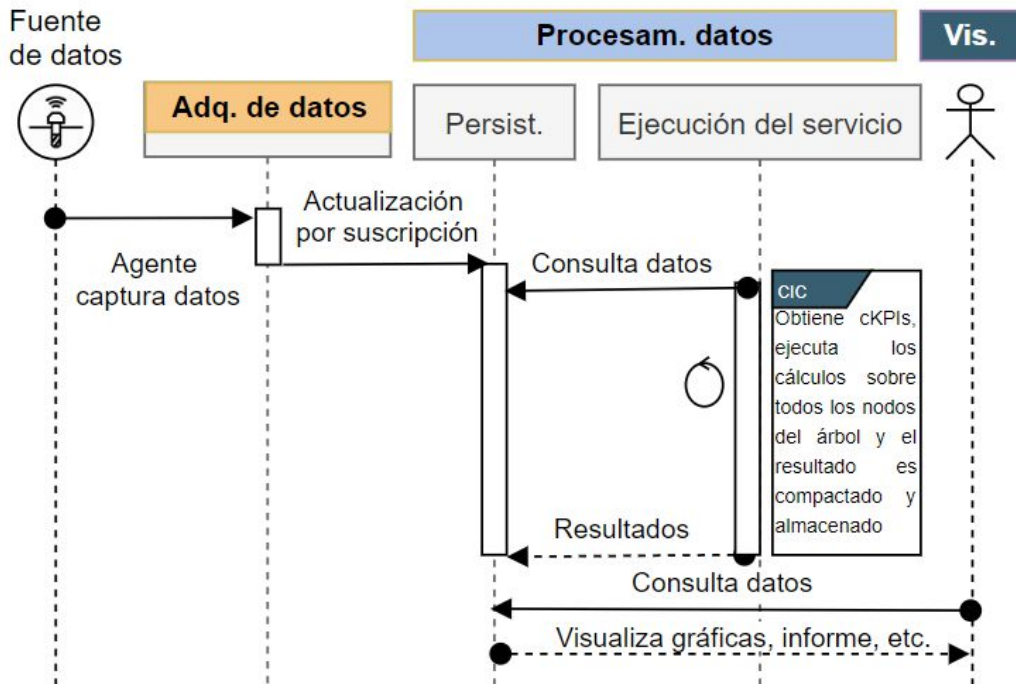


Figura 4.32: Diagrama de secuencia de la actuación del servicio



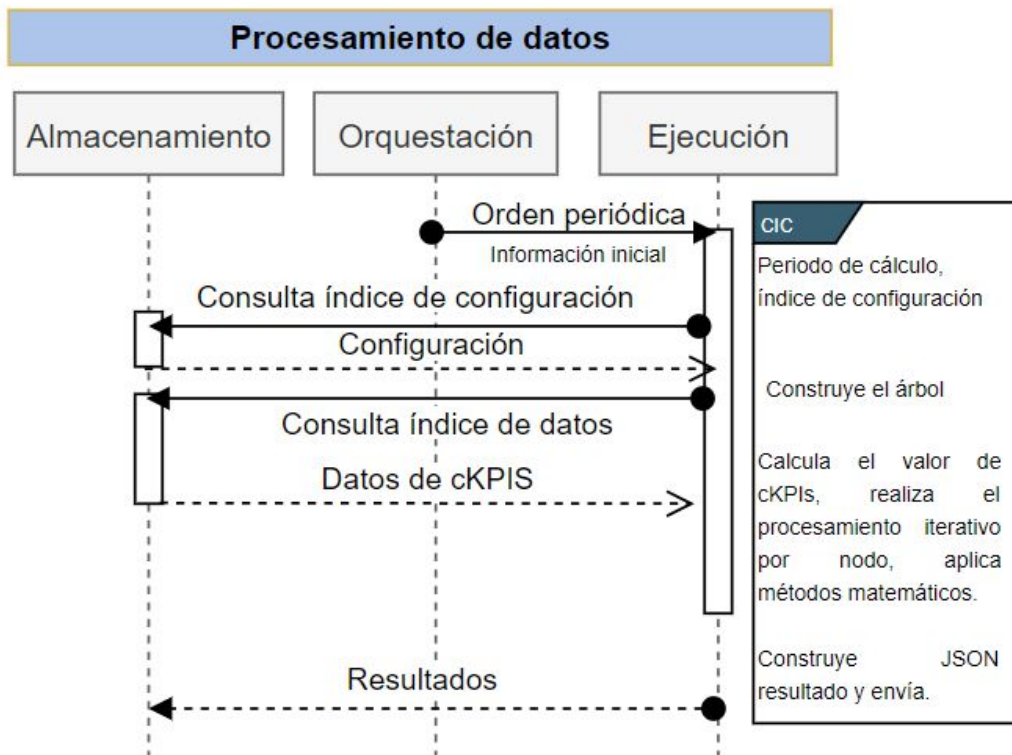


Figura 4.33: Diagrama de secuencia dentro del módulo de procesamiento de datos

## 4.2.3. Predicción de índices

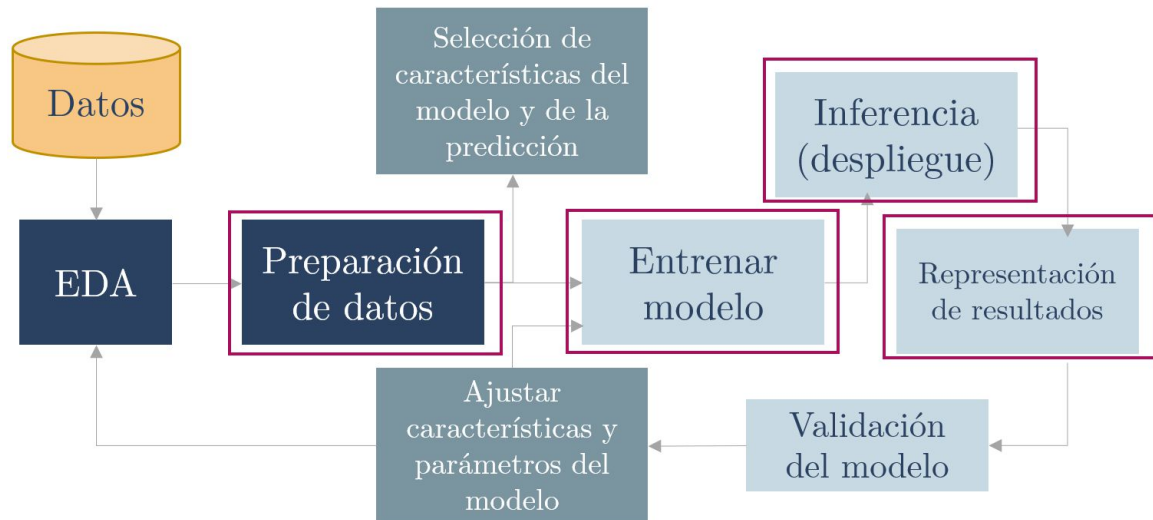
### 4.2.3.1. Predicción de indicadores cKPIs

El segundo servicio objetivo de la tesis doctoral fue la predicción de índices compuestos. Esta subsección aborda la descripción de este servicio dentro de la solución de IoT planteada.

El objetivo es que la arquitectura permita la ejecución automatizada de este servicio. Acorde a los requisitos expresados en la sección 3.1.1, se desea que “servicios de aprendizaje automático (ML), éstos puedan incluirse de forma nativa en la arquitectura. La arquitectura debe estar preparada para este procesamiento y debe disponer de una metodología clara para la creación y uso de modelos ya que la naturaleza de éstos difiere de la de otros servicios”. La automatización e integración surge tras el desarrollo y aplicación iniciales de estos mecanismos de forma aislada, tal como son expuestos en la publicación del trabajo del candidato en la revista Sensors [460]. El servicio integrado ha sido validado en varios escenarios (véase sección 5)

A la hora de implementar un servicio de este tipo, debe plantearse qué acciones (dentro del espectro de tareas asociadas a *machine learning*) han de ser incluidas en el mismo. Según la siguiente imagen (inspirada en los pasos en ciencia

de datos [461]), los ámbitos a incluir dentro del servicio son: la preparación de los datos, el entrenamiento de los modelos y la inferencia de datos a partir de los modelos obtenidos. Finalmente, también debe representarse dichos resultados de inferencia.

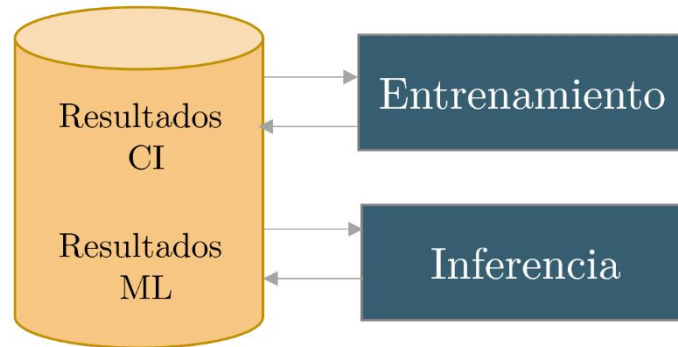


**Figura 4.34:** Ámbitos de acción del servicio de predicción de índices compuestos

Se ha llegado a esta conclusión partiendo de la base que el análisis exploratorio de los datos (EDA – *Exploratory Data Analysis*) se realiza “*offline*”, previa a la integración en la solución. Además, el ajuste de los parámetros se ha dejado también a criterio del desarrollador/instalador que controle el servicio, excluyéndolo de la ejecución automática planteada.

Tal y como se ha expuesto en diferentes capítulos, datos de diversas fuentes son capturados e insertados por parte de unos agentes que realizan un preprocesamiento de los mismos (lo que se conoce como *data curation* en ciencia de datos) y los sirven al sistema en un formato unificado (*cKPI*), contextualizados a un momento del tiempo (véanse los campos del modelo de datos de entrada – sección 4.2.1). Es decir, los datos manejados por la solución son series temporales. Por tanto, una vez dentro del sistema, se recibe información de dichas series en un formato concreto. El planteamiento de modelos de predicción de series temporales en entornos de IoT ha sido revisado en la sección 2.4.1.

Considerando todo lo anterior, se ha dividido la actuación de este servicio, de hecho, en dos servicios independientes (microservicios – sección 3.2.1), cada uno de ellos cubriendo una de las dos acciones. Así (como se muestra en la siguiente figura), un primer servicio de entrenamiento de modelos actuará sobre los datos históricos almacenados de estas series temporales y los guardará en el módulo de persistencia para poder ser utilizados en la fase de inferencia posterior. Este segundo servicio (predicción) recuperará dichos módulos y almacenará los resultados estimados de nuevo en la base de datos.



**Figura 4.35:** Descomposición de servicios de predicción

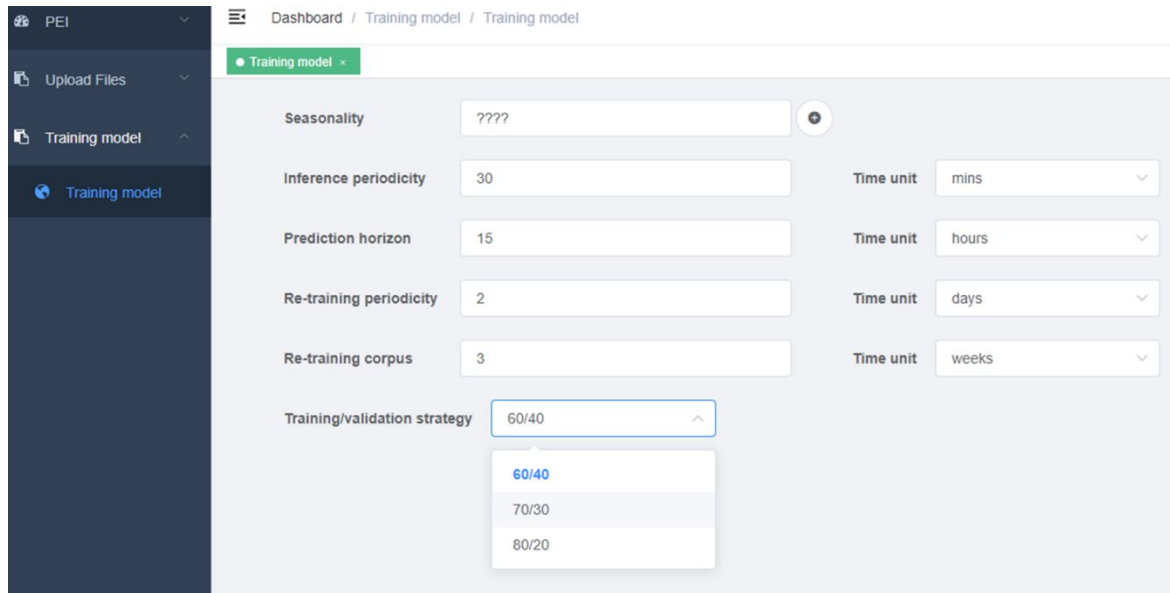
En el caso de la solución propuesta, debe también elegirse qué dato es el que desea predecirse. Como se ha expuesto, el cálculo de CI parte de un número de indicadores iniciales (*cKPIs*). Estos indicadores son manipulados de forma iterativa generando nodos intermedios y un nodo final de un árbol jerárquico, cuya reducción a un único valor resulta en el índice compuesto deseado. Entendiendo cada uno de dichos elementos como una serie temporal con registros crecientes con el tiempo, podría aplicarse el entrenamiento de modelos e inferencia sobre cualquiera de ellos, sobre todos ellos, sobre alguno... Se optó por una creación de modelos por cada uno de los indicadores iniciales (*cKPIs*) por cuatro motivos principales:

- 1) Aporta más valor a un usuario poder ver cómo va a evolucionar cada uno de los valores que contribuyen al indicador compuesto, adelantándose a aquellos que vayan a tener mayor impacto. A nivel de representación gráfica también resulta más interesante.
- 2) Si se eligiera realizar la predicción sobre el indicador final únicamente, se perdería riqueza y variabilidad en los modelos, ya que dicho valor ha sido ya reducido desde varias líneas de entrada. Por tanto, posee mayor entropía que los elementos individuales [462].
- 3) Si se desea obtener una predicción sobre el valor del índice compuesto final, basta con ejecutar el servicio de CI principal modificando el índice/colección de la base de datos en el JSON de configuración por parte del instalador, apuntando al espacio donde se encuentran los datos predichos (sección 4.2.2).

Como visión general del ciclo del servicio, en primer lugar deberá realizarse las configuraciones pertinentes, seguido del entrenamiento de un modelo de *machine learning* para cada indicador inicial. Posteriormente, estos modelos se utilizarán para calcular la predicción de cada *cKPI* en un horizonte determinado (a corto plazo) que debe ser representada en el módulo de visualización del sistema.

Similarmente al caso del servicio principal (sección 4.2.2), las primeras acciones son la configuración de parámetros para la ejecución del servicio. Del mismo modo, existe tanto configuración por parte del usuario final (relacionada con aspectos operativos de los modelos y la predicción) como por parte de un personal más técnico (conocedor de los detalles de despliegue de la arquitectura).

La información que debe introducir el usuario final es la que se muestra en la captura de pantalla de la siguiente figura, que se ha incluido dentro del módulo de visualización de la arquitectura (sección 3.5). Esta imagen está conectada con los desarrollos realizados por el candidato para el proyecto de investigación PIXEL. Esta interfaz, junto con la mostrada en la Figura 41, fueron modificadas y adaptadas para el funcionamiento dentro de la arquitectura de dicho proyecto:



**Figura 4.36:** Configuración del usuario para el servicio de predicción

- Estacionalidades: Las series temporales presentan, frecuentemente, componentes estacionales, que son ciclos de duración fija o variable que se repiten con una frecuencia determinada. A través de estas opciones el usuario puede introducir (mediante una herramienta de selección de calendario) si existe alguna estacionalidad sobre los datos. Este conocimiento debe provenir del análisis exploratorio de datos realizado previa integración del servicio. Una estacionalidad frecuente (que se ha utilizado en los escenarios de validación) es días festivos. Esta información es útil para el modelo ya que le permite ser consciente y actuar (entrenar) en consecuencia.
- Periodicidad de la inferencia: Cada cuánto debe realizarse la predicción. Este valor es usado como entrada por el servicio de predicción. Puede elegirse la unidad de tiempo (minutos, horas, días, meses).
- Horizonte de predicción: El tiempo futuro para donde se predice dicho valor. En general, como se ha comentado en la sección 2.4.1, el caso de esta tesis se interesa por predicciones a corto plazo de tiempo (hacia la siguiente ejecución planificada del servicio de CI principal).
- Periodicidad de reentrenamiento: Cada cuánto tiempo se realiza un reentrenamiento (*fit*) de cada modelo.

- Corpus de reentrenamiento: Este parámetro sirve de entrada al servicio de entrenamiento y tiene un efecto directo sobre la cantidad de datos a utilizar, y por ende, el tiempo y recursos computacionales requeridos para completar dicho entrenamiento.
- Estrategia de entrenamiento/validación y test: Se especifica la estrategia de partición del *data set* para el entrenamiento. Se ofrecen las opciones 60/40, 70/30 y 80/20, que son las más utilizadas en estos problemas de *machine learning*.

Por otro lado, el servicio requiere de una información técnica que deriva directamente de la naturaleza de la arquitectura (sección 3.2.1).

```

1  [
2  {
3    "input": [
4      {
5        "element": "indicators",
6        "URL": "http://172.1.18.65:9200",
7        "index": "arh-lts-output-kpis",
8        "collection":""
9      },
10     {
11       "element": "parameters",
12       "URL": "http://172.1.18.65:9200",
13       "index": "predict-params",
14       "collection":""
15     }
16   ],
17   "output": [
18     {
19       "element": "predicted-indicators",
20       "URL": "http://172.1.18.65:9200",
21       "index": "arh-lts-predicted-kpis",
22       "collection":""
23     }
24   ]
25 }
26 ]

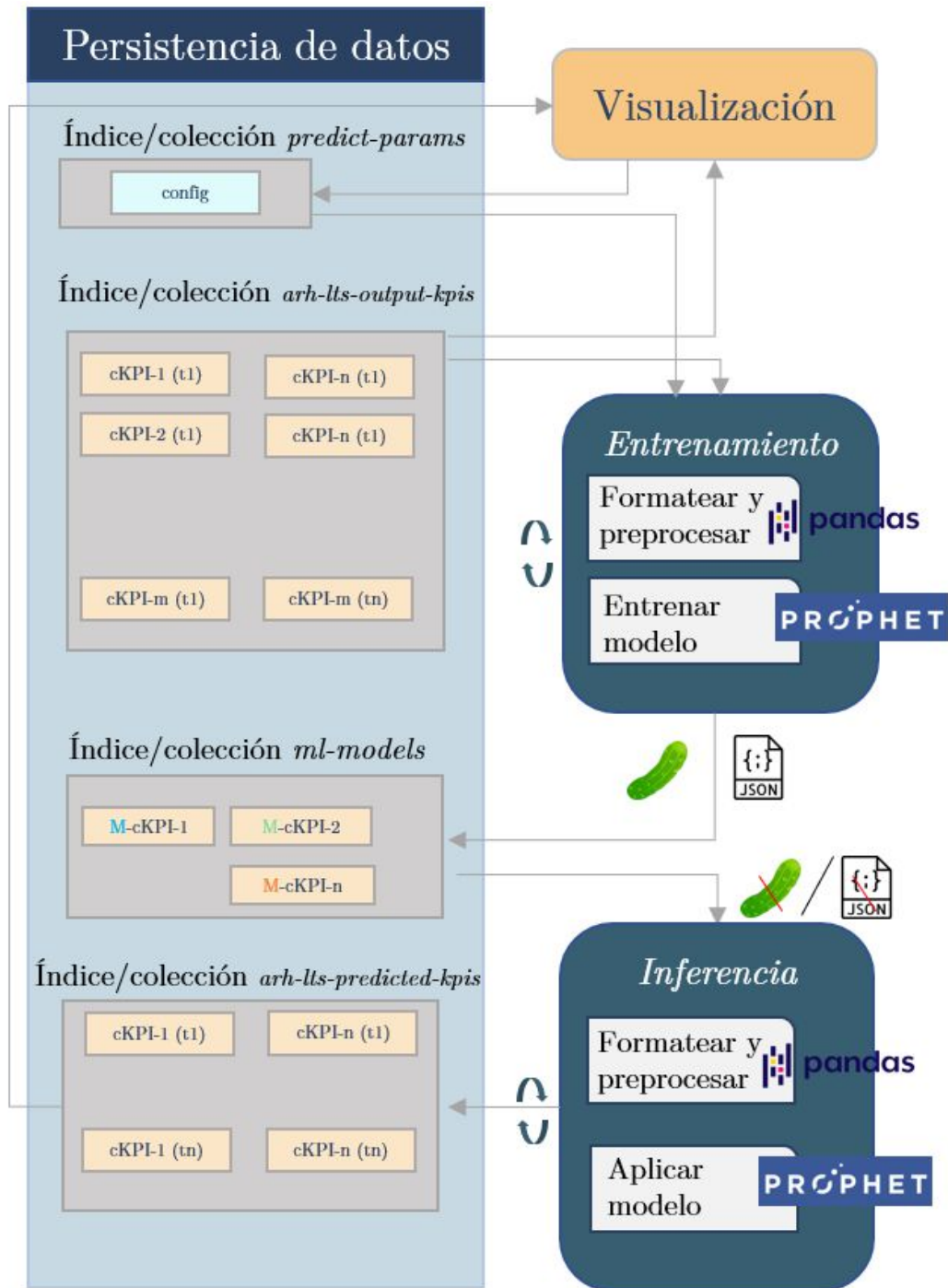
```

**Figura 4.37:** Configuración de desarrollador de servicio de predicción

De manera similar a lo especificado para el caso anterior, debe determinarse la información de dirección, puerto e índice o colección de la base de datos que contiene (véase figura anterior):

- Los datos de configuración del usuario (página anterior).
- Los indicadores iniciales *cKPIs*.
- Dónde se debe depositar el resultado del servicio de predicción.

En la siguiente figura se ilustra de forma esquemática el flujo de los servicios, las interacciones con los módulos de la arquitectura y sus componentes internos.



**Figura 4.38:** Estructura de funcionamiento de servicios de predicción

El resultado de esta estructura ha partido de las siguientes decisiones de diseño:

**Detalles de interacción con los módulos de la arquitectura:**

Del mismo modo que en servicios anteriores, es necesario llegar a una serie de acuerdos para el correcto funcionamiento de este.

En primer lugar, el servicio de entrenamiento deberá conocer el índice o colección donde se encuentran los valores de configuración inicial marcada por el



desarrollador y el usuario final. Se escogió almacenar dichos valores con un formato simple JSON clave-valor en el índice/colección *ml-config*.

A continuación, tras leer esa información, el servicio de *training* realiza su función. Consiste en un *script* (tecnología escogida: Python) que agrupa los datos de los *cKPIs* existentes en dicho índice (en función de la periodicidad marcada por la configuración del usuario), convierte, limpia y adapta la información para el entrenamiento/reentrenamiento de los modelos. Una vez finalizado dicho servicio, para almacenar esos modelos, se utiliza la biblioteca incorporada de Python *pickle* [463] o la serialización JSON de éstos [464]. Esta acción se realiza iterativamente hasta disponer de un modelo para cada *cKPI*. Este almacenamiento se realiza sobre el índice *ml-models* y, en caso de escoger el primer modo de registro, se inserta como un binario en la base de datos (tanto Elasticsearch como MongoDB aceptan este formato como entrada).

En cuanto a la aplicación (inferencia), el segundo servicio (instalado a parte del primero) recupera los modelos almacenados en la base de datos, aplica una descompresión de los mismos (*unpickle* o deserialización JSON) y, según los parámetros leídos de la configuración, realiza una predicción sobre el horizonte temporal determinado. El resultado es insertado en el índice/colección *predicted-ckpi*. El formato con el que se guardan estas predicciones coincide exactamente con el definido para los indicadores iniciales (*ComposedKeyPerformanceIndicator* – véase sección 4.2.1). Esto se realiza de esta forma para asegurar que pueden ser utilizados como base para el cálculo del CI asociado para un tiempo futuro.

En la siguiente imagen se aportan ejemplos de la instalación de dichos servicios, alineado con el diseño de la arquitectura de la solución (sección 3.4.2).

Para el caso de estos servicios, no hubiera sido necesaria la utilización de la herramienta cron, ya que se encuentran para actuar de forma continua (servicio ininterrumpido) realizando las iteraciones internamente según los parámetros leídos.

```
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
* * * * 0 sudo docker run cic_service -e DB_URL='http://172.1.18.65:9200' DB_INDEX='ckpi-config'
* * * * 0 sudo docker run prediction_service -e DB_URL='http://172.1.18.65:9200' DB_INDEX='ml-models'
* * * * 0 sudo docker run training_service -e DB_URL='http://172.1.18.65:9200' DB_INDEX='ml-config'
```

**Figura 4.39:** Instalación de servicios de predicción

Por último, el resultado (registros temporales futuros con valores de *cKPIs* predichos) se utiliza para ser representado para el usuario a través del componente UI. En las siguientes imágenes se muestran las gráficas específicas creadas para visualizar la predicción, aportando visiones generales del servicio en dos escenarios de validación. Las dos imágenes a continuación están en directa conexión con los desarrollos realizados por parte del candidato para el proyecto PIXEL, donde la solución propuesta fue adaptada para el cálculo del impacto medioambiental en los puertos de Pireo, Monfalcone, Burdeos y Tesalónica.

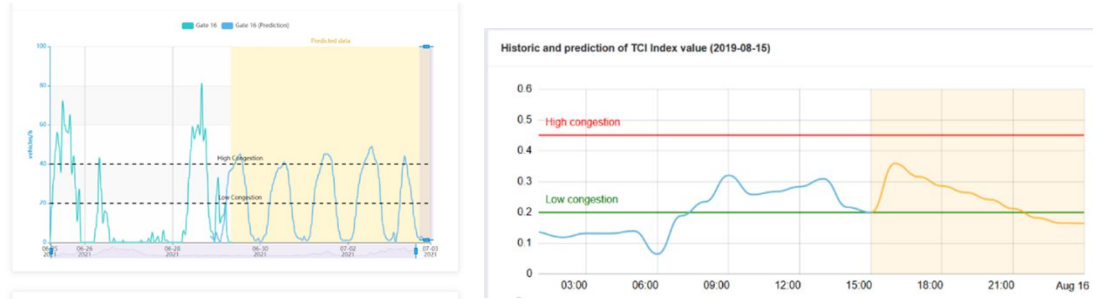


Figura 4.40: UI resultante para la observación del servicio (I)

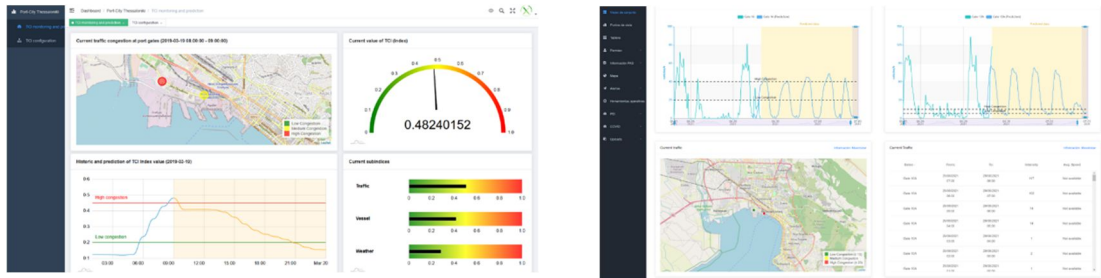


Figura 4.41: UI resultante para la observación del servicio (II)

**Consideraciones y simplificaciones para la elección del modelo:**

Dentro de las ramas de modelo de *machine learning*, tal y como se ha expuesto en la sección 2.4.1, éstos pueden responder a una estructura de aprendizaje supervisado o no supervisado. Lo que se concreta en esta tesis es un problema de series temporales, cuyo objetivo de predicción es atributo numérico a partir de una serie de datos de entrada “etiquetados” con su correspondiente valor cuantitativo. Este tipo de casos requieren modelos de regresión lineal y logística y en enmarcan dentro de los modelos de aprendizaje supervisado [465].

Dentro de éstos, se ha escogido para la tesis una aproximación en que el modelo no necesite consultar los *cKPI* actuales como datos de entrada para la inferencia. Con esta aproximación, los modelos no requieren datos nuevos alimentando a la “caja negra” para alcanzar una predicción. En este caso, se asegura la “validez” y “actualización” de modelo debido a que se entrena periódicamente y se mantiene relevante. Esta decisión se ha debido a:

- 1) Con la intención de reducir el procesamiento necesario a nivel I/O desde/hacia la base de datos.
- 2) Para simplificar la arquitectura separando entrenamiento e inferencia.
- 3) A la elección de la tecnología realizada (véase a continuación).

**Descripción del modelo de *machine learning* escogido y su utilización:**

Para la elección del mejor modelo a utilizar para estos servicios, se realizó un repaso a la sección 2.4.1 así como se revisó los modelos más frecuentemente utilizados en este tipo de problemas. Especialmente interesantes fueron los modelos



ARIMA o los modelos de regresión de series temporales como ARIMA Estacional (SARIMA) o de Vector de Auto-regresión [466], todos ellos ampliamente utilizados y disponibles como código Python en la comunidad *open source*.

Reflexionando sobre las series temporales a manejar (y tras realizar un análisis exploratorio de los datos en los escenarios de validación – sección 5), existen *cKPIs* que presentaban valores perdidos. Además, se observaron varias (y relevantes) estacionalidades diarias, mensuales y anuales. Por lo tanto, era necesario seleccionar un modelo que manejara adecuadamente estos aspectos. A partir de lo anterior, y considerando la facilidad de uso para su integración en la arquitectura, la selección más adecuada fue utilizar Prophet [467], una herramienta de predicción de series temporales de propósito general desarrollada por Facebook. Esta elección se vio reforzada al observar algunas de sus características: fácilmente integrable en despliegues ágiles con datos heterogéneos. En la sección 2 se ha realizado un análisis más detallado de otros posibles modelos del estado del arte para la resolución de aprendizaje supervisado, como métodos de gradiente modernos [468], que podrían superar a Facebook Prophet en márgenes significativos, pero que requerían más esfuerzo y datos para ser integrados, dificultando la capacidad de automatización. Además, estos modelos se antojaron fuera del alcance de la tesis doctoral en cuanto a objetivos y recursos requeridos (especialmente escasos si se trata de despliegues de *edge computing*), por tanto se decidió no explorarlos como opciones realistas. 1

Además de dicha elección, se acordaron los siguientes aspectos para la implementación de ambos servicios con Prophet:

- Los servicios (dos) son desarrollados en Python y encapsulados como imágenes Docker para su despliegue según la arquitectura.
- Tanto el servicio de entrenamiento como el de predicción deben contar con mecanismos de limpieza y preparación de los datos. En este sentido, fue importante eliminar los “falsos valores cero” y marcarlos como ausentes para una mejor gestión por parte del modelo. Además, se utilizó la librería *pandas* [469] para extraer los valores útiles de los *cKPIs* de cara al desarrollo de estos servicios.
- Prophet solo admite la generación de modelos para un valor único a la vez, por tanto resultó ideal para cumplir con el diseño de un modelo para cada *cKPI*.
- A la hora de almacenar los modelos, aunque *pickle* fue una de las opciones indicadas inicialmente, ésta presenta dificultades con Prophet, así que se utilizó un almacenamiento por serialización JSON [470].

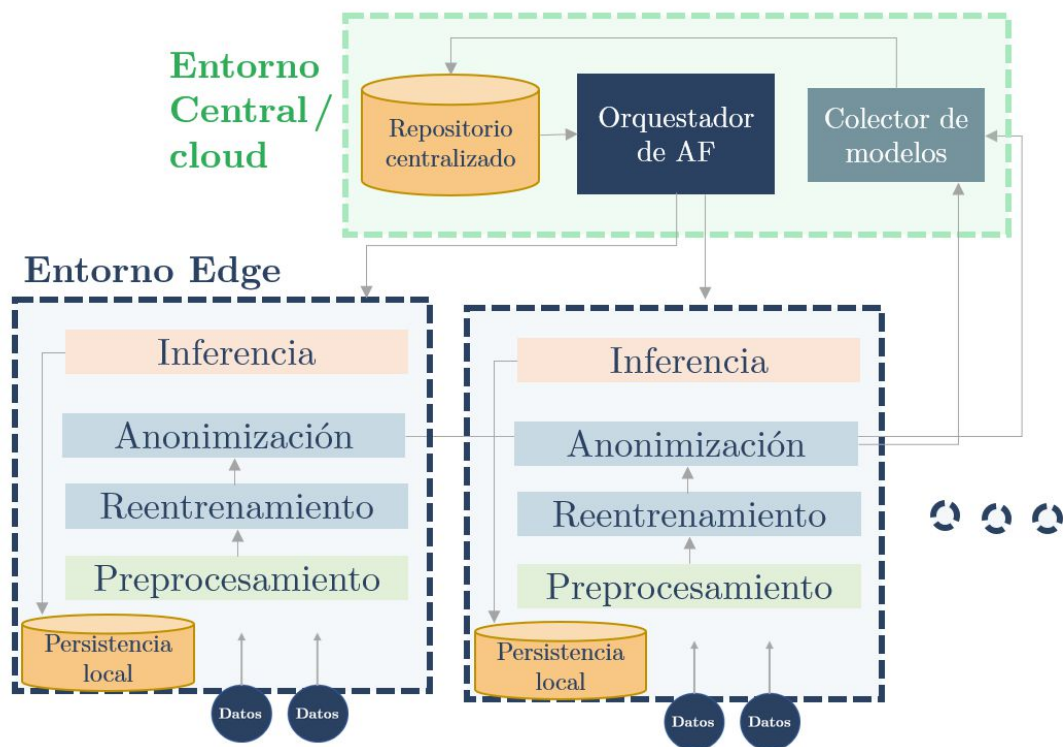
### 4.2.3.2. Propuesta de aprendizaje federado

La sección 2.4.2 contiene un repaso a la perspectiva actual del estado del arte del aprendizaje federado. En pocas palabras, este concepto representa aquellos sistemas distribuidos que tienen como objetivo aprovechar el conocimiento adquirido por parte de entidades separadas (normalmente, distribuidas

geográficamente) colaborando para mejorar la precisión en la predicción, que resultaría complicada, más lenta y con potenciales problemas de seguridad si se realizara en un entorno centralizado.

En este sentido, en la tesis doctoral se ha realizado únicamente una aproximación teórica a cómo podría introducirse un sistema de aprendizaje federado alineado con la arquitectura diseñada. Ninguno de los escenarios de validación ha contemplado el aprendizaje federado, ni ha podido validarse las asunciones realizadas en esta subsección más allá de pruebas individuales por parte del candidato. Además, conviene destacar que esto ha sido debido, en parte, a la inmadurez tecnológica de este tipo de soluciones. No se ha encontrado una gran evidencia de desarrollo de entornos de aprendizaje federado en ámbitos industriales operativos.

Para realizar la propuesta teórica de este marco de uso potencial, se ha partido de la base taxonómica descrita en [471]. Acorde a dicha nomenclatura y funcionalidad objetivo, se plantea el diseño que se muestra en la siguiente figura:



**Figura 4.42:** Propuesta de integración de aprendizaje federado en la arquitectura

Los diferentes elementos serían los siguientes, que se desplegarían y ejecutarían como servicios independientes según la arquitectura propuesta (3.2):

- **Orquestador de Aprendizaje Federado:** Elemento que se encarga de establecer el flujo de despliegue, recogida, programación de los trabajos, entrega de versiones iniciales de los modelos, estructura de compartición y gestión de errores en el proceso de aprendizaje federado.

Residiría en el entorno centralizado y requeriría acuerdos y mecanismos sobre configuraciones de usuario e índices de la base de datos.

- **Repositorio de modelos:** Espacio en base de datos para el almacenamiento de los modelos compartidos/a mejorar, así como de metadatos para informar sobre sus usos, fecha, etc. Dentro de la arquitectura de la tesis podría constituir un índice del módulo de persistencia, y permitiría almacenar los modelos como binarios a partir de archivos *pickle* o en formato JSON serializado.
- **Colector de modelos:** El módulo responsable de recibir las actualizaciones locales del modelo preparadas por cada entorno local *edge*, así como depósito en el repositorio. Éstos serán recuperados por el orquestador para su despliegue en las diferentes entidades para las rondas de entrenamiento posteriores
- **Operaciones Locales:** Diferentes componentes ejecutados en cada entidad del *edge* que se encargarán del preprocesado de los datos (verificación de compatibilidad de formatos, transformaciones...), del entrenamiento local del modelo según las instrucciones del orquestador, del envío de las recomendaciones encriptadas al colector de modelos para su agregación, y de la inferencia del modelo sobre los datos recién llegados (o, si se sigue la asunción de la sección anterior, sin necesidad de la incorporación de los datos más recientes).

Para la implementación de este sistema, se debe adaptar el trabajo propuesto en Flower [472] a las características del servicio buscado y del escenario planteado. Si bien se propone un marco de trabajo (fase de madurez reducida) amigable que permite el uso de modelos TensorFlow, PyTorch y MXNet, y proporciona unas instrucciones de aplicación razonables, se trata de un funcionamiento excesivamente complejo para los casos de cálculo de CIs con respecto al objetivo expuesto en esta tesis doctoral.

En particular, esta propuesta se realiza para aquellos casos en que los datos contribuyentes al cálculo del índice compuesto se recojan, almacenen y procesen en entornos *edge computing* separados geográficamente. Así, cada entorno de operaciones locales dispondría de los datos históricos y actuales de una serie de *cKPIs* (los asociados a las fuentes de datos de ese sitio) y podría realizar una predicción de estos de forma local. Estos modelos se compartirían posteriormente para poder ser aprovechados en otras localizaciones con fuentes de datos similares. Este despliegue también puede ser útil para escenarios donde diferentes entidades adoptantes decidieran colaborar para mejorar su precisión en la predicción de indicadores.

#### 4.2.4. Servicios auxiliares

En las dos subsecciones anteriores se han expuesto los dos servicios principales objetivo de la tesis doctoral (cálculo y predicción de CIs). No obstante, para cumplir con los requisitos definidos (sección 3.1.1) se precisó la inclusión de servicios complementarios adicionales.

En particular, en dicha sección se especificó que el CI debe ser capaz de aceptar indicadores (*cKPIs*) de diferentes formas, al igual que debía permitir la creación de reglas sobre los resultados de la ejecución y la indicación de recomendaciones sobre las mismas. Se incluyó, además, la necesidad de generar informes sobre los resultados de un periodo determinado. Al margen de ello, algunas peculiaridades de ciertos escenarios del servicio llevaron al candidato al planteamiento de varios servicios externos.

- **Servicio de extensión del cálculo a escenarios particulares:** En los casos en los que varias entidades adopten el mismo servicio de CI, y este deba ser comparado entre ellas o deban regirse por un mismo mecanismo de normalización conjunto, no será suficiente con incluir la lógica dentro del servicio CIC. Se necesitará la interacción con algún elemento externo.
- **Servicio de recomendaciones en base a reglas:** Deberían poder introducirse reglas (umbrales sobre los valores resultado en los nodos) para realizar recomendaciones. Siendo el servicio planteado en la tesis una herramienta orientada a guiar la toma de decisiones en materia de políticas o acciones para mejora de algún impacto (por ejemplo, medioambiental o de sostenibilidad), el hecho de aportar recomendaciones de manera automática basadas en los valores obtenidos del indicador compuesto otorga a la solución de un valor añadido.
- **Servicio de contextualización de índice compuesto:** A pesar de que la solución cumple con el requisito de aceptar fuentes de todo tipo, no todas ellas pueden ser consideradas iguales. Para evitar una interpretación sesgada del índice compuesto, se decidió incluir un índice adicional adjunto al mismo para representar, de alguna manera, la fiabilidad del valor del CI en función de las fuentes y los mecanismos de captura de datos planteados.

Si bien estas funcionalidades podrían incluirse directamente dentro de la lógica del servicio (en el código del mismo programa Java), este hecho le hubiera restado flexibilidad, ya que un mínimo cambio sobre estas funcionalidades complementarias hubiera requerido una nueva versión del servicio, nueva instalación, despliegue, etc. De este modo (externalización del servicio) se desacoplaban dichas atribuciones, permitiendo un mejor ajuste a los principios de diseño de la solución. Este aspecto sirvió, además, para validar la naturaleza modular y flexible de la arquitectura. No

debió realizarse ninguna modificación estructural ni integral sobre la solución, únicamente llegar a una serie de acuerdos y convenciones sobre el formato y modelo de datos a compartir y en qué índices/colecciones, e instalar dichos servicios auxiliares del mismo modo que se realizó con los principales (véase sección 3.4.2).

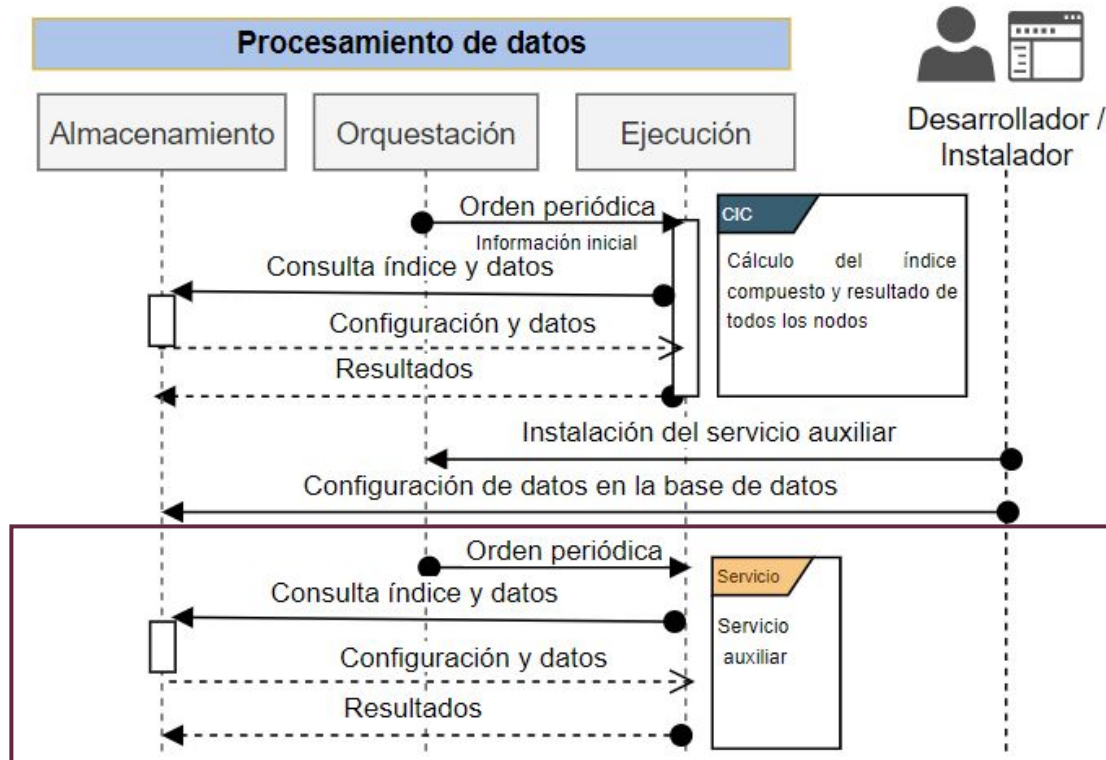


Figura 4.43: Flujo de integración de servicios adicionales

**Servicios para la extensión del cálculo a escenarios particulares:**

Estos servicios se diseñaron para aquellos casos en los que la utilización del índice compuesto fuera relativamente fija y fuera desplegado (con las mismas condiciones y configuraciones) en varias entidades (“adoptantes”). Este caso se correspondería, por ejemplo, con uno de los perfiles de explotación identificados en la sección 6, donde una entidad pública (o de auditoría o certificación) requeriría la instalación de la solución en varias compañías de un sector para medir su impacto sobre un determinado fenómeno. En dicho caso, una de las motivaciones subyacentes podría ser la comparación entre el rendimiento/impacto resultante en cada una de ellas, bien fuera para realizar concesiones/imponer sanciones o simplemente para permitir dicha comparación desde un punto de vista informativo.

Para poder realizar estas comparaciones (“ranking”), se precisa que todas las entidades adopten la solución asumiendo las siguientes condiciones:

- Todas las instancias desplegadas de la solución cuentan con los mismos indicadores iniciales (*cKPIs*) y éstos no son modificables (eliminación, adición, edición).
- Todos los *cKPIs* de todas las entidades son normalizados de la misma forma. Según lo estudiado al principio de este capítulo (sección 4.1.2.2), la normalización es el instrumento que se utiliza para hacer que dos valores adquieran la misma escala y sean, por tanto, agregables y comparables.

En cuanto al último punto, se infiere que se requiere una forma de comunicar a los diferentes adoptantes de la solución de los valores sobre los que tienen que normalizar sus *cKPIs*. Por tanto, los métodos de normalización de reescalado (min-max) o estandarización sobre la misma base de datos de una entidad particular no son posibles, ya que no se estaría utilizando la misma referencia. Lo que resultó evidente fue la necesidad de establecer una entidad centralizadora de esta referencia (entidad central imparcial) que mantuviera actualizada esta información para poder ser consultada por todas las entidades adoptantes.

Tras el análisis realizado, se ha concluido que una buena manera de gestionar esta normalización es implementando esta base de datos centralizada de conocimiento para establecer el valor de referencia de normalización. Esta opción, implica, no obstante, una serie de premisas:

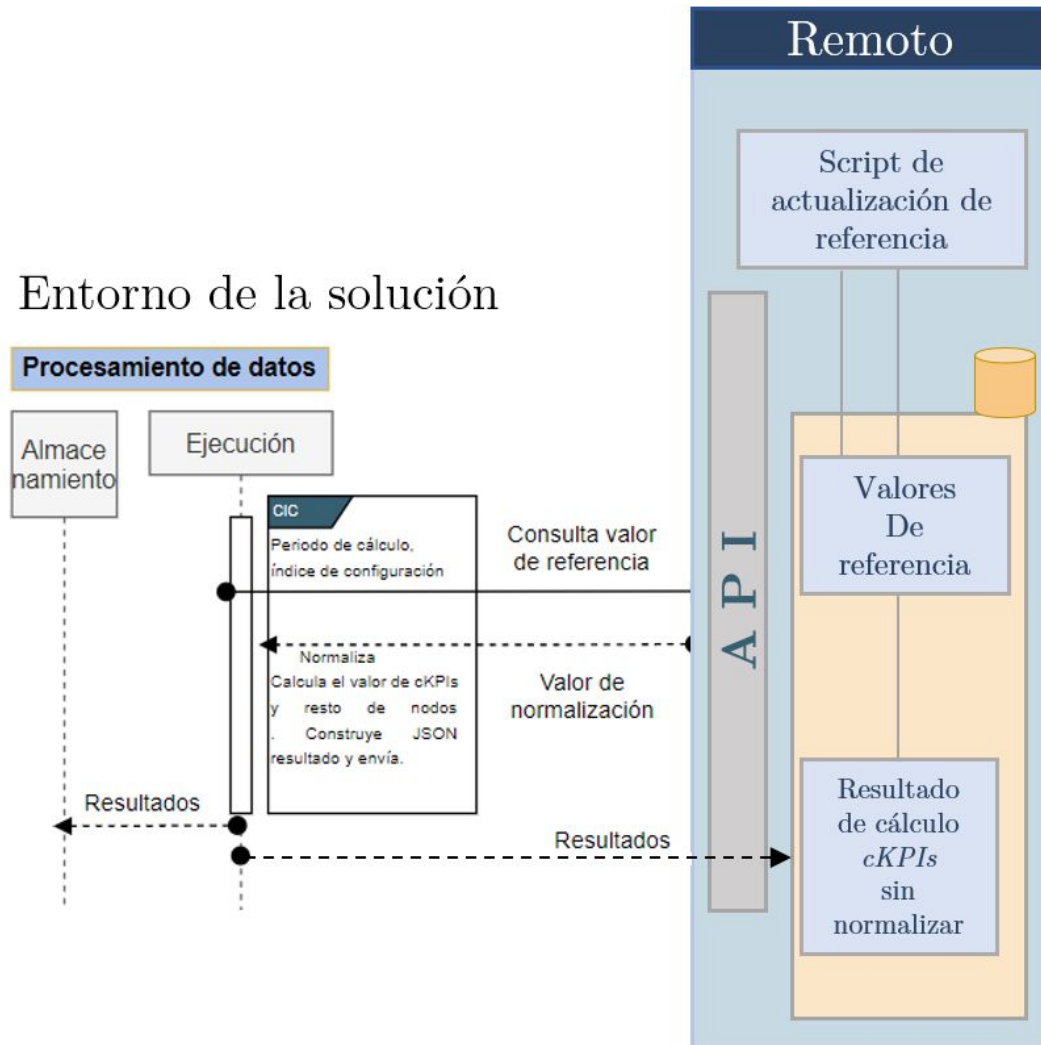
- (i) No se conoce, a priori, la escala de valores en las que las entidades van a moverse, por tanto se corre el riesgo de establecer unos valores de referencias irreales, conduciendo a unos *cKPIs* normalizados muy grandes o muy bajos, dificultando la interpretabilidad del índice compuesto.
- (ii) La realidad es cambiante y con el paso del tiempo, para mantenerse válida y no incurrir en los riesgos mencionados en (i), esta referencia debía actualizarse.

Por tanto, se decidió que la mejor manera de proceder fuera establecer una normalización por distancia a punto de referencia y que éste fuera calculado dinámicamente a partir de la información de cálculo del CI de todas las entidades. De este modo, la entidad que mejor rendimiento tuviera (con granularidad de *cKPI*), sería la que establecería el punto de referencia de cara al siguiente periodo de refresco. Es por ello que resulta crucial que todas las entidades posean la misma estructura del indicador (árbol), ya que el servicio externo debe basarse en este conocimiento para ofrecer la información de normalización.

En un primer momento, cuando aún no se ha realizado ninguna ejecución del servicio por parte de ninguna entidad, se introduce como registro inicial de referencia una información extraída del análisis de los datos. Con ellos, se establece un valor de referencia por cada indicador (*cKPI*), y se normaliza contra el mismo (primera ejecución). A partir de dicho momento, el elemento centralizador recibiría los valores de *cKPIs* (sin normalizar) de cada ejecución de las entidades, captaría

el valor mínimo obtenido en cada *cKPI* de todas las entidades en el último periodo y lo establecería como la nueva referencia.

El funcionamiento de este servicio adicional se ilustra en la siguiente figura:



**Figura 4.44:** Servicio externo de normalización desde entorno remoto

Durante la ejecución del servicio de cálculo de CIs, en el momento de consultar la base de datos para obtener el valor de normalización, en lugar de hacerlo sobre el sistema de persistencia de la solución, se realiza una consulta a una API remota que devuelve la información requerida (por *cKPI*). Para ello, el elemento externo (en este caso se trata de un servicio en un entorno diferente al de la solución) posee un índice/colección con los datos de normalización actualizados y un API capaz de responder satisfactoriamente a dicha consulta. A continuación, prosigue la ejecución del servicio de CI. Una vez finalizado, se realiza un envío a la misma API remota conteniendo la información del resultado de cada *cKPI* (incluyendo el valor normalizado y sin normalizar). Para no realizar complicaciones innecesarias en la lógica del servicio, se aprovecha las consultas HTTP realizadas contra el API REST del módulo de persistencia y se duplican indicando la dirección (y otros parámetros) del servidor remoto, enviando únicamente la información de los nodos hoja (*cKPIs*).

Tecnológicamente, el servicio remoto contiene una base de datos con los valores de referencia actuales (en un espacio) y un registro de los valores calculados recibidos por parte de todas las entidades adoptantes. Como se trata de un servicio altamente cerrado y concreto, en este caso no existe una fuerte recomendación de bases de datos no relacionales en favor de las basadas en tablas. El servicio debe ofrecer también un API para interactuar con la base de datos. Además, dicho servicio debe contar con una pequeña funcionalidad específica. Según la frecuencia de actualización del valor de referencia deseada (debe poder ser configurable), debe contener una inteligencia (en la implementación, basada en un *script*) que periódicamente interprete los valores de *cKPIs* (sin normalizar) recibidos (de todas las entidades) en dicho periodo, obtenga el valor mínimo por indicador y actualice los registros de normalización. Finalmente, se ha mencionado que una de las funcionalidades deseadas provistas por este servicio sea el establecimiento de rankings. Para ello, este servicio remoto implementa una lógica específica dentro de su API mediante la cual, ante una petición de ranking, devuelve un archivo JSON con la información del valor del CI final de un determinado periodo de todos los participantes en los despliegues. Se deja a responsabilidad de la lógica de interfaces el interpretar la información devuelta y situar a la entidad en el puesto correspondiente en el ranking. A nivel de seguridad, el desarrollador/instalador de este servicio remoto es responsable de implementar las medidas oportunas.

De forma subyacente a todo lo anterior se encuentra la necesidad de mantener la privacidad de la información compartida. Depende de qué aplicación se trate, el contenido del índice compuesto puede ser sensible (por ejemplo, impacto medioambiental o sostenibilidad energética), tratándose de una información que debe ser protegida. Además, por la experiencia del doctorando en los escenarios de validación, existe una gran reactancia a compartir este tipo de información entre compañías privadas si no es exigido [473]. Del mismo modo, atendiendo a las reglas de privacidad establecidas en la sección 3.6, ningún tipo de información personal o sensible puede ser almacenada, compartida o trazable. Por tanto, se decidió que esta comunicación debiera ser anonimizada. Para ello, se contó con la codificación en “*hashes*” [474] de la información compartida por parte de las entidades. De este modo, al enviar la información de resultados de indicadores a la entidad centralizada, el servicio utilizaría un hash de cada dato sensible (“*organization*”, “*name*” y “*type*”). Estos hashes se almacenan en un índice concreto del módulo de almacenamiento de la solución, ya que, si se realizara dentro del servicio, esta información no sería persistente (los contenedores Docker de ejecución comienzan y acaban sin guardar la información en volúmenes [475]). De este modo, el entorno remoto centralizado no podría ser capaz de inferir qué entidad ha enviado los resultados, aun pudiendo mantener un registro ordenado según el valor de dichos *hashes*. Esta aproximación resultó igualmente válida para la información de ranking, ya que cada entidad debe ser capaz de situarse a sí misma en la lista (posee información de su *hash*), pero en ningún caso puede descifrar cuál es la entidad del resto de resultados recibidos.



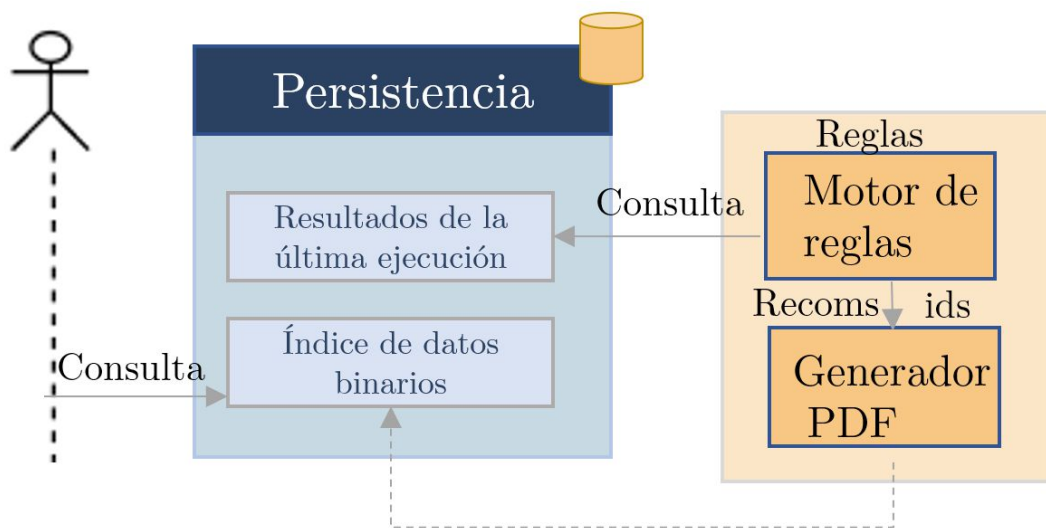
**Servicio de recomendaciones en base a reglas:**

Tal y como se expresa en los requisitos, se ha incorporado una utilidad mediante la cual se puedan establecer umbrales sobre los resultados del servicio de CI y ofrecer al usuario final una serie de recomendaciones. Estas recomendaciones están orientadas a la mejora de dichos resultados, por tanto, a la optimización del valor del índice obtenido. A efectos prácticos, estas recomendaciones se traducen en orientaciones de acciones que mejorarían la implicación de los agentes involucrados en el fenómeno representado. Si estas recomendaciones llevan a dichos agentes a implementar las medidas sugeridas, el resultado del índice mejorará (más cercano a 1).

Para plantear este servicio, se diseña un elemento software aislado (según arquitectura, a instalar como un contenedor Docker) y se establecen una serie de mecanismos y acuerdos para su funcionamiento. El objetivo es obtener un componente ligero que, a partir de la lectura de unos datos determinados (valores de cKPIs y CI resultado), aplique unas reglas y realice unas acciones (emitir recomendaciones). Para ello decidió utilizarse el software abierto llamado *EasyRules* [476], un *framework* de Java para la implementación sencilla de un motor de reglas en base a una estructura declarativa. Se escogió esta tecnología por (1) familiaridad, (2) la existencia de clases y funcionalidades ya definidas que permiten generar reglas de negocio de forma simple y aplicarlas de manera independiente o conjunta, (3) dispone de un API muy fácil de utilizar y (4) consume muy pocos recursos.

Se planteó su utilización como servicio integrado (o no) con el resto de la solución de dos posibles maneras:

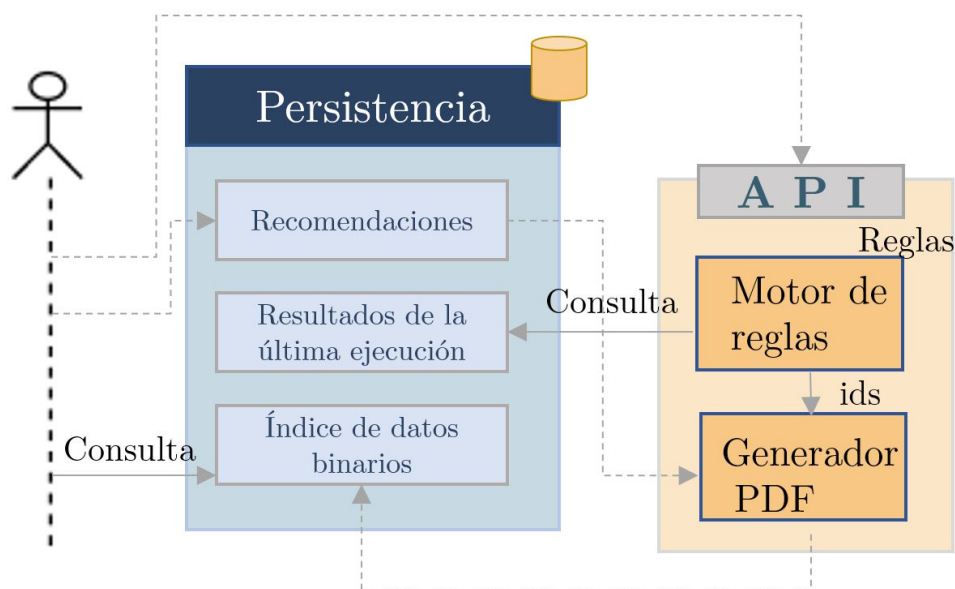
La primera (menos flexible) consiste en la inserción de las reglas (formato Java) y recomendaciones (texto en cadenas de caracteres) directamente sobre la lógica del servicio. Se trata de una implementación orientada a casos de uso de despliegue personalizado y sin perspectivas de actualización a lo largo del tiempo.



**Figura 4.45:** Servicio de recomendaciones basada en reglas sobre indicadores (I)

El servicio contiene toda la información necesaria para funcionar, siendo ejecutado periódicamente según instalación (típicamente, automáticamente después de la ejecución del servicio de CI). A partir de los resultados de la última ejecución, las reglas (fijas) generarían una serie de recomendaciones. En este punto, una herramienta Java de generación de PDFs [477] actuaría e insertaría (en formato binario, aceptado tanto por Elasticsearch como por MongoDB) la información de la última ejecución, junto con dichas recomendaciones, en un formato determinado.

La segunda aproximación planteada se centró en la flexibilidad, tratándose de un servicio más completo que el primero. En este caso, en lugar de insertar sobre la lógica del servicio (Java) directamente las recomendaciones y las reglas, las primeras serían introducidas por el usuario y almacenadas en el sistema de persistencia y las segundas se informarían al nuevo módulo a través de un API. Este API (personalizado) conseguiría la activación dinámica de reglas. El resto del funcionamiento sería equivalente al primer caso.



**Figura 4.46:** Servicio de recomendaciones basada en reglas sobre indicadores (II)

El resultado de la transformación de estos datos binarios en un PDF descargable puede observarse en la Figura 5.54: Resultado de generación de reporte con recomendaciones del PEI Este servicio auxiliar fue utilizado en dos escenarios de validación:

- PEI: Se utilizó la primera aproximación mencionada. En este caso este servicio ha sido integrado en una instalación completa de la solución dentro del proyecto de investigación PIXEL.
- OAI: Se utilizó la segunda aproximación mencionada. En este caso se utilizó este servicio de forma aislada y de manera previa al servicio de índice compuestos. No obstante, se trata de un escenario útil para demostrar la flexibilidad del planteamiento de la tesis. Esta implementación se ha realizado dentro del proyecto ASIGNAT.

### **Servicio de contextualización de índice compuesto:**

Idealmente, en un servicio de cálculo de indicadores compuestos como el planteado en esta tesis, todos los datos deberían proceder de fuentes automatizadas y precisas, como sensores, sistemas informáticos integrados o de bases de datos actualizadas o servicios remotos. Sin embargo, considerando tanto los requisitos expresados como los escenarios de validación encontrados, debe contemplarse que no todas las entidades adoptantes podrán proveer de indicadores obtenidos de forma automática (o más cercanos a tiempo real o al Internet de las Cosas). De ahí que haya que considerar un enfoque híbrido.

Como servicio adicional a los principales, se ha diseñado e implementado una metodología que contemple diversos niveles de digitalización en entidades que desplieguen la solución. En general, en la mayor parte de los sectores (concretado especialmente en esta tesis en contextos de ciudad-puerto marítimo), las empresas más grandes y con más recursos se encuentran actualmente en un nivel superior de digitalización y automatización de la adquisición de datos que aquellas pequeñas y medianas. El servicio de CI planteado resulta una herramienta flexible y escalable capaz de ser adquirida y adoptada por cualquier entidad. No obstante, esto debe ser reflejado en los resultados ofrecidos al usuario final. Para que dos valores de CI finales de dos entidades puedan ser comparados, no basta pues con un único índice final (CI), sino que se debe expresar esta disonancia en términos de adquisición de los datos iniciales (*cKPI*) para permitir la **contextualización de dicho índice**.

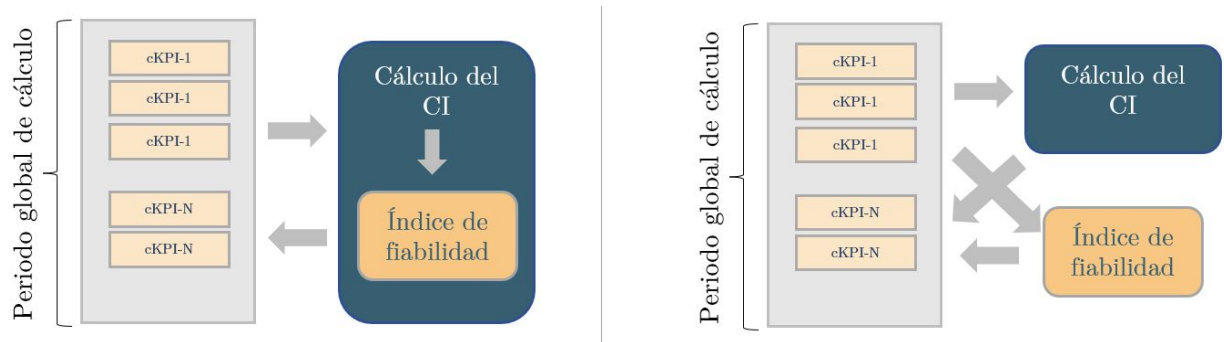
El objetivo es, por tanto, aportar información sobre el tipo de fuente de la que se han obtenido los datos y la fuente ideal de la que se podrían obtener. A nivel usuario, se muestra un resumen visual de cómo se ha realizado el cálculo en términos de captura de información, a través de un indicador adicional denominado índice de fiabilidad (se concibió, en inglés, como RR – *Reliability Rating* o *IoT-readiness level*).

Para plantear el diseño de este servicio se partió de las siguientes consideraciones:

- Cualquier fuente de datos que provea de información al cálculo del CI será interactuada por un agente (sección 3.3.1) y será introducida como un *cKPI* (individual o en combinación con otras fuentes).
- Cada agente informará de uno o más *cKPIs*. La información de las fuentes originales debe estar contenida en el modelo de datos utilizado para la introducción de estos *cKPIs*.
- Se desea obtener un único valor que exprese el nivel de fiabilidad del índice. Este valor debe ser una combinación de las características expresadas sobre la fuente de datos (por parte de cada agente) sobre los *cKPIs* que se usan para el CI.

- Por tanto, el índice de fiabilidad (RR) es, en sí, otro índice compuesto que debe calcularse utilizando un servicio análogo al principal de la tesis doctoral (véase la sección 4.2.2).

Debido a que la estructura del árbol de cálculo, el número de indicadores iniciales, el indicador final y los agentes a informar sobre las fuentes de datos serán los mismos que para el servicio principal, podría plantearse de dos formas diferentes:



**Figura 4.47:** Opciones de implementación servicio auxiliar índice de fiabilidad

- Como una lógica adicional integrada dentro del servicio original. Esto implicaría una ejecución simultánea. No se observan grandes inconvenientes más allá de que no se sigue el principio estricto de separación en microservicios.
- Como un servicio adicional a instalar aparte del principal. Dota de más flexibilidad en caso de que alguno de los dos (CI o fiabilidad) debieran modificarse o crecer en funcionalidad.

Independientemente de la modalidad escogida (ambas son aceptables), el funcionamiento del servicio auxiliar es invariable. En las siguientes páginas se recoge el diseño e implementación de dicho servicio y las consideraciones y acuerdos que debieron alcanzarse.

Cada fuente de datos es introducida, en un despliegue particular de la solución, con una forma de captura en concreto. Ésta puede ser a través de sensores, a través de APIs expuestas por un servicio externo, conectadas a un sistema informático de la entidad, a través de datos manualmente imputados (formularios, etc.) o simplemente inferida de datos de informes o disponibles en la literatura. Dependiendo de qué tipo de información se trate (por ejemplo, ruido), el abanico de opciones será más o menos amplio (en dicho caso, el ruido puede ser provisto por un sensor o por el resto de modalidades de captura de información).

El primer ejercicio consistió, entonces, en definir, para cada fuente de datos necesaria para el cálculo del CI, cuál sería su modo de captura “ideal” y compararlo con la modalidad utilizada en una ejecución del servicio determinada. Para ello, en los escenarios de validación de la arquitectura donde se desplegó este servicio (PEI -5.4- y  $\mu$ PEI -5.5), se realizó un análisis de las fuentes necesarias, cuál sería su modo

de captura ideal y cuál el realmente utilizado. En la siguiente tabla se incluye solo unas pocas filas de ejemplo de las hojas elaboradas (de mayor tamaño):

**Tabla 4.6:** Extracto de comparación entre modalidades de captura de datos

| Fuente de datos                           | Modalidad ideal | Modalidad de captura actual      |
|---|-----------------|----------------------------------|
| Ruido (dB)                                | Sensor          | Sensor                           |
| Tipo de barco                             | Servicio web    | Informe disponible en literatura |
| Tiempo de espera de camiones en el puerto | Sensor          | Servicio web                     |

En este punto, lo que se trata es de trasladar esa “diferencia operativa” entre el modo ideal de captura y el actual (por cada dato) a un valor cuantitativo. La idea subyacente es modelar con un número cuánto lejos del modo de captura óptimo se encuentra la forma de adquisición usada para una ejecución determinada. Como se ha comentado, tecnológicamente (software del servicio) se trata de otro índice compuesto, cuyo cálculo será equivalente al anterior, siguiendo las mismas configuraciones de métodos matemáticos que el servicio principal. No obstante, a nivel operativo, requiere un trabajo previo para la obtención de dicha cuantificación. Se diseñó una metodología para calcular dicha cercanía al registro ideal por cada dato necesario, que puede ser resumida en los siguientes puntos:

- Se concentraron los diferentes tipos de captura de datos en cinco posibles modalidades: (1) Sensores, (2) Servicios externos o APIs, (3) servicios internos – como modelos propios de la entidad, (4) formularios o introducción manual de la información y (5) valores estimados, medias según informes públicos o valores disponibles en la literatura.
- Cada uno de los tipos de captura tiene unas características que, en función de qué dato se trate, pueden ser consideradas como ventajas o como inconvenientes, o tendrán mayor o menor relevancia. Estas características fueron modeladas como aspectos de evaluación. Estos indicadores son híbridos: técnicos (como la precisión en la medición, el tiempo real, la disponibilidad) y operativos (como el cumplimiento de la normativa, la propensión a los fallos, la intervención humana). En el Apéndice B se muestra la reflexión realizada sobre estos aspectos y su relevancia para los tipos de captura de datos definidos.
- Para cada dato, la “forma óptima” (modalidad de captura de datos ideal) se cataloga como 100%, y el otro tipo de opciones de recuperación de datos será disminuido en percentil dependiendo de su calificación obtenida de la evaluación de la característica. Esto significa que la fuente de datos ideal tendrá un 100% y, dependiendo de la fuente de datos utilizada por la entidad adoptante, se disminuirá en cierto margen tratando de mostrar lo cerca que está del ideal.

- No obstante, al haber disgregado la evaluación en diferentes características o aspectos, la reducción de porcentaje responde a varios elementos. Para obtener un porcentaje total del tipo de captura de ese dato, se utilizó el método de asignación de presupuesto (puede encontrarse más explicación en la Tabla 5.12). Este método consistió en reunir a una serie de expertos del campo IoT y del escenario de aplicación particular (puertos marítimos en el despliegue de la tesis doctoral PEI) y encuestarles para lograr obtener una ponderación sobre cuán relevante es una característica u otra de cada fuente en el contexto de los datos necesarios para este servicio de CI. Un total de 15 (quince) expertos respondieron, y la media aritmética del presupuesto asociado a cada aspecto fue utilizada para este percentil relativo sobre la fuente ideal.

Con todo lo anterior, el resumen de la asociación de ponderación relativa de cada modalidad de captura por cada dato se ilustra en la siguiente imagen.

Este valor representa el porcentaje relativo de fiabilidad de cada modalidad de captura de datos en comparación con la ideal (primera fila).

| Captura                           | Aspecto #1 | Aspecto #2 | ... | RR agregado |
|-----------------------------------|------------|------------|-----|-------------|
| <b>Opción 1</b><br>(Sensor)       | 100%       | 100%       | ... | 100%        |
| <b>Opción 2</b><br>(Servicio/API) | 70%        | 30%        | ... | 69,23%      |
| <b>Opción 3</b><br>(Proxy)        | 25%        | 40%        | ... | 57,21%      |
| <b>Opción 4</b><br>(Manual info.) | 10%        | 25%        | ... | 30%         |
| <b>Opción 5</b><br>(Literatura)   | 5%         | 10%        | ... | 15%         |

Este valor representa cómo de relevante es esta característica (para los expertos consultados) en función de qué tipo de captura de datos se refiera.

**Figura 4.48:** Metodología de cálculo del porcentaje relativo de fiabilidad de datos

Como resultado, cada fuente de datos tendrá asociado un porcentaje correspondiente a todas las modalidades de introducción. Esto debe realizarse, en cada escenario, para cada dato.

Como esta base de conocimiento, puede ser de un tamaño considerable (en escenarios con muchos datos de entrada), esta información debe ser insertada en el módulo de persistencia de la solución en un formato determinado, accesible para el servicio de índice de fiabilidad. Se llegó al acuerdo de ser insertada como una serie de archivos JSON con el formato que se expone en la siguiente figura.

El ejemplo mostrado corresponde a un registro que existiría en el escenario de validación PEI (sección 5.4) asociado a la fuente de datos: *información del barco (IMO y carga procesada)*, cuya fuente de captura ideal es a través de un API (100%) y el resto de valores relativos se encuentran indicados siguiendo la nomenclatura:

- Sensores (no se contempla en este ejemplo): *sensor*.
- Servicio externo o API. Se dividió en dos: en tiempo real (*realTimeAPI*) o periódico (*periodicAPI*).
- A través de servicios integrados o con herramientas informáticas intermedias. Se dividió en dos: *ProxyTool* y *externalData*.
- Formularios (*handtypedForms*) o introducción manual de la información (*manualUploadToServer*).
- Estimación o valor extraído de referencias (*averageValueFromLiterature*).

```
{
  "_index": "ckpi-config-rr",
  "_type": "_doc",
  "_id": "1M5X1nkB9zN0nbt0fvBP",
  "_version": 1,
  "_score": 0,
  "_source": {
    "dataPiece": "IMO number and gross tonnage of ships",
    "optimal": "realTimeAPI",
    "realTimeAPI": 1,
    "periodicAPI": 0.8056,
    "ProxyTool": 0.8611,
    "externalData": 0.75,
    "manualUploadToServer": 0.6111,
    "handtypedForms": 0.111,
    "averageValueFromLiterature": 0.2778
  }
}
```

**Figura 4.49:** Ejemplo del formato de la fiabilidad relativa de fuentes de datos

Este conocimiento debe haberse introducido con anterioridad en el módulo de persistencia por parte del desarrollador/instalador y depende en gran medida del sector de aplicación del despliegue. En concreto, para que este servicio pueda tener lugar, deben introducirse dos archivos de conocimiento en el módulo de persistencia:

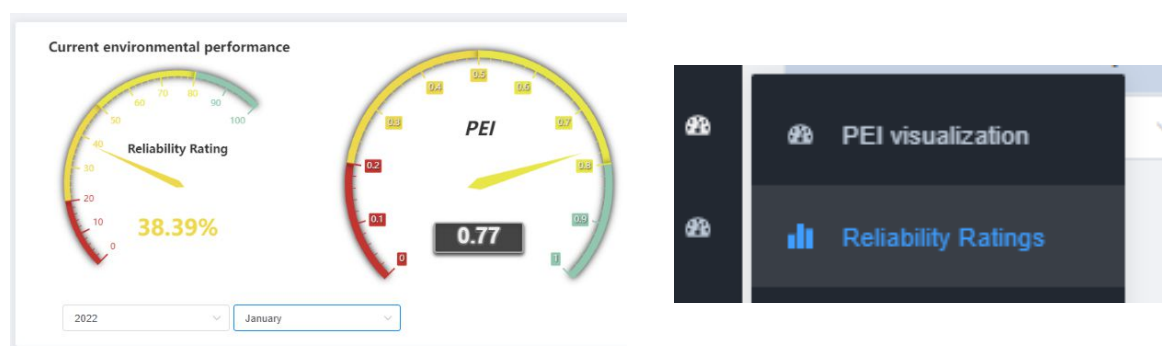


- Árbol para la ejecución del índice de fiabilidad. Esta información, que se almacenará en el índice *ckpi-config-rr*, debe corresponderse con el árbol establecido para el servicio. No obstante, pueden indicarse métodos de agregación y ponderación alternativos. En particular, se define que la agregación entre nodos sea aritmética en todo caso.
- Base de conocimiento de los porcentajes de fiabilidad relativos por cada fuente de datos. Se tratará del almacenamiento (en el mismo índice indicado) de tantos JSON como el ejemplo de la Figura 4.49: Ejemplo del formato de la fiabilidad relativa de fuentes de datos como fuentes de datos existan para el despliegue particular.

En relación a la ejecución del servicio, una vez dichos datos están disponibles en el componente de persistencia, seguirá el siguiente flujo:

- 1) Los agentes capturan la información de las fuentes de datos para introducirlas al cálculo del CI (véase 3.3.1). Dentro del formato del modelo enviado al gestor de contexto, se utiliza el campo *dataPieces* (Tabla 4.5) para indicar qué modalidad de captura se utiliza (usando los códigos mencionados en la página anterior).
- 2) Al ejecutarse el servicio (conjuntamente o separado del principal), se recuperan los datos del módulo de persistencia del árbol del índice de fiabilidad y de los porcentajes relativos.
- 3) Se ejecuta un servicio de CI (sección 4.2.2) con dichos datos de configuración, obteniendo un valor de fiabilidad (RR) de cada nodo del árbol y, finalmente, del nodo raíz final.

Tras el proceso anterior, se obtendría un CI final:  $RR_{CI}$ , siendo un único indicador con un valor entre 0 y 1 acompañando al valor del CI del servicio principal. Según el planteamiento de la solución, se incorporó un elemento en el entorno de visualización para que esta información pudiera visualizarse desde el cuadro de mando de la lógica de interfaces. Principalmente, se muestra el resultado como un velocímetro más pequeño al lado del CI principal. Además, se introdujo un panel adicional en un espacio separado dentro del menú de la interfaz para aportar un mayor detalle sobre dichos valores.



**Figura 4.50:** Interfaz de resultados del índice de fiabilidad (I)



Para cada dato, existe una forma de recuperación óptima y actual, con una calificación independiente y otra agregada (para los subíndices y el origen). Esta información es la que se muestra en dicha interfaz mostrada, junto con los resultados parciales y agregados de fiabilidad. Además, permite acceder a información específica y detallada sobre las entradas de datos para cada *cKPI*.

| Data origin    | Subindex     | Piece of data                         | Optimal retrieval way | Current retrieval way         | Reliability Rating | Aggregated RR (subindex) | Aggregated RR (origin) |         |
|----------------|--------------|---------------------------------------|-----------------------|-------------------------------|--------------------|--------------------------|------------------------|---------|
| Ships          |              | IMO number and gross tonnage of ships | Real time API         | Periodic API                  | 80.56 %            | 67.96 %                  | 83.47 %                |         |
|                | Air Emission | Main and auxiliary engine of ships    | Periodic API          | Average value from literature | 36.84 %            |                          |                        |         |
|                |              | Berth and maneuvering time of ships   | Sensors               | Periodic API                  | 86.49 %            |                          |                        |         |
|                | Waste        | MARPOL annexes about waste            | Pixel proxy tool      | Periodic API                  | 97.14 %            |                          |                        | 97.14 % |
|                | Wastewater   | MARPOL annexes about wastewater       | Pixel proxy tool      | Periodic API                  | 85.29 %            |                          |                        | 85.29 % |
| Terminal       | Wastewater   | Wastewater produced by the terminals  | Pixel proxy tool      | Handtyped forms               | 21.95 %            | 21.95 %                  | 21.95 %                |         |
| Port authority | Wastewater   | Wastewater produced by the PA         | Pixel proxy tool      | Handtyped forms               | 9.76 %             | 9.76 %                   | 9.76 %                 |         |

2022 | January

**Figura 4.51:** Interfaz de resultados del índice de fiabilidad (II)

En este punto, todas las consideraciones previas sobre la capacidad de comparación, la cercanía a la fuente de datos ideal y la evolución del CI con el tiempo se representarán mediante componentes gráficos.



# Capítulo 5

## Aplicación en entornos reales

### 5.1.Introducción

Esta sección evidencia la fase de experimentación de la tesis doctoral. En particular, se centra en exponer los resultados del despliegue de la solución en diversos casos de uso. Como se ha comentado, la arquitectura y el servicio diseñados han sido planteados de manera agnóstica, es decir, de forma generalista para poder ser aplicados a diferentes sectores y casuísticas. Es por ello que, para validar las bondades de la solución, debieron escogerse varios escenarios que contribuyeran a confirmar el cumplimiento de los requisitos (véase sección 3.1.1).

Asimismo, la arquitectura propuesta se compone de varios módulos; algunos de ellos considerados estructurales (esenciales) y otros de ellos optativos (dotando de determinadas características/funcionalidades al despliegue en cuestión). Por esta razón, se creyó conveniente encontrar diferentes aplicaciones de la solución para

demostrar la capacidad de flexibilidad de la arquitectura y para validar la composición modular.

El candidato ha tenido la oportunidad de participar en varios proyectos de investigación durante los tres años de desarrollo de la tesis, donde pudieron plantearse escenarios de validación de (al menos una parte) de la solución. Así, este capítulo trata de contextualizar el despliegue de esta arquitectura en los escenarios de validación de los proyectos de investigación, indicando el rol ejercido y las peculiaridades de cada caso que han permitido efectivamente confirmar la hipótesis inicial y cumplir con los objetivos de la tesis.

Las consiguientes subsecciones (secciones 5.3 a 5.7) detallan el proceso llevado a cabo en cada uno de ellos, exponiendo en primer lugar el contexto en el que se plantearon, los datos disponibles y la definición del índice objetivo, pormenorizando posteriormente el despliegue tecnológico y los resultados y visualización, correspondientemente.

No obstante, previo a la explicación de cada uno de los escenarios de manera detallada, se hace necesario realizar una introducción de los mismos, tratando de ofrecer una visión global de conjunto. Como puede observarse en la lista inferior, han escogido casos de uso mayormente relacionados con entornos ciudad-puerto o marítimo portuarios, dado al gran enfoque sobre este sector por parte del candidato durante su participación en los proyectos. La lista de los cinco casos de uso, reconocidos por el índice diseñado, se expone a continuación:

- **TCI – *Traffic Congestion Index* - Instanciación cloud en ciudad-puerto para análisis de congestión de tráfico:** El escenario de validación más ambicioso que se planteó para la validación de la arquitectura. Se trata de un caso de uso enmarcado dentro de la monitorización de la ocupación de las puertas de acceso al puerto de Tesalónica basado en observación y predicciones de valores asociados al proceso (tráfico en calles aledañas, próximos atraques de buques en el puerto, condiciones meteorológicas, históricos...). Este experimento fue desarrollado extensivamente e incluye la utilización de los módulos de predicción y entrenamiento de series temporales, validando una de las características más innovadoras de la arquitectura. Este caso de uso se considera el más completo y es analizado con más profundidad en las siguientes subsecciones, incluyendo un análisis exploratorio de los datos utilizados.
- **PEI – *Port Environmental Index* - Instanciación cloud en puertos marítimos para análisis de impacto medioambiental:** Se trata del principal escenario de validación, que sirvió como punto de partida para los otros cuatro casos de uso. Este caso de uso se centra en la creación y cálculo en cuatro puertos marítimos europeos de un índice para el análisis de su impacto medioambiental. Tecnológicamente, se trata de un despliegue cloud (en la nube o en instalaciones de servidor centralizado del puerto), y su ejecución se enmarca dentro del proyecto PIXEL. La aportación del

candidato fue extensiva y sostenida durante un largo periodo de tiempo. Se evidencia en la subsección 5.3.

- **μPEI - micro PEI - Instanciación *edge* para la evaluación rápida de impacto medioambiental a través de un kit IoT:** Se trata de una versión simplificada del primer caso de uso (PEI) orientada a una ejecución en tiempo real y en entornos de computación de borde (*edge computing*). El objetivo es demostrar que el sistema planteado puede ser miniaturizado en términos de computación ubicua, permitiendo su aplicación como un “*kit*” rápido para despliegues ágiles; sin necesidad de un entorno centralizado de computación tipo *cloud*. El candidato planteó y desarrolló este caso de uso, incluyendo una demostración en tiempo real que se adjunta como evidencia.
- **CEDI – *Composed Estimated Density Indicator* - Instanciación cloud en un entorno de terminal logística para predicción de distancia social en época de COVID-19:** Aplicación de la arquitectura y del servicio diseñado en la tesis doctoral para cubrir un caso de uso particular en una terminal portuaria: la estimación de la densidad de trabajadores en un espacio determinado de la terminal de contenedores en función de los trabajos esperados. En contexto de las restricciones temporales de mantenimiento de la distancia social, fue requerida la creación de este piloto dentro de un proyecto de investigación, lo que valió a la solución propuesta en la tesis de validar su capacidad de desplegarse de forma ágil y rápida en un nuevo escenario. Tecnológicamente, este caso de uso aporta la integración con una herramienta de simulación de ocupación de áreas en una terminal. El candidato fue el encargado de diseñar el sistema y de colaborar en su puesta en marcha.
- **OAI – *Optimal Assignment Indicator* - Instanciación híbrida para la recomendación de asignación en transporte logístico por carretera.** Este escenario de validación se centra en la recomendación de asignación del tándem conductor-camión en una orden de transporte de un contenedor de mercancía desde/hacia el puerto de Valencia. En particular, trata de ofrecer un valor sobre cuán óptima es una selección a través del índice compuesto. Tecnológicamente, este caso de uso se enmarca en una acción más amplia orientada a la creación de un Gemelo Digital. A pesar de que este caso de uso incluye predicción de serie temporal de dicho índice, ésta se realiza por otro sistema (externo a esta arquitectura), demostrando la flexibilidad y capacidad de adaptación de la solución. Conviene destacar (véase la figura anterior) que este escenario de validación aún no ha sido completamente finalizado. El motivo es que se encuentra enmarcado en el proyecto de investigación ASIGNAT, aún en fases iniciales/intermedias de su ejecución (finalización esperada: octubre 2023).

Cada uno de los escenarios elegidos posee algún elemento diferenciador que ha permitido la validación de un módulo/característica especial de la arquitectura

planteada. Con la intención de ilustrar este hecho, la siguiente tabla trata de ofrecer una visión general de las características del Internet de las Cosas de Nueva Generación cubiertas en cada uno de los índices (escenarios) anteriores:

**Tabla 5.1:** Escenarios de validación de la arquitectura propuesta

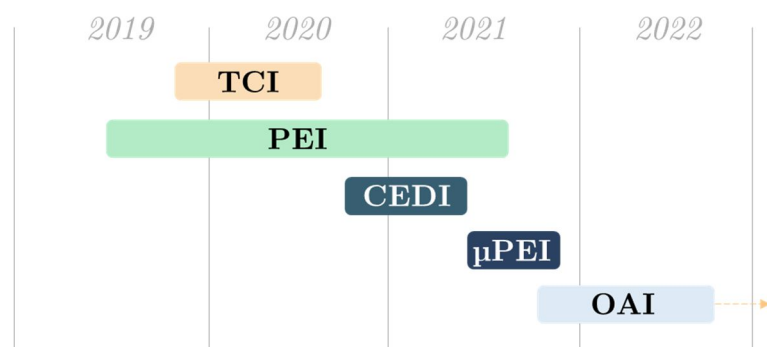
|           | Elementos esenciales de la solución | Ránking o índice de fiabilidad | Predicción de índices | Edge computing / despliegue ágil | Combinación con otros servicios |
|-----------|-------------------------------------|--------------------------------|-----------------------|----------------------------------|---------------------------------|
| TCI       | X                                   | X                              | X                     |                                  | X                               |
| PEI       | X                                   | X                              |                       |                                  | X                               |
| $\mu$ PEI | X                                   | X                              |                       | X                                |                                 |
| CEDI      | X                                   |                                |                       | X                                | X                               |
| OAI       | X                                   |                                |                       |                                  | X                               |

Del mismo modo, una arquitectura IoT no puede ser concebida sin su validación con fuentes de datos que se conectan usando diversos protocolos de comunicación. En este sentido, la selección de los escenarios de validación comentados cubren las siguientes casuísticas:

**Tabla 5.2:** Acceso a datos IoT cubierto en los casos de uso

|           | HTTP | MQTT | GPS | FTP | RFID |
|-----------|------|------|-----|-----|------|
| TCI       | X    |      |     |     | X    |
| PEI       | X    | X    |     | X   |      |
| $\mu$ PEI | X    | X    |     | X   | X    |
| CEDI      | X    |      |     |     |      |
| OAI       | X    |      | X   |     |      |

Por último, en aras de ilustrar las posibles diferencias de profundidad entre uno y otro, se expone en la siguiente imagen la temporalidad de cada uno de los escenarios de validación:



**Figura 5.1:** Temporalidad de los escenarios de validación

En la sección siguiente se expone la metodología seguida para el despliegue de la arquitectura y la implementación de la tesis doctoral en dichos escenarios.

## 5.2. Metodología de aplicación y despliegue de la solución

Para llevar a cabo las validaciones de la solución propuesta en la tesis, se definió y se llevó a cabo una metodología de varios pasos. Las explicaciones que se ofrecen a continuación constituyen una guía de uso para cualquier persona que desee reproducir un caso práctico similar utilizando la tecnología desarrollada en el marco de esta tesis.

Como se ha mencionado previamente, la tesis doctoral puede dividirse en dos grandes bloques (de arquitectura IoT y de servicio). Por tanto, la metodología diseñada para su despliegue se estructuró con estas premisas en mente.

### 5.2.1. Visión general

Para comenzar, cada despliegue de la solución propuesta debe plantearse las siguientes preguntas: ¿qué tipo de sistema se busca – *edge*, *cloud* o híbrido?, ¿qué equipamiento debe ser necesario para soportar el sistema?, ¿se dispone de los datos para emprender el despliegue? Las soluciones a estas preguntas deberían poder responderse analizando las secciones 2.3 y 3.7.

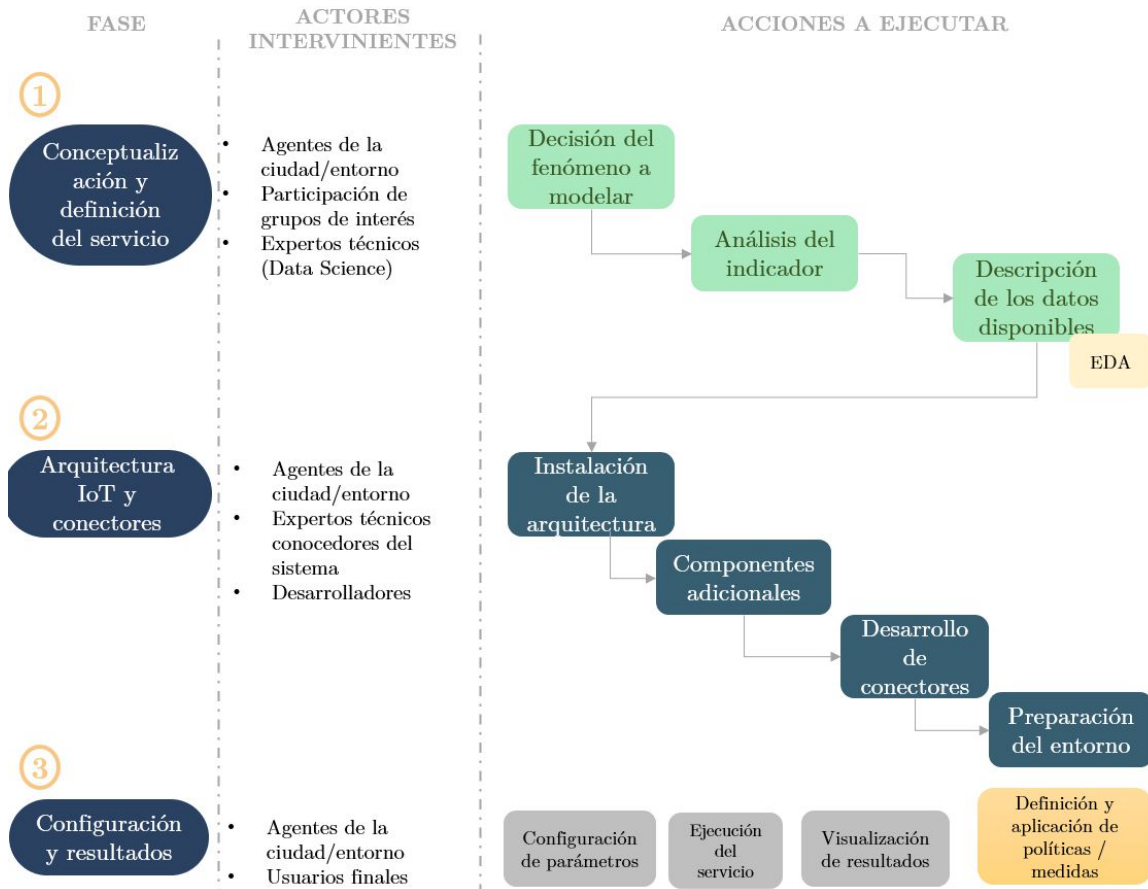


Figura 5.2: Fases de la metodología de aplicación y despliegue de la solución

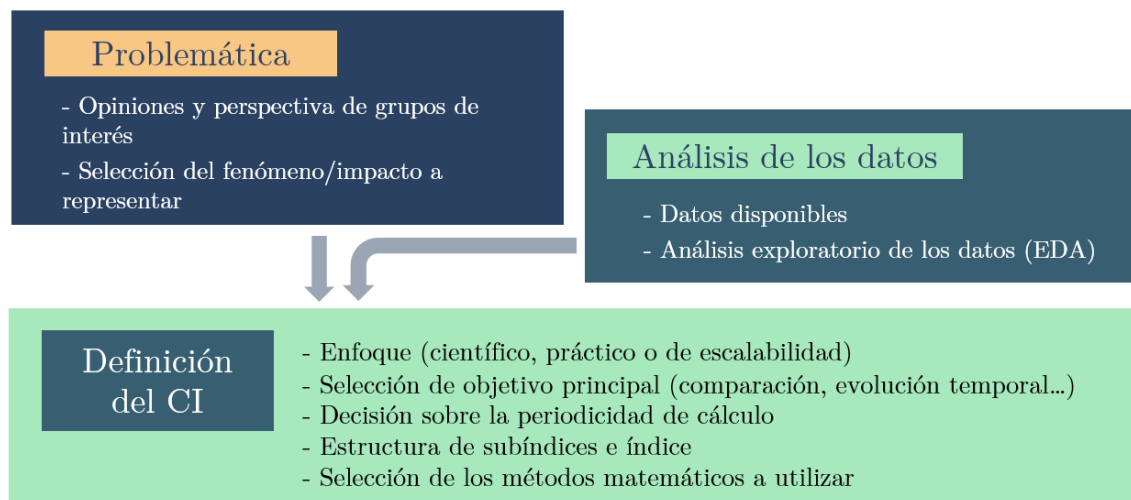
Como se observa en la figura anterior, una vez aclarado lo anterior, la primera fase de la metodología corresponde a la parte conceptual del bloque de servicio, y está compuesta por las acciones que deben ser abordadas por las personas implicadas en la operativa a tratar (puerto, ciudad, ciudadanos, agentes sociales...). El objetivo es encontrar el indicador compuesto (CI) útil que pueda llevar a dichos actores relevantes a establecer políticas sostenibles e inteligentes en función del análisis de dicho IC. Además, se deben definir los subíndices que lo forman, qué datos utilizar para llegar hasta ellos y algunos parámetros tan relevantes como la normalización o agregación a utilizar (véase sección 4). Por lo tanto, esta actividad es de suma importancia. A continuación, la arquitectura IoT (los diferentes módulos expuestos en la sección 3- integrados según la sección 3.7) debe ser instalada y customizada para la aplicación en cuestión (recuérdese, arquitectura modular), incluyendo el desarrollo de los conectores de datos suficientes para poblar la solución IoT y la puesta en marcha del módulo de servicio de índice compuesto. Por último, una integración de ambas fases tiene lugar mediante la comprobación de los resultados y, si es necesario, un ajuste fino de las configuraciones del servicio y de la arquitectura.

Conviene recordar que esta metodología sufrió cambios y mejoras a lo largo de su utilización, y entre un caso de uso y otro. Los pasos detallados en esta sección se corresponden a la última versión de la misma.

En las siguientes subsecciones, se describen los detalles de cada una de las tres fases del proceso:

### 5.2.2. Fase I – Conceptual y de servicio

Esta fase consta de tres pasos, como se muestra en la siguiente figura. El diagrama muestra las acciones correspondientes y quién se ha encargado de ellas en los escenarios de validación. En los párrafos posteriores se expone un detalle de cada una de las principales acciones.



**Figura 5.3:** Actividades de la fase I en la metodología de despliegue



### **Identificación de la problemática y de los datos necesarios:**

No se realiza un análisis exhaustivo de esta fase. Para los escenarios de validación de la tesis, se tuvo la suerte de contar con actores interesados que, en colaboración con diversos agentes, definieron la necesidad de los indicadores, así como potenciales acciones a realizar sobre ellos. Similarmente, el análisis de datos se debe realizar partiendo del conocimiento del sector y de la operativa de cada entidad. En ocasiones, el conocimiento técnico de fuentes de datos abiertas por parte del candidato (u otros especialistas) ayudó a identificar fuentes como predicción meteorológica, datos de tráfico de las ciudades, etc.

### **Definición del indicador compuesto:**

La elección de los indicadores es un aspecto ciertamente complicado, que tiene una influencia directa en la universalidad de las medidas a aplicar y en la posibilidad de comparar entidades a partir de dicho índice. En este sentido, la sección 4 ha expuesto una serie de ejemplos y procedimientos que deben guiar esta actividad. Como referencias generales, en los escenarios de validación planteados en esta tesis esta definición se abordó desde tres perspectivas:

- Enfoque “científico”, que consiste en la lectura de artículos científicos e informes de investigación relacionados con el tema y en la selección de los más afines y relevantes para la definición del indicador. Además, la frecuencia con la que se describen los distintos aspectos influyentes como “importantes” o “significativos” en esos documentos debe tener importancia en la selección del índice (por ejemplo, no todos los factores medioambientales tienen la misma relevancia, e incluso algunos no son mencionados en una gran cantidad de referencias en la literatura).
- Enfoque “práctico”, que consiste en recoger información de diferentes compañías/entidades pertenecientes a un sector para sondear la relevancia de diferentes aspectos relacionados con el índice. En los escenarios de validación expuestos, esta recogida de información fue llevada a cabo a partir de reuniones con actores interesados o mediante encuestas. Estas respuestas se examinaron cuidadosamente y, junto con la información encontrada en la literatura, constituyeron la base para la decisión final sobre el índice a establecer, la importancia de los diferentes datos y factores que lo componen.
- Enfoque de “escalabilidad”, que consiste en la realización de un ejercicio de prospectiva: cómo deben representarse las funcionalidades diseñadas para sobre el CI de forma que sean lo más útiles posible para la entidad que va a adoptar la solución y para futuras actividades. Una parte de bloques principales) del software del servicio CIC.

En este momento también debería decidirse cuál va a ser la periodicidad de ejecución del servicio deseada (diaria, semanal, mensual...). Como se expone en la sección 4, el CIC está pensado para ser un servicio eminentemente retrospectivo. Aunque la solución es perfectamente viable para un caso de tiempo real (véase sección 5.5), el caso de uso habitual contará con una periodicidad determinada. Esta decisión se encuentra condicionada a la frecuencia de muestreo más baja de

cualquiera de los datos. Si uno de los valores que alimentan el índice solo se actualiza una vez al mes, y se asume que no puede darse validez a un valor que desprecie dicha entrada, entonces esta periodicidad (un mes) será la más limitante y la que fije la frecuencia de cálculo del indicador compuesto.

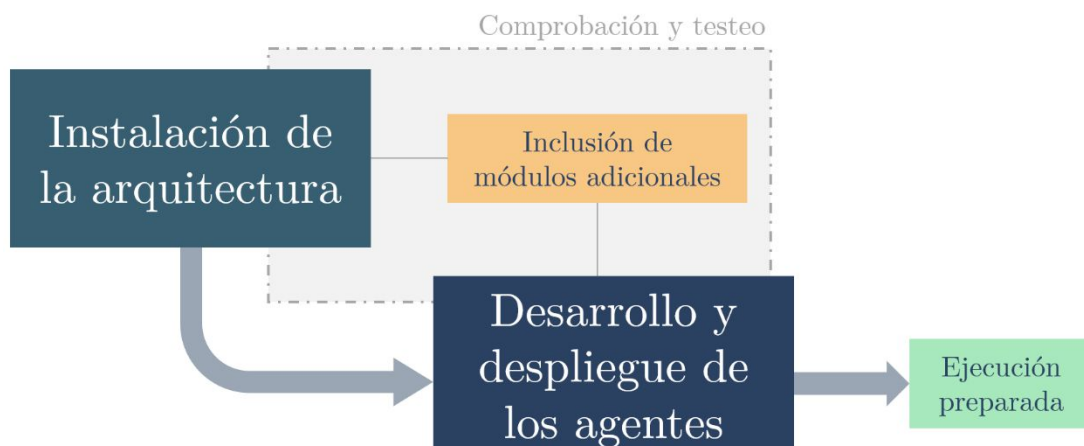
### **Estudio de los datos para la selección de los parámetros de ejecución del servicio:**

Una vez clarificados los índices, subíndices y los datos que alimentarán el índice, se deben abordar dos tareas para finalizar el diseño del servicio del cálculo del índice compuesto:

1. Estudio exploratorio de los datos. Esta acción puede realizarse analizando los datos con métodos software automatizados (ver sección 4.1.2). El objetivo es analizar la correlación entre los diferentes datos y su potencial influencia sobre el índice. De esta manera, se dispondrá de una información relevante para seleccionar el método de normalización (p.ej., min-max), podrán diseñarse filtros previos (p.ej., eliminación del 5% de valores superiores e inferiores) y ayudará a establecer una estrategia de ponderación de los diferentes datos y subíndices.
2. Selección de los métodos matemáticos más adecuados para cada caso de uso. En este sentido, en la sección 4.1.2 se pueden encontrar una serie de directrices para guiar estas decisiones, tanto para normalización como ponderación, agregación u otros tratamientos de los datos.

## **5.2.3. Fase II – Arquitectura IoT y conectores**

Esta fase comprende todas las acciones para desplegar la solución en el entorno de aplicación. Se trata de la fase más tecnológica y en la que más hincapié se ha puesto tanto en el diseño como en su ejecución en los escenarios de validación. La fase II se compone de dos grandes bloques, que a su vez se descomponen en subtareas, como puede observarse en la Figura 5.4.



**Figura 5.4:** Fases de la metodología de aplicación y despliegue de la solución

### **Instalación de la arquitectura:**

En primer lugar, se deben instalar los módulos esenciales de la arquitectura. Para hacerlo, se ha preparado un *Docker compose* [478] incluyendo las imágenes de los diferentes componentes básicos de la arquitectura. Este archivo *.yml* se adjunta como Apéndice C de esta tesis doctoral. En él se pueden apreciar los diferentes elementos estructurales de la arquitectura.

Posteriormente, se debe proceder a la comprobación de la instalación correcta de los componentes, comenzando con el gestor de contexto (ORION). Para ello puede usarse la API expuesta por el componente y realizar consultas básicas. Debe realizarse comprobaciones equivalentes con la base de datos de persistencia. Se recomienda consultar la interfaz *Kibana* (debe encontrarse en el puerto 5601). Seguidamente, debería realizarse una introducción de dato de prueba, insertando un dato a través del gestor de contexto sobre una entidad inexistente. Si el sistema se ha instalado correctamente, el dato debería estar registrado en un índice del entorno de persistencia (*ElasticSearch*), visible desde la interfaz accedida.

Después ha de validarse la correcta instalación del servicio que se ha desarrollado en la tesis doctoral (CIC). Para ello, basta con un análisis de las imágenes instaladas y la comprobación del script planificado periódicamente (haciendo uso de la utilidad *crontab*).

Evidentemente, la interfaz de visualización desarrollada para el servicio debe estar del mismo modo disponible, en este caso bastaría con acceder (usando el nombre otorgado) a la dirección del servicio, accediendo por el puerto web.

Por último, si se trata de un despliegue en un caso de uso que cuenta con los módulos de entrenamiento e inferencia de series temporales para la predicción del índice compuesto, deberá realizarse una comprobación adicional. Usando el comando “**docker ps -a**”, se debe constatar la existencia de los contenedores de dichos módulos. Esto aplica en caso de que otros módulos adicionales hubieran sido incluidos en la instalación.

Una vez que los puntos previos hayan sido correctamente instalados y validados, la instalación de la arquitectura puede considerarse “finalizada”.

### **Despliegue de los agentes**

Una vez asegurado el estado anterior, es el momento de conectar los datos (que proveerán de información al índice) con la arquitectura. Para ello, se recomienda la utilización del framework *pyngsi* (no desarrollado por el candidato). Este framework, implementado en Python y concebido dentro del seno del proyecto PIXEL (público a través de PyPI), tiene el fin de simplificar desarrollo de dichos conectores (llamados “agentes”). En relación a su utilización en los escenarios de validación de la tesis, este último ha sido el caso predominante. Conviene destacar que esto es únicamente una recomendación, un agente puede desarrollarse de diferentes maneras y con multitud de tecnologías/lenguajes de programación.

En este sentido, aunque *pyngsi* facilita la tarea (una serie de plantillas y código pre-generado están disponibles), completar un agente es una actividad que requiere mucho tiempo y es un reto técnico. Además, difiere de un caso a otro, por lo que la replicabilidad es posible pero no inmediata. Se ha de tener en cuenta que un índice compuesto puede necesitar una gran cantidad de fuentes de información para aglutinar todo el conocimiento en un único valor. A continuación se muestran los agentes que tuvieron que desarrollarse dentro de cada uno de los casos de uso para poder llevar a cabo las validaciones de la solución:

**Tabla 5.3:** Agentes NGSi desarrollados en los escenarios de validación

|     | Número de agentes |  | Número de agentes |
|-----|-------------------|--|-------------------|
| PEI | 25*               | $\mu$ PEI  | 4                 |
| EDI | 4                 | OAI  | -                 |
| TCI | 4                 | * De los cuales, 8 fueron desarrollados por el candidato |                   |

Una herramienta muy útil, que acompaña a la librería *pyngsi*, es el manual completo sobre el desarrollo de agentes NGSi (realizado por los socios del proyecto PIXEL). El llamado “*cookbook*” de los agentes NGSi se encuentra disponible online y ha sido un apoyo fundamental para permitir la ejecución de los casos de validación de la tesis doctoral.

Como se ha detallado en la sección 4.2.1, el servicio para el cual se ha diseñado toda la solución propone como medida la confluencia de todos los datos a un modelo de datos específico (*cKPI*). Por tanto, no se hace hincapié en esta fase del despliegue en la descripción de posibles modelos de datos (*data models*) a especificar como formato resultado del agente. No obstante, conviene mencionar que la arquitectura acepta entradas con formato sintáctico diferente. Sensores que precisaran de un agente directo incompatible con este modelo, podrían proveer de datos a la arquitectura; siendo necesario un agente intermedio conversor (o puente). Este hecho se encuentra más detallado en la sección 4.2.1.

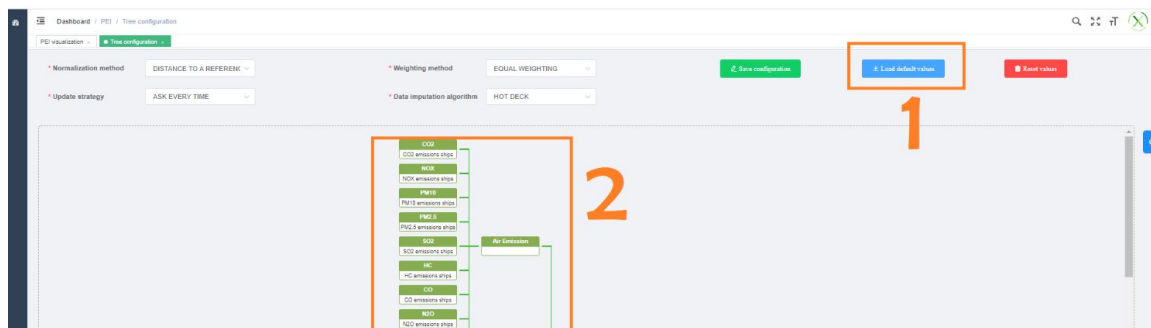
Según buenas prácticas de instalación, tras este desarrollo, el agente debería ejecutarse (“**docker run**”) y, tras ello, debería ejecutarse un conjunto de pruebas unitarias. Si la ejecución finaliza sin errores, los resultados obtenidos deben ser observados y seguidos para comprobar el correcto funcionamiento del agente. Todos los agentes desplegados en los escenarios de validación superaron dichas comprobaciones, primero en un entorno local (sin desplegar sobre la arquitectura) y seguidamente integrados.

## 5.2.4. Fase III – Configuración y resultados

Por último, para que el servicio de cálculo (y predicción) de índices compuestos pueda ejecutarse sobre la arquitectura, deben configurarse una serie de parámetros de antemano. Para ello, debido a que el sistema ya se encuentra funcionando, se hace uso de la interfaz desarrollada para el servicio (véase sección 3.5).

En primer lugar, partiendo de la estructura de índices y subíndices definida en la fase I, se debe dar de alta el árbol (*Tree*) que guiará la composición de los cálculos. Como se definió anteriormente, el cálculo de un índice puede entenderse como una estructura de nodos, partiendo de nodos hoja y finalizando en un nodo raíz. Esta comparación se ha traducido a una implementación de interfaz que permite al usuario definir dicha estructura.

Se debe, primeramente, cargar los valores por defecto (1 en Figura 5.5), lo que genera una estructura predeterminada. A partir de la misma, el usuario puede añadir y eliminar nodos, cambiar sus nombres y generar “hijos” hasta alcanzar la disposición deseada (2 en Figura 5.5). En las siguientes subsecciones en la sección 5 se ilustra el resultado de esta configuración para cada caso de uso. Conviene mencionar que esta misma estructura exacta se utilizará para efectuar el cálculo del índice de fiabilidad del CI.



**Figura 5.5:** Composición del árbol para el CI en los escenarios de aplicación

Posteriormente, partiendo de los métodos matemáticos definidos tras el análisis de los datos y la mejor opción según el caso, se debe emplear los desplegables seleccionables de la interfaz para indicar dichas decisiones. Los valores establecidos por defecto son:

1. **Método de normalización:** Distancia a un punto de referencia (*distance to a reference*).
2. **Estrategia de actualización de datos:** Preguntar cada vez (*ask every time*).
3. **Método de ponderación:** Igualdad de pesos (*equal weighting*).

#### 4. Algoritmo de imputación de datos faltantes: *Hot deck*.



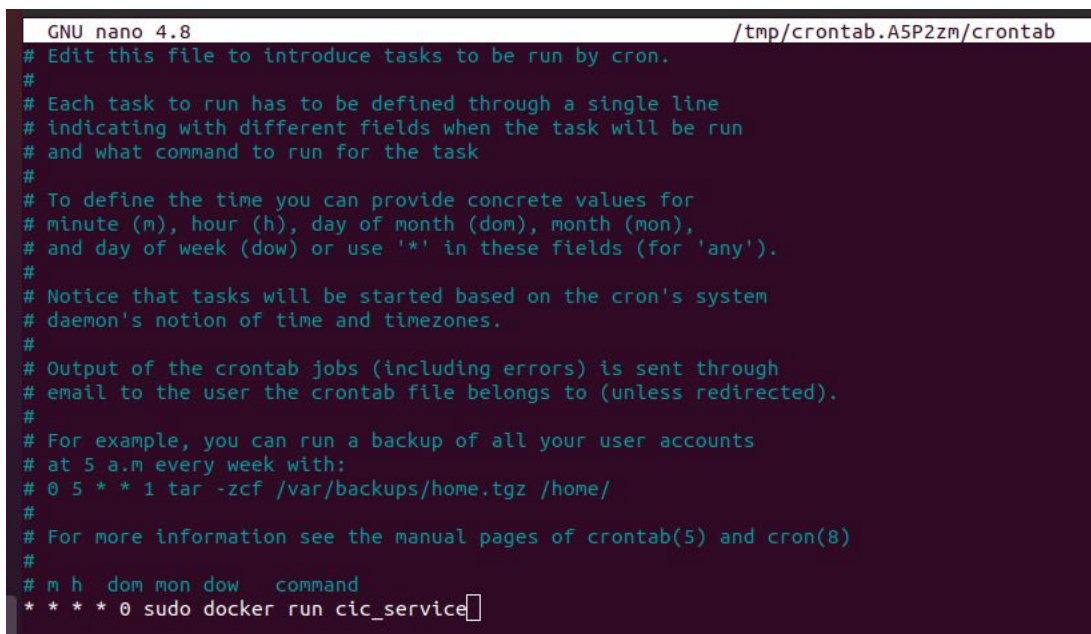
**Figura 5.6:** Selección de métodos matemáticos en la interfaz

Por último, partiendo igualmente de las decisiones tomadas en la primera fase, se debe programar la temporalidad con la que se ejecutará el servicio desarrollado. Para ello, se hace uso de la herramienta *crontab* de Linux.

En concreto, debe encontrarse la línea en la que se especifica el servicio CIC y modificar los primeros caracteres para especificar la periodicidad deseada, siguiendo la siguiente nomenclatura:

```
* * * * * (comando a ejecutar)
1º * : día de la semana (0 a 7)
2º * : mes (1-12)
3º * : hora (0-23)
4º * : minuto (0-59)
```

Como resultado, el servicio se ejecutará cada cierto tiempo y almacenará los resultados. Como puede observarse, la frecuencia mínima indicable es 1 minuto.



**Figura 5.7:** Configuración de periodicidad de ejecución del servicio

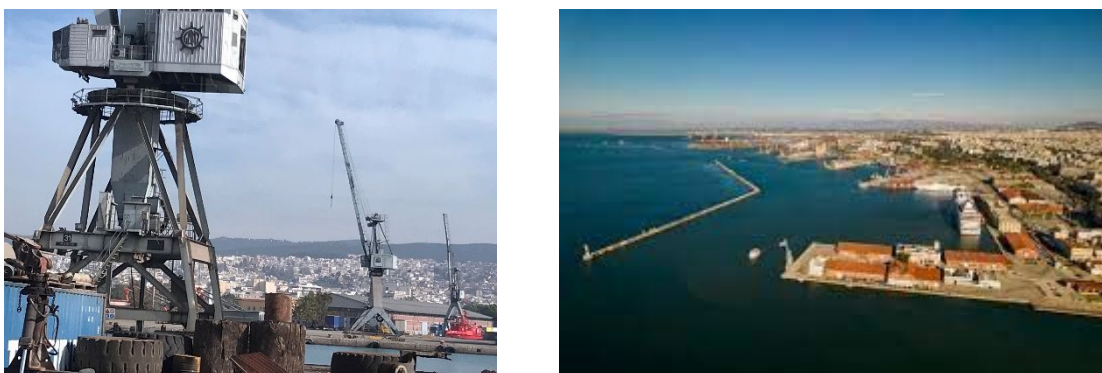
A raíz de lo indicado en la sección 3.4.2, si un despliegue desea obtener un índice con una frecuencia mayor, deberá instalarse un servicio alternativo. Como se trata del caso menos habitual, se indica como opción en la metodología global de aplicación y despliegue.

## 5.3. Instanciación *cloud* en ciudad-puerto para análisis de congestión de tráfico

Este caso de uso se centra en la monitorización y predicción de la congestión a las puertas de acceso del puerto de Tesalónica. Se trata del escenario de validación más importante de la solución desarrollada en la tesis doctoral. Esto se debe a que se ha desplegado la totalidad de los módulos planteados, incluyendo el entrenamiento y la predicción de los índices compuestos. Además, ha sido uno de los escenarios que más se ha prolongado en el tiempo, por lo que ha permitido una ejecución pormenorizada de todos los pasos expuestos en la metodología (sección 5.2.2). Ha sido, pues, la oportunidad perfecta para validar la solución de una manera holística. Todos los pasos han sido debidamente documentados y se exponen en las siguientes subsecciones:

### 5.3.1. Contexto

El puerto de Tesalónica está situado muy cerca del centro de la ciudad, que a lo largo de los años se ha construido en torno a las oportunidades económicas que brinda un puerto. En los últimos diez años, un gran número de actividades de diversa índole se han trasladado a los alrededores del puerto, que ha sido seleccionado cada vez más como el lugar ideal de establecimiento de empresas de servicios (con oficinas). Además, varios hoteles de lujo y nuevos centros de negocios, incluyendo edificios nuevos y remodelados, se han construido muy cercanos a la zona portuaria, por lo que su acceso (y vida, en general) está en directa relación con las actividades diarias del puerto. Establecer una base física cerca del puerto puede tener ciertas desventajas, como el ruido o el (posible) impacto de polución en el aire. Sin embargo, también tiene sus ventajas, como las vistas que ofrece a los clientes y empresarios y el privilegio de estar en contacto directo con el corazón empresarial de la ciudad.



**Figura 5.8:** Vistas del puerto de Tesalónica y su posición respecto a la ciudad



Las actividades portuarias y el crecimiento de las empresas están afectando enormemente a la congestión del tráfico en las calles circundantes al puerto y al centro de la ciudad. Además, la zona industrial de Tesalónica, la principal estación de autobuses y los centros logísticos están situados en el lado oeste. La puerta principal de camiones del puerto está en su lado oeste, contribuyendo con una cantidad significativa de tráfico durante las horas punta. En este sentido, la hora punta corresponde a los momentos más ajetreados de movimientos de camiones (entradas y salidas del puerto). Este contexto, hace que la zona sea incómoda para una serie de servicios públicos y la convierte en un área más contaminada.



**Figura 5.9:** Actividades portuarias relacionadas con el escenario de validación

De igual manera que estos picos de actividad (debidos en gran parte a los horarios de estiba y de transporte de mercancías) son difícilmente controlables, la apertura de puertas (nuevas líneas abiertas) o el establecimiento de flujos controlados son acciones que pueden tomarse para reducir dicho impacto. No obstante, hoy en día no se dispone de una monitorización ni interpretación suficiente de la información para aplicar estas medidas de una manera eficaz.

Mientras que es cierto que existe una observación (con tags RFID) en las puertas (interfaz real puerto-ciudad), el valor aislado de vehículos atravesándolas no es suficiente para comprender la congestión actual ni predecir cuál puede ser en las próximas horas. Por un lado, las actividades portuarias (por ejemplo, por el número de buques que se operan), las condiciones meteorológicas (por ejemplo, al afectar a las operaciones portuarias disminuyendo la productividad [479] o al observar que las lluvias provocan una mayor congestión del tráfico en las ciudades [480]), la estacionalidad (estación, mes, día dentro de la semana) y, naturalmente, la contribución para la propia ciudad, es decir, el tráfico en los alrededores del puerto atribuible a la ciudad, son factores influyentes en dicha congestión. Se podría argumentar que hay que tener en cuenta un número mayor de requisitos y de naturaleza más compleja [481], pero para el ámbito de este trabajo, los principales factores que afectan a la congestión fueron considerados los expuestos.

Por otro lado, los principales actores implicados en el evento de congestión del tráfico son la Autoridad Portuaria, el ayuntamiento de la ciudad e, indirectamente, los ciudadanos de Tesalónica. Durante el trabajo expuesto en este documento, se recibió una participación activa de la Autoridad Portuaria, canalizada a través de la participación en el proyecto PIXEL. La Autoridad Portuaria de Tesalónica (THPA) colaboró (dentro de ese proyecto) proporcionando datos de tráfico en las puertas del puerto, co-diseñando el concepto del experimento



(posibles influencias en el tráfico, fuentes de datos relevantes a explorar), indicando una serie de requisitos funcionales y esbozando el diseño de visualización que una Autoridad Portuaria querría tener en ese tipo de escenario.

Recapitulando, el evento que se desea modelar es la congestión del tráfico en la interfaz del puerto con la ciudad. El índice compuesto que se define es, por tanto, un índice de congestión del tráfico (**TCI – Traffic Congestion Index**). El TCI debe considerar los diferentes elementos que influyen en dicha congestión, así como los diferentes actores que intervienen en la definición de las políticas.

Además, este escenario incluye la necesidad de predecir dicho índice. El usuario final desea predecir/prever el tráfico rodado que implica posibles cuellos de botella en sus puertas de entrada/salida. Esta información servirá de herramienta para la toma de decisiones al operador de la terminal. La intención es disponer de una predicción del TCI a corto plazo, basada en la información disponible del puerto y de otros proveedores de datos, a partir de la construcción del índice compuesto. Esta acción requiere un algoritmo de predicción (módulo de previsión de series temporales de la arquitectura propuesta).

El potencial de la predicción de los tiempos de espera permitirá al puerto identificar los cuellos de botella y estimar el impacto del flujo actual de entrada/salida de camiones que entran/salen del puerto, teniendo en cuenta el tráfico real en la (cercana) ciudad. Ello podría redundar en una mejora de los flujos de tráfico y reducir el impacto medioambiental del puerto hacia la ciudad de Tesalónica, por ejemplo regulando el número de puertas de entrada/salida en funcionamiento para optimizar el tráfico de entrada/salida sin afectar demasiado al tráfico de la ciudad (aliviar el cuello de botella).

Para abordar esta historia de usuario, se necesitan los siguientes aspectos: (i) datos históricos para crear una base sólida para la previsión, (ii) datos actuales del nivel de tráfico en la ciudad, de modo que se detecten las próximas situaciones de atasco, (iii) sensores/información del estado de las puertas para mapearlo con la situación circundante, (iv) un índice fácilmente interpretable que proporcione información sobre el número de vehículos y otros factores que influyen en la congestión real (TCI).

Conviene destacar un aspecto adicional. Además del TCI, se especificó que desearía visualizarse el estado del tráfico en los alrededores de las puertas. En este sentido, este escenario de validación requirió de una adición en el módulo de visualización de la arquitectura de la solución (sección 3.5). Asimismo, se requirió la creación de dos umbrales de valor de dicho indicador, a partir de los cuales se visualiza un código de colores (similares a un semáforo).

En resumen, el puerto de Tesalónica aprovecharía este despliegue para explotar la información obtenida de los dispositivos ya instalados y de otros nuevos, con el fin de hacer que el tráfico y las operaciones portuarias sean ecológicos. La solución serviría para mejorar la toma de decisiones de los operadores de las puertas.

### 5.3.2. Implementación de la solución

A continuación se exponen los hallazgos, decisiones y resultados obtenidos durante el escenario de validación asociado al índice TCI.

#### *Fase I – Conceptual y de servicio*

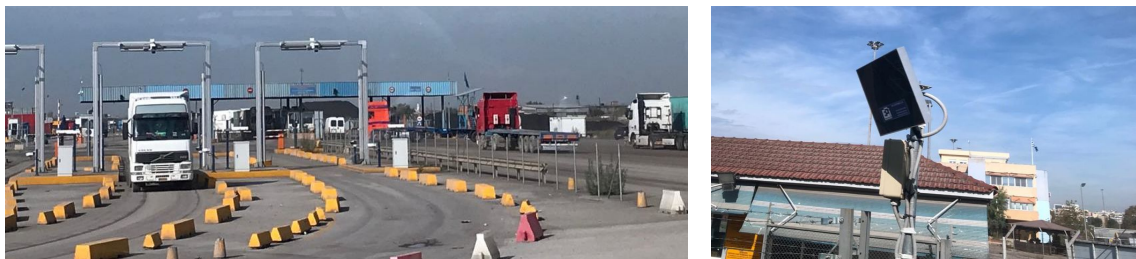
Como se ha indicado en el capítulo anterior, se contó con la colaboración estrecha de agentes directamente involucrados en el proceso para definir el índice compuesto. En este caso, la herramienta se enfocó principalmente para realizar análisis internos (monitorización, predicción, tomar medidas preventivas y correctivas). A diferencia de otros expuestos, el índice no fue concebido para la comparación entre diferentes entidades ni para establecer rankings de cara a análisis públicos de rendimiento/impacto. Por ello, el diseño del CI en este caso fue algo más relajado en el sentido de revisión literaria y escalabilidad. Sin embargo, se abordó desde un enfoque eminentemente práctico, tomando como entradas los datos influyentes más importantes y priorizando la interpretabilidad de los subíndices.

Según la definición del caso de uso, el índice compuesto TCI representa el estado de congestión debido a diversos factores. Para construir la estructura del CI, se procedieron a analizar los datos, establecer un esquema de árbol estructurado y ponderar los nodos que lo conforman.

Partiendo de esa base, se expone un breve detalle de los datos históricos y en tiempo real que se utilizaron (tras un análisis de la información disponible) para el cálculo y predicción del índice compuesto.

#### **Tráfico a las puertas del puerto usando sensores de radiofrecuencia (RFID):**

Los datos de referencia para el escenario de validación proceden de dos sensores RFID situados en las puertas del puerto de Tesalónica. Las puertas 10A y 16 (numeradas por el puerto) están equipadas con sensores basados en RFID que permiten identificar y catalogar todos los vehículos que entran y salen del puerto.



**Figura 5.10:** Datos procedentes de los sensores RFID en las puertas del puerto (a) arcos RFID en la entrada al puerto de la puerta de camiones (16); (b) etiqueta RFID y placa de detección de vehículos en la puerta de vehículos comunes (10A).

Los datos (provistos por la Autoridad Portuaria de Tesalónica) provienen directamente de los sensores y son servidos a través de una API REST alojada en sus servidores. De cara al entrenamiento a la muestra de información histórica, se contó con datos desde abril 2018 hasta febrero 2020 (para no influenciar la predicción por la reducción de tráfico marítimo debido a la pandemia por COVID-19). Se descubrió que una media entre 500 y 900 coches atraviesa cada día cada una de las puertas en cada dirección.

### **Tráfico de la ciudad (alrededores del puerto):**

Se definió que uno de los parámetros influyentes en la congestión en el acceso era el tráfico en las calles circundantes al puerto de Tesalónica. Para ello, se hizo uso de una fuente de datos abierta que proporciona información (procesada partir de datos GPS de los vehículos) sobre la velocidad de circulación en tramos de la ciudad. Esta página (TrafficThess [482] proporciona tanto datos históricos como en tiempo real – con una frecuencia de refresco de 15 minutos).

Es importante localizar las puertas de acceso/salida del puerto, así como las vías de comunicación pertinentes que hay que seguir, como se muestra en la figura siguiente. La THPA sólo está interesada en controlar el tráfico cerca de las puertas 16 y 10A.



**Figura 5.11:** Información aportada por la web TrafficThess para el cálculo del TCI contextualizando la posición de las puertas 16 y 10A del puerto de Tesalónica

### **Barcos atracados en el puerto:**

Al igual que el tráfico en las puertas del puerto, pudieron obtenerse datos históricos y en tiempo real de los buques procesados en el Puerto de Tesalónica por cortesía de la Autoridad Portuaria (AP). Estos datos fueron servidos por el departamento de Tecnologías de la Información del Puerto de Tesalónica a través de una API web REST análoga a la anteriormente mencionada. Esto fue posible gracias a la participación del candidato en el proyecto PIXEL, participado igualmente por la Autoridad Portuaria.

Los datos utilizados para el entrenamiento inicial del modelo predictivo se remontan (para mantener un paralelismo con las otras fuentes) a abril de 2018.

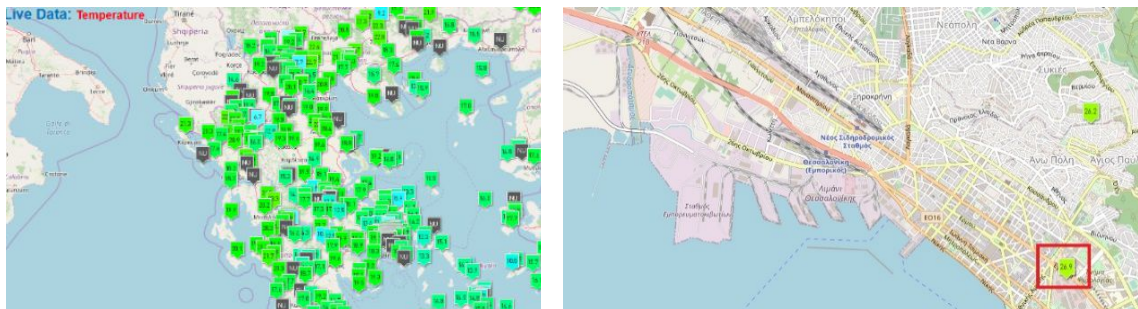
No obstante, durante el experimento de validación se utilizaron datos “en tiempo real”, ya que dicha API se actualiza cada vez que la llegada de un nuevo buque es anunciada a la Autoridad Portuaria.



**Figura 5.12:** Información de barcos procesados por el Puerto de Tesalónica, ofrecida por la Autoridad Portuaria para el cálculo del TCI

### Datos meteorológicos:

Tras el análisis inicial, se decidió que las condiciones meteorológicas (actuales y predicción a corto plazo) influían en el estado de la congestión en las puertas. Para ello, se propuso la utilización del servicio de datos abiertos *Stratus Meteo* de Grecia [483]. Hay diferentes sensores instalados por todo el país y para este caso se analizó cuál era la estación más apropiada (la más cercana al puerto de Tesalónica), tal y como se muestra en la siguiente figura:



**Figura 5.13:** Distribución de estaciones Stratus Meteo Grecia y selección de la más apropiada para alimentar una fuente de datos para el cálculo del TCI

Los datos ofrecidos por esta fuente de datos eran, sin embargo, de una peor calidad que las fuentes anteriores. Con una frecuencia de muestreo diaria y una disponibilidad de (solo) desde septiembre de 2018, supusieron una limitación para el servicio de predicción del TCI.

A continuación se muestra una tabla resumen de los datos que se utilizaron para este escenario de validación. Una tabla equivalente se ofrece en el resto de casos de uso. El objetivo es condensar la información relacionada con las fuentes de datos que se emplearon. Este ejercicio sirvió igualmente al candidato para comprender la variedad de datos que la arquitectura propuesta es capaz de gestionar sin requerir modificaciones (únicamente conectores ad-hoc). Se indica el tipo de fuente, su significado, la tecnología utilizada, cómo se ofrece al sistema, los

parámetros más relevantes, la periodicidad y las unidades de medida. Además, en este escenario de validación se indica el histórico de datos utilizado ya que este módulo de la arquitectura ha sido incluido en el despliegue.

**Tabla 5.4:** Tabla resumen de datos en el escenario TCI

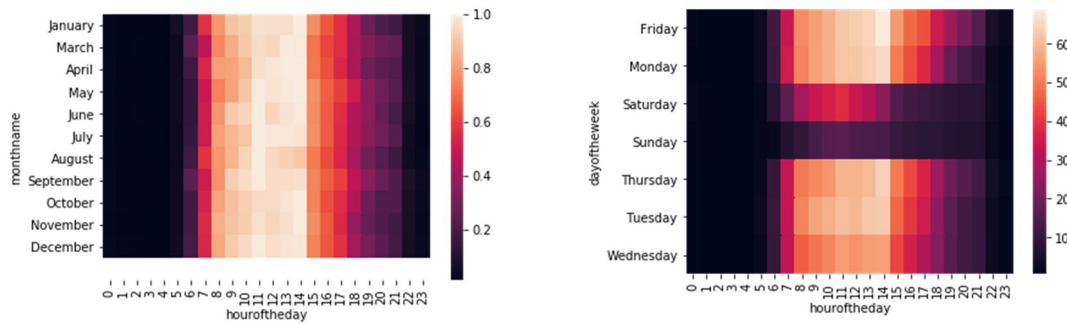
| Fuente de datos      | Conectividad            | Parámetros relevantes                | Frecuencia de muestreo | Unidades       | Histórico       |
|----------------------|-------------------------|--------------------------------------|------------------------|----------------|-----------------|
| Tráfico en puertas   | RFID                    | Vehículos                            | Por hora               | # de vehículos | Abril 2018      |
| Tráfico en la ciudad | GPS                     | Velocidad media                      | Por hora               | Km/h           | Septiembre 2018 |
| Meteo                | Estación meteo          | Temperatura, viento, precipitaciones | Por día                | °C, km/h, mm   | Abril 2018      |
| Buques atracados     | Anuncio de barcos (API) | Tiempo de estancia                   | Por hora               | # de barcos    | Abril 2018      |

Basado en lo anterior, tal y como se especifica en la metodología, debió decidirse la frecuencia con la que deseaba calcularse el servicio de cálculo (y, en este caso, predicción) del índice compuesto. Acorde a los datos, la entrada más restrictiva es la estación meteorológica, lo que llevo a asociar un índice por día. Por tanto, en este caso, el CI es una herramienta retroactiva que muestra el estado de las últimas 24 horas.

Avanzando en la ejecución de la fase I, este escenario de validación incluyó la indagación más profunda de los datos en forma de análisis exploratorio (EDA – del inglés *Exploratory Data Analysis*). Este ejercicio se ha realizado ya que es posible que el índice deba tener en cuenta la correlación de diferentes elementos (escalas de buques, tráfico en la ciudad, estacionalidad, etc.), para poder condensar la información influyente relevante. El objetivo principal fue descubrir la naturaleza y de cada fuente de datos (si presenta estacionalidad, si se mantiene constante con el tiempo, si existen patrones...). En los siguientes párrafos y gráficas se muestran los hallazgos en este sentido. Esta actividad no fue replicada con el mismo grado de profundidad en el resto de escenarios de validación debido, principalmente, a una menor capacidad de obtención de históricos. Como nota adicional, estos análisis se realizaron fuera del contexto de la arquitectura (no existe un módulo de EDA en la solución), usando cuadernos de *Jupyter (Notebooks)* y la librería de ciencia de datos llamada *pandas*.

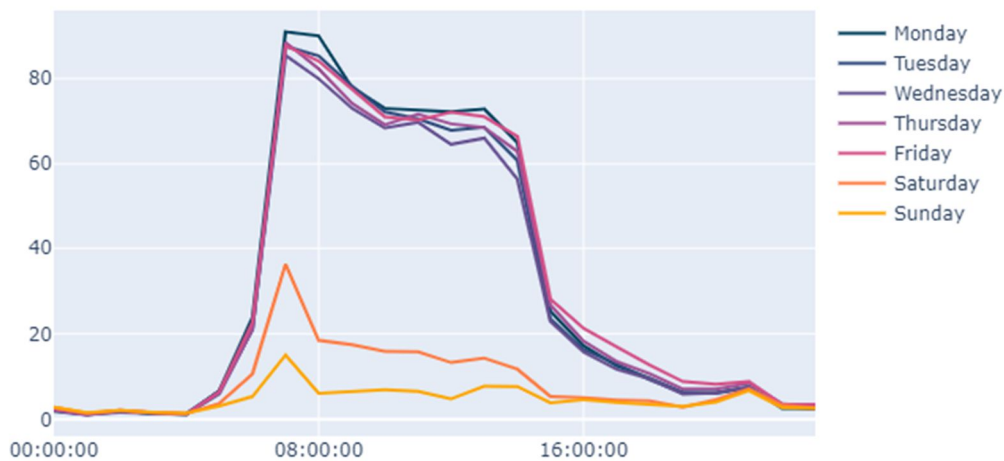
La estrategia seguida fue la de realizar un análisis pormenorizado de los datos de tráfico en las puertas 16 y 10A del puerto en base al histórico disponible, reduciendo la intensidad de análisis para las otras tres fuentes de datos, constriniéndolo a un estudio de la correlación con la fuente de datos principal (la primera mencionada).

Primeramente, en cuanto a las puertas, se observó enero, mayo y septiembre son los meses con más congestión de tráfico, tanto en volumen como en número de horas de alta congestión (hasta las 17 horas). Por otro lado, una exploración de la misma información por semana desveló que no existe el mismo tráfico entre semana que los festivos (domingos, principalmente). Ambos análisis revelaron que las horas centrales del día son las de mayor congestión, especialmente entre las 12 y las 14 horas, donde se notan los índices máximos.



**Figura 5.14:** EDA de los datos de puertas en el puerto de Tesalónica. (a) por mes y (b) por día de la semana.

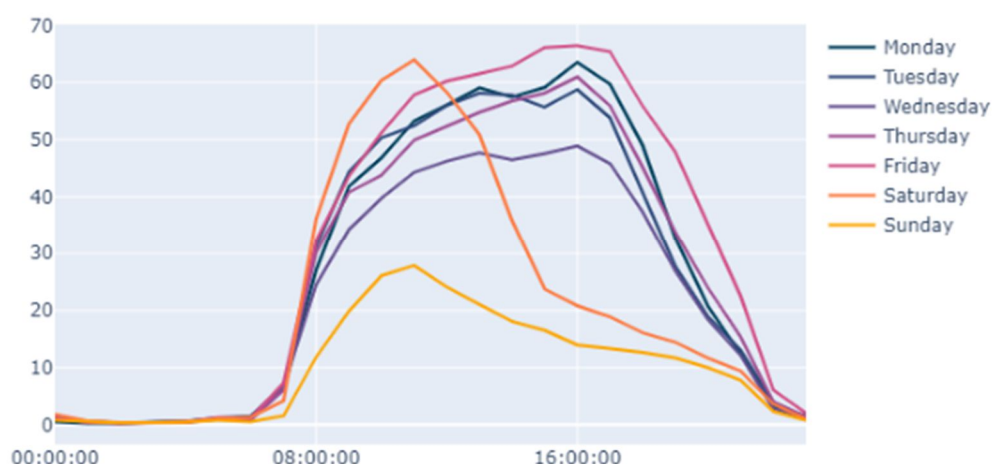
Se aporta también en el siguiente par de gráficas las “curvas” habituales de evolución del tráfico en las puertas del puerto durante cada día de la semana. Se ha decidido mostrar el tráfico en figuras separadas, para ilustrar las diferencias entre entradas y salidas hacia/desde dichas puertas.



**Figura 5.15:** Gráfica de número de vehículos entrantes por puerta 10A en TCI

Observando los escenarios del experimento, la evolución en cuanto a la entrada en los días martes, jueves y sábado fue especialmente relevante para representar en este resumen: los sábados no están muy congestionados en cuanto a tráfico, lógico al no ser un día laborable.





**Figura 5.16:** Gráfica de número de vehículos entrantes por puerta 16 en TCI

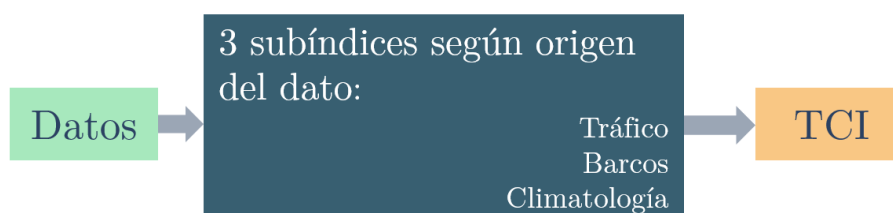
Respecto a las salidas, los jueves (con los viernes) son los días con más tráfico, posiblemente debido a la necesidad de procesar la carga/actividades pendientes antes de terminar la semana. Para los martes, se observa que mantienen una distribución casi idéntica por hora considerando la suma del tráfico de las dos puertas en ambos sentidos.

Finalizando el análisis exploratorio de los datos, se muestra a continuación una tabla que resume las correlaciones encontradas entre las diferentes series históricas de las fuentes secundarias (buques atracados, datos meteorológicos y tráfico en la ciudad – en calles colindantes) con la fuente primaria (vehículos a la entrada de las puertas). En relación a los datos meteorológicos, ya que se han obtenido diversos campos que pueden ser considerados separables, se procedió a un análisis independiente para cada uno de ellos. Se añade además una posible explicación a dichas correlaciones.

**Tabla 5.5:** Resumen de correlación entre las fuentes de datos en el escenario TCI

| Fuente de datos                       | Valor de correlación (%) | Posible explicación y reflexión de cara al cálculo del TCI  |
|---------------------------------------|--------------------------|---|
| Velocidad media en calles colindantes | -19,2%                   | A menor velocidad (más congestión), se experimenta más tráfico en las puertas. Fuente secundaria con mayor correlación. |
| Número de buques atracados en puerto  | +11,7%                   | A más barcos en el muelle, más tráfico en las puertas. Menor relevancia, pero apreciable                                |
| Temperatura                           | -7,7%                    | No hay mucho impacto estadístico  |
| Velocidad del viento                  | +8,6%                    | Cuanto más viento, más tráfico, pero sin relación relevante para ser concluyente  |
| Intensidad de precipitación           | +3,7%                    | Muy poco acoplado, casi sin correlación estadística.  |

A partir de este momento, se procedió al diseño de los subíndices. Como se ha citado previamente, el enfoque práctico predominó en este escenario de validación. Siguiendo esta mentalidad, aún con una mentalidad subyacente similar a las propuestas de PIXEL o CITYKeys, se optó por la separación de fuentes únicamente por origen y significado [484]. Así, se estableció un subíndice asociado a la contribución “portuaria” (englobando las fuentes de número de vehículos en las puertas 16 y 10A y a los buques atracados en el puerto), otro subíndice representado la contribución de la ciudad (velocidad de circulación en las calles colindantes) y, por último, un tercer subíndice asociado a los datos meteorológicos.



**Figura 5.17:** Identificación de subíndices en el escenario TCI

Por último, para completar esta fase, se realizó una reflexión sobre cuáles debían ser los métodos matemáticos a escoger de entre los disponibles en cada paso (véase sección 4.1.2).

**Tabla 5.6:** Elección de métodos matemáticos para el escenario TCI

|   |                                |                   |
|---|--------------------------------|-------------------|
| <b>Gestión de datos ausentes:</b>   | <i>Hot deck imputation</i>     | Último valor      |
| Tras el análisis de los datos exploratorios, se observó que todas las series temporales contaban con todos los registros en el periodo abril 2018-febrero de 2020. Durante la ejecución del escenario de validación (captura de datos continua), no se observaron grandes alteraciones a nivel datos ausentes. Por lo tanto, se eligió el método más habitual, para que, en el caso que sucediera, se tuviera suficiente cobertura. En particular, se eligió <i>Hot Deck</i> , que en el caso de la solución propuesta consiste en replicar el último valor recibido.   |                                |                   |
| <b>Normalización:</b>   | Re-escalado ( <i>Min-Max</i> ) |                   |
| Para que los <i>cKPIs</i> puedan ser introducidos dentro del árbol de cálculo del CI deben estar expresados en la escala 0 a 1. Para ello, debió elegirse la mejor estrategia de normalización. El método de ranking se descartó debido a la existencia de una gran variación de la magnitud, así como el método de clasificación (o referencia a un punto) debido a que no se trataba de un escenario con más de una entidad que comparar (solo el puerto de Tesalónica). El método Z-score y el Min-Max fueron los principales candidatos. Teniendo en cuenta que se pudo realizar un EDA en profundidad con datos de más de 18 meses, se interpretó la naturaleza de los mismos y se seleccionó la opción Min-Max. Según la solución propuesta, este método era capaz de seleccionar los datos mínimo y máximo (aplicando márgenes) y normalizar los <i>cKPIs</i> en función de ellos. |                                |                   |
| <b>Ponderación:</b>   | Método analítico               | Según correlación |



Uno de los métodos más importantes dentro de la construcción del índice compuesto es la ponderación. La elección no solo del método en sí sino también de los valores otorgados finalmente puede tener un efecto directo en la interpretabilidad del índice y subíndices. La ponderación equitativa suele utilizarse cuando no se conoce información previa sobre los elementos, lo cual no era el caso en este escenario de validación. El método basado en la opinión se utiliza sobre todo en las ciencias sociales y en los casos en que el indicador objetivo tiene un fuerte componente de subjetividad, lo cual no era deseado por la Autoridad Portuaria (grupo de interés principal en este escenario). Por ello, se optó por el método analítico; concretamente, por basar los valores en función de la correlación de valores con el principal ítem de referencia, ya que el histórico disponible permitía a los autores aplicar dicho esquema. Para establecer los valores de ponderación a partir de los números y la estructura de los puntos anteriores, los autores llegaron a mapear *cKPI*-ponderación, subíndice-ponderación, lo que llevó a resolver tres sistemas de ecuaciones. Los resultados se muestran en la subsección, cuando se introdujeron en la configuración a través de la interfaz de usuario:

$$\sum P_i \times Corri - otras_{fuentes} = 1 \quad (16)$$

|  |         |
|--|---------|
| <b>Agregación:</b>   | Aditiva |
| <p>Como se ha comentado en la sección 4.1.2, los métodos más comunes son la agregación aditiva, agregación geométrica y agregación no compensatoria. A pesar de estar de alguna manera acoplados, la sinergia y relevancia de todos los indicadores no fue considerada igual, por lo que se decidió seleccionar la agregación aditiva para este escenario de validación. Los motivos principales fueron: (i) la falta de información para proponer una agregación geométrica y (ii) debido a la transparencia y facilidad de la agregación aditiva para analizar los resultados bajo una perspectiva lógica. No obstante, conviene destacar que el servicio desarrollado en esta tesis ha sido diseñado para permitir la selección de cualquiera de ellos.</p> |         |

### *Fase II – Arquitectura y conectores*

Una vez concluida la fase de diseño del servicio, se procedió a la fase de instanciación de la arquitectura en el escenario de validación para permitir la ejecución del mismo.

De igual modo que se plantea en el resto de casos de uso, esta fase comenzó con un análisis de las necesidades modulares. Como se ha explicado, los principios de diseño han guiado a una confección modular (orientación a microservicios) de la arquitectura propuesta. Por ello, y aunque se han establecido una serie de componentes esenciales que no deben ser prescindidos, se procedió con el siguiente análisis:

No se dispone de una plataforma inicial sobre la que la solución deba integrarse. Si bien es cierto que el puerto de Tesalónica se encontraba durante el desarrollo del escenario participando en el proyecto PIXEL, el estado de desarrollo de la plataforma de dicho proyecto no se encontraba en un nivel de madurez suficiente (cuenta con módulos mucho más tecnológicamente sofisticados que la arquitectura de esta tesis, como por ejemplo el módulo orquestador de modelos heterogéneos). Por tanto, los módulos esenciales del sistema bajo estudio debían ser incluidos: adquisición de datos, persistencia (con conversor incluido), seguridad y visualización. Además, este escenario particular, debido a los requerimientos de predicción a corto plazo, fue el caso perfecto para incluir los módulos de entrenamiento e inferencia de la arquitectura. Adicionalmente, modificaciones de la interfaz de visualización se antojaron necesarias para cumplir con los requisitos planteados.

De esta manera, el despliegue de la arquitectura fue el siguiente:

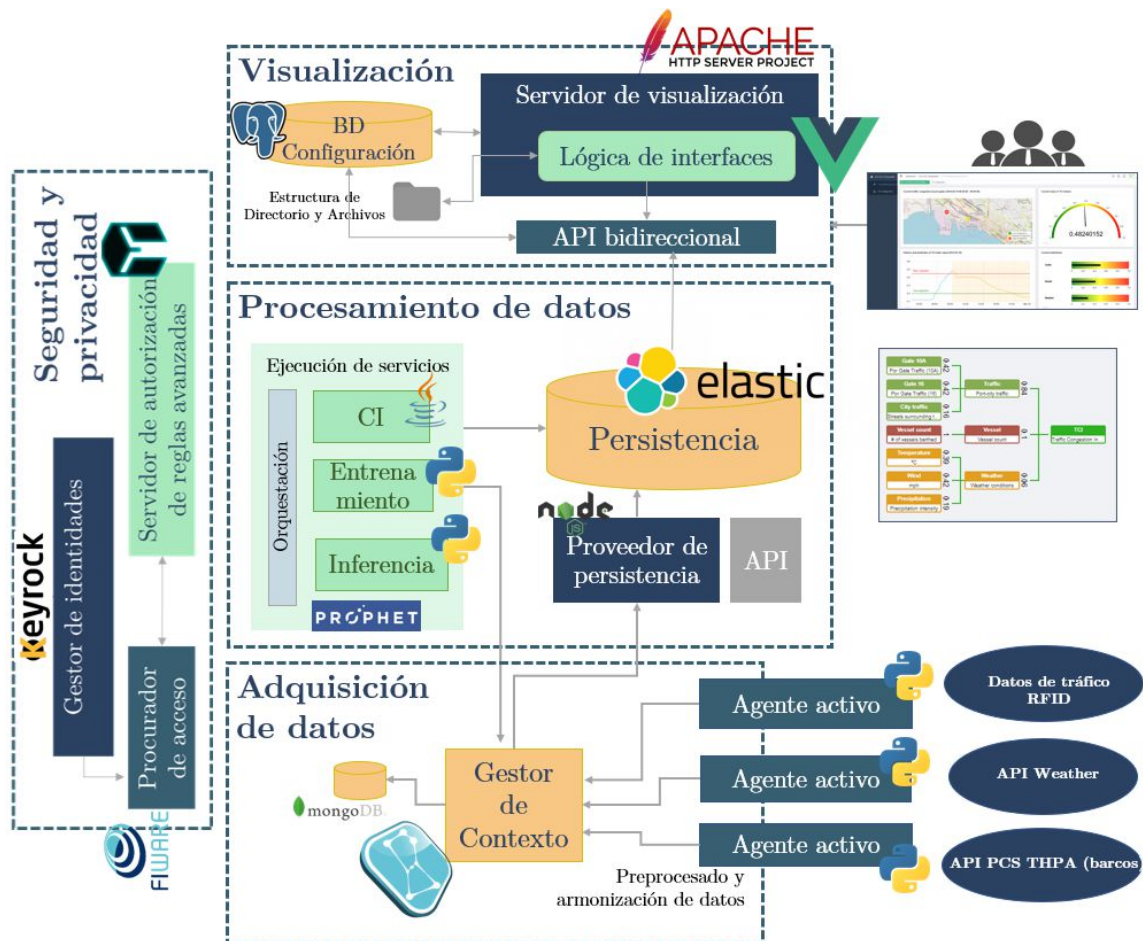


Figura 5.18: Arquitectura desplegada en el escenario TCI

En este momento, se procedió a instalar los elementos esenciales de la plataforma, llevando a cabo las comprobaciones indicadas en la metodología (sección 5.2.3).

No obstante, la arquitectura carece de uso si no se le conectan datos IoT para dar valor y poder ejecutar el servicio de índices compuestos. Por ello, se desarrollaron los agentes suficientes para integrar la información proveniente de las diferentes fuentes de datos (vehículos, tráfico, barcos y datos meteorológicos). A continuación se expone una pequeña referencia sobre dichas implementaciones. Cabe destacar que para su desarrollo software se empleó la librería *pyngsi*.

**Agente de conexión de datos de vehículos a las puertas del puerto:**

El preprocesamiento realizado en este agente fue incluido tanto en su versión histórica (ejecutado una sola vez para los datos desde abril 2018) como para su versión en tiempo real (en continua ejecución). Se consultaron los datos de abril de 2018 de las entradas/salidas pasadas de los vehículos y se estableció la misma estructura para ambos usos. El proceso consistió en recuperar todos los vehículos del periodo, filtrar por el campo “*gate*”, analizar los horarios en los que cruzaron por las puertas (extracción de la hora dentro del campo “*entry\_time*” y/o “*exit\_time*”), establecer rangos horarios y contar después el número de vehículos en ese rango. Este ejercicio se materializó en scripts de Python y a continuación los agentes fueron “dockerizados” e integrados en la arquitectura. En la siguiente figura se puede observar el modelo de datos de entrada que se recibía en este agente. La salida se realizó, según acordado por arquitectura, según el modelo *cKPI*.

```

{
  "id": "1648422",
  "vehicle_type": null,
  "brand": "",
  "model": "",
  "colour": null,
  "entry_time": "9\6\2020 06:49:01",
  "exit_time": "21\6\2020 07:24:08",
  "entry_gate": "Gate 10A - Entry",
  "exit_gate": "Gate 10A - Exit",
  "dwell_time": "12d 35' 7\" ",
  "notes": ""
},

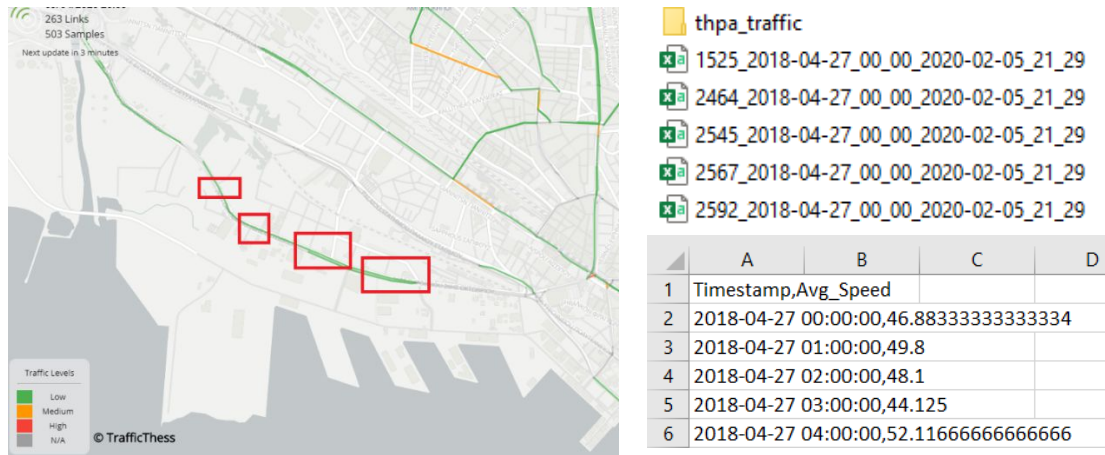
```

**Figura 5.19:** Modelo de datos de entrada del número de vehículos atravesando las puertas del puerto en el escenario TCI

**Agente para la obtención de velocidad media en las calles circundantes al puerto de Tesalónica:**

El principal factor de relación entre la fuente principal (la expuesta inmediatamente anterior) y la secundaria se dio con esta entrada de datos. Se trata del estado del tráfico alrededor del puerto representado por la velocidad media de los vehículos que las circulan. El preprocesamiento seguido sobre los datos brutos fue: (i) seleccionar cinco de los enlaces circundantes a las puertas del puerto, (ii) extraer los datos de esos cinco enlaces desde abril de 2018 hasta febrero de 2020

(mismo razonamiento que antes, incluyendo la dualidad agente histórico y agente de “tiempo real”), (iii) seleccionar los campos interesantes de la información proporcionada, (iv) construir un CSV con la media de las velocidades medias de los cinco enlaces circundantes. Esto se ilustra brevemente a través de las imágenes de la siguiente figura:



**Figura 5.20:** Representación de los datos procesados por el agente sobre tráfico en calles circundantes al puerto de Tesalónica en el escenario TCI

Como se ha explicado, el histórico de TrafficThess se muestrea cada 15 minutos. Para que los cálculos fueran coherentes, se promediaron los datos históricos y en tiempo real y se condensaron para tener una granularidad de 60 minutos (lo cual es correcto ya que la periodicidad de cálculo del TCI fue establecida en 1 día).

**Agente para la obtención del número de buques atracados en el puerto de Tesalónica:**

La Autoridad Portuaria puso a disposición una API que devuelve todos los buques operados en formato JSON incluyendo abundante información que debía ser preprocesada (el formato puede consultarse en la Figura 5.21: Modelo de datos recibido por el anuncio de buques en el puerto y CSV preprocesado por parte del agente (para caso histórico).).

Esta información sólo está referida a la fecha incluyendo los campos de “start\_work” y “end\_work”, por lo que se debió: (i) descargar todos los datos de todos los buques procesados en abril de 2018-febrero de 2020 (para el agente histórico), (ii) ajustar el marco temporal, (iii) agrupar, filtrar y contar los buques para tener un archivo CSV - preparado con la información adecuada: número de buques atracados/maniobras en el puerto separados por periodos de 60 minutos (similar al caso anterior).

A partir de este CSV, se realizó un segundo componente software convirtiendo esta información en el formato adecuado para el CI (*cKPI*). En este caso también se estableció un agente equivalente para la extracción en tiempo real.



**Figura 5.21:** Modelo de datos recibido por el anuncio de buques en el puerto y CSV preprocesado por parte del agente (para caso histórico).

**Agente para la obtención de datos meteorológicos:**

Las mediciones históricas y en tiempo real de la temperatura, la velocidad del viento, la intensidad de la precipitación, la humedad y el punto de rocío (entre otros) fueron recuperados de este servicio externo con una granularidad diaria. En este sentido, el preprocesado de este agente incluyó la remodelación de los datos para ajustar la estructura al *cKPI*, así como el filtrado de la información innecesaria. En particular, desde este agente no se informó al sistema de una sola fuente de datos, sino de tres de ellas. Como se consideró información relevante tanto la temperatura como la velocidad del viento y la intensidad de precipitación, se generaron tres “salidas” para este agente. Todo ello fue introducido dentro de la imagen *Docker* generada.

Tras la comprobación de la integración de estos datos, este escenario de validación avanzó hacia su componente adicional: la predicción de los índices.

**Servicio de predicción de índices:**

Según lo expuesto en la sección 4.2.3, la tecnología para la predicción de las series temporales (los propios *cKPIs* e índices) se encuentra fijada (*Facebook Prophet*), y los módulos de la arquitectura se encuentran preparados para proveer dicho servicio.

En este sentido se instalaron dichos módulos, no sin antes haber realizado una serie de pruebas y modelos (por *cKPI*) en cuadernos de *Jupyter* para analizar la

mejor estrategia de entrenamiento e inferencia (esta actividad se puede consultar en el artículo del candidato publicado en la revista *Sensors* [460]).

En cuanto a la implementación del servicio, el módulo está preparado para recibir una serie de parámetros de configuración que le permitan proceder con el entrenamiento y la inferencia de manera automática. En este paso se configuraron dichos parámetros de la siguiente manera:

- **Introducción de datos de estacionalidad:** Se introdujeron los días festivos de Grecia, que se añadieron a los que por defecto incorpora el framework *Prophet* (de estos, se seleccionaron por día – según EDA hay patrones horarios que se repiten- y por semana – ídem).
- **Periodicidad de la predicción:** Cada cuánto se realiza una inferencia. Esto tiene influencia directa en los resultados observables por parte del usuario. Según discutido, se estableció 1 hora como la periodicidad (observando que es la frecuencia de actualización común de casi todas las fuentes de datos – excepto datos meteorológicos).
- **Horizonte de predicción:** Desde el comienzo del escenario de validación se definió que se trataba de una predicción a corto plazo. Se escogió como horizonte 24 horas (1 día). De esta manera, trataba de proveer información a la Autoridad Portuaria de lo que podría esperarse durante las próximas horas.
- **Frecuencia de re-entrenamiento:** Se estableció como frecuencia de re-entrenamiento 1 semana. Como se expone en la sección 4.2.3, *Prophet* no es un modelo supervisado continuo, por tanto genera predicción sin un input instantáneo. No obstante, observando las tendencias de los datos en más de 18 meses, se estableció esta frecuencia como suficiente.
- **División del corpus de datos para el re-entrenamiento:** Se decidió optar por una de las estrategias más comunes, 70/30, empleando el 70% de datos para el entrenamiento y el 30% para la validación, usando diferentes métricas de error.

Algunos resultados obtenidos (contextualizados a un marco más amplio) se han incluido en la siguiente subsección.

#### **Software y equipamiento utilizado:**

Por último, se concluye esta subsección (como se realiza en el resto de escenarios de validación) con la indicación de los detalles técnicos del despliegue.

En este caso, todos los módulos se desplegaron en la misma máquina, excepto las APIs REST de las fuentes de datos (tráfico de la ciudad, meteorología y THPA), que son servicios ofrecidos por proveedores externos. Esta máquina (en el contexto de esta tesis doctoral, se indica como escenario “*cloud*” aunque se trata de un



servidor *on-premise*) pertenece a la nube privada del grupo de investigación del doctorando y sus especificaciones de hardware se detallan en la siguiente tabla.

**Tabla 5.7:** Equipamiento *hardware* utilizado en el escenario de validación TCI

| Elemento    | Especificación técnica / modelo                 |
|-------------|---|
| CPU         | 4 CPUs x Intel® Xeon® CPU E3-1220 v5 @ 3.00 GHz |
| Disco duro  | HDD 100 GB                                      |
| Memoria RAM | 16,05 GB  |
| Clúster     | FUJITSU PRIMERGY TX1330 M2                      |

Además, tanto las especificaciones de software de los módulos preconstruidos (componentes FIWARE y servidor Apache) como las de las tecnologías utilizadas para desarrollar y ejecutar los demás módulos se detallan en la siguiente tabla:

**Tabla 5.8:** Tecnología *software* utilizada en el escenario de validación TCI

| Elemento                       | Tecnología / versión   |
|--------------------------------|--|
| Sistema operativo del servidor | Ubuntu Server 18.04.4 LTS  |
| Java                           | OpenJDK 1.8.0_252  |
| Apache Maven                   | 3.5.4  |
| Node.js + npm                  | 12.18.1 LTS + 6.14.5   |
| Python + pip                   | 3.8.3 + 20.1.1   |
| Docker + Docker Compose        | 18.09.7 + 1.17.1   |
| Apache Server                  | 2.4.43   |
| Elasticsearch                  | 7.8.0  |
| MongoDB                        | 3.6.3  |
| FIWARE Orion                   | 3.4.0  |
| FIWARE Cygnus                  | 2.2.0  |
| Entorno de desarrollo (IDE)    | Eclipse IDE for Enterprise Java Developers (v. 2020-06)<br>Visual Studio Code (version 1.46) |

### *Fase III – Configuración y resultados*

Por último, se procedió a ejecutar la fase III de la metodología, orientada a la configuración de parámetros (mediante la interfaz de la solución - 3.5) y a la observación de los resultados.

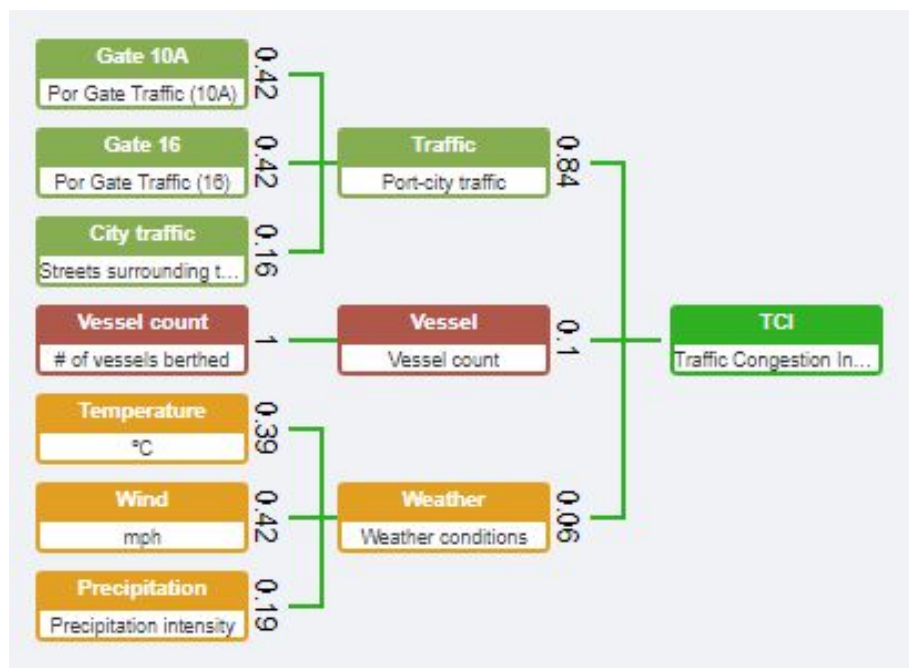
En este sentido, se siguieron las indicaciones de la Fase I en las que se indicaba los métodos matemáticos a elegir. En la Figura 5.22: Selección de métodos matemáticos en el escenario TCI se puede observar la selección (el árbol se muestra en la figura posterior):

|                        |                      |                             |           |
|------------------------|----------------------|-----------------------------|-----------|
| * Normalization method | RE-SCALING           | * Weighting method          | HANDTYPED |
| * Update strategy      | REPLICATE LAST VALUE | * Data imputation algorithm | HOT DECK  |

**Figura 5.22:** Selección de métodos matemáticos en el escenario TCI

En cuanto a la composición de la estructura de cálculo (árbol), los nodos iniciales (o de “hoja”), correspondiente a los *cKPIs* generados por los agentes se separaron individualmente por piezas aisladas de información (puerta 16 del puerto, puerta 10A del puerto, recuento de buques, temperatura, velocidad del viento e intensidad de las precipitaciones) Esto llevó a una composición de tres niveles, siendo los *cKPIs* los nodos de la hoja, tres subíndices (véase fase I) y el TCI como nodo raíz, resultando en una matriz 7:3:1.

En la siguiente imagen pueden observarse también los valores de ponderación de cada elemento, que resultaron de la resolución de los sistemas de ecuaciones (utilizando el método de Gauss) planteados en la sección anterior:



**Figura 5.23:** Árbol de composición del indicador TCI usando la solución

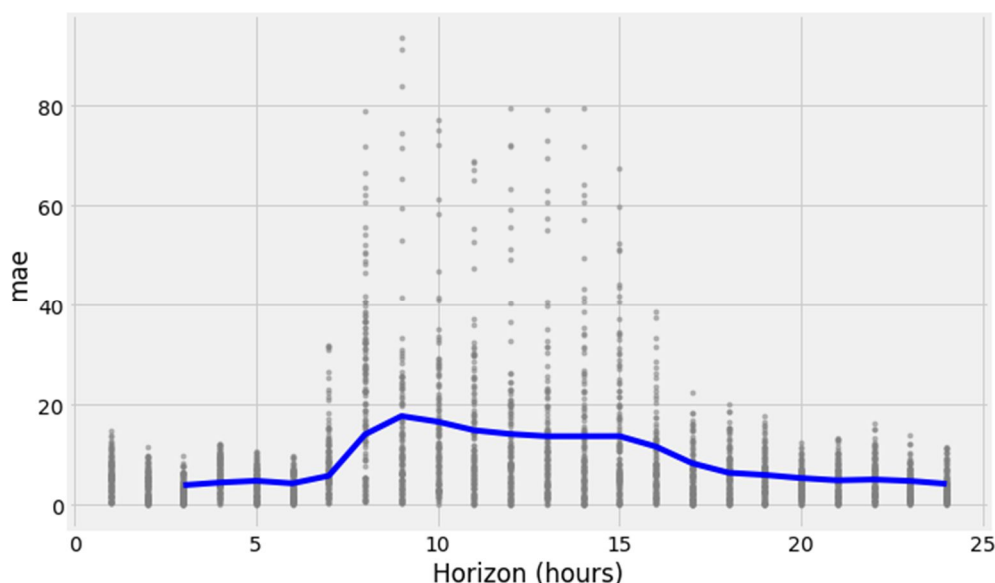
En este punto, antes de mostrar los resultados de visualización del índice, se aportan unas notas sobre el modelo de predicción, propio de este escenario.



Los resultados de la ejecución del servicio predictivo en este escenario de validación fueron parcialmente incluidos en las actividades del proyecto PIXEL, más en concreto en el entregable D4.4 [113]. No obstante, en dicho informe no se reportó la inclusión de la predicción de los índices (a partir de la serie temporal *cKPI*), sino sobre los valores y formatos de las fuentes de datos originales.

Partiendo de esa información, se puede hacer una pequeña reflexión sobre la apropiada elección de la librería *Prophet* de Facebook y la precisión de predicción que podría esperarse en este escenario de validación para los índices (que fue corroborada en la práctica). Para ello, se incluyen a continuación varias gráficas que representan los modelos específicos obtenidos sobre la fuente de datos principal del caso de uso (puertas 16 y 10A del puerto).

En primer lugar, se muestra un análisis de la métrica MAE (del inglés *Mean Absolute Error*) sobre la predicción de vehículos en dichas puertas con respecto al rendimiento del modelo sobre todo el set de datos (considerando la regla 70/30 escogida). Esta métrica proporciona información sobre la media de los errores absolutos entre el valor real y su predicción. En este caso, se indica “por hora” con un horizonte de un día (es decir, los valores seleccionados en la fase anterior). Por tanto, aunque esta gráfica se corresponde con el valor en las puertas y no con el índice y subíndices, es posible una extrapolación directa para ofrecer una perspectiva sobre la precisión alcanzada:

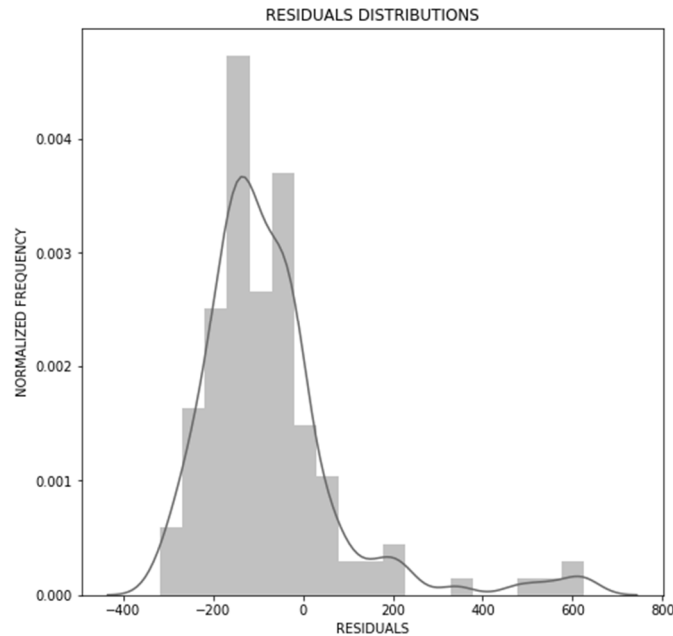


**Figura 5.24:** Precisión expresada en MAE del modelo predictivo usado en el TCI

De acuerdo con esto, en las horas de trabajo más activas del día (7-15) el error en el número de vehículos (por media) fue inferior a 20, lo que se considera una buena precisión (alrededor de 850 vehículos al día). Este patrón se repite con periodicidad diaria.

La siguiente figura muestra la distribución (en un diagrama de frecuencia) de los valores predichos frente a los valores reales. Como se puede observar, la “tasa

de error” (número de vehículos) que más se repite es la franja -80/+20. Además, se puede ver que esta discrepancia es mayor cuando el número de vehículos es menor. Esto significaría que para los días laborables de la semana este modelo predice mejor que para el fin de semana.



**Figura 5.25:** Precisión expresada en MAE del modelo predictivo usado en el TCI

La denominación de esta discrepancia son los “residuos”. La cifra más frecuente (en valor absoluto) es de 150 y el segundo residuo más frecuente es de 50 vehículos. Considerando que la media por día es de unos 850, una desviación de 75 es un 8,8% menos de vehículos previstos por día

Un resumen sobre las métricas obtenidas se puede encontrar en la siguiente tabla.

**Tabla 5.9:** Métricas de precisión en el escenario TCI sobre el total del data set

| Métrica  | Data set de test | Data set de validación |
|--|------------------|------------------------|
| Valor de correlación entre el valor y su predicción  | <b>0.851736</b>  | 0,826675               |
| Media diaria de MAE (desviación sobre el set de datos total con el horizonte de predicción del periodo completo) | <b>97,488</b>    | 139,289                |
| Mediana de los residuos  | 24,4823          | -                      |
| Valor medio de residuos  | -3.6398          | -                      |
| Sesgo  | -2,496           | -                      |
| MAPE   | 29,65%           | 64,09%                 |

Con respecto a los resultados del servicio integrados en la arquitectura propuesta, se aportan las siguientes evidencias. En primer lugar, se destaca que debió realizarse un desarrollo adicional sobre el módulo de visualización prediseñado en la solución (sección 3.5) para reflejar la predicción de los índices integrada en la misma visualización. Además, debido a requerimientos de este caso de uso en el proyecto, se incluyó una vista de mapa (usando la librería *Leaflet.js*) en la que se observa el valor de monitorización actual de las puertas 16 y 10A del puerto.

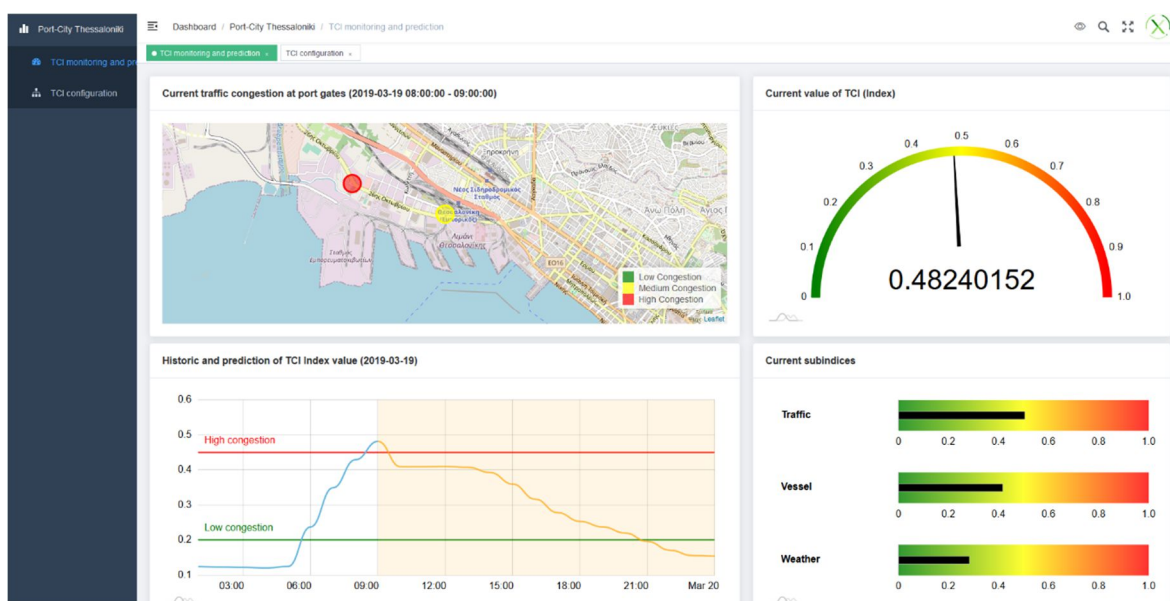


Figura 5.26: Resultado de visualización en escenario TCI (1)

Además, se deseaba visualizar en un mismo espacio (en cuatro cuadrantes, como se observa en las figuras) de una forma muy específica. Por tanto, aún partiendo de la base desarrollada para la solución, se trató de un desarrollo customizado para el escenario.

Por último, tras petición expresa, la representación de colores en los puntos del mapa donde se ubican las puertas de interés (16 y 10A) debía seguir unas reglas basadas en umbrales. Esto es, para favorecer una interpretabilidad directa del índice de congestión (TCI), se marcaron de color rojo, amarillo y verde dichos puntos siguiendo el esquema:

$$\begin{aligned}
 0 \leq \text{TCI} < 0.2, & \text{ Congestión baja} \\
 0.2 \leq \text{TCI} < 0.46, & \text{ Congestión media} \\
 0.46 \leq \text{TCI} < 1, & \text{ Congestión alta}
 \end{aligned}
 \tag{17}$$

Como se puede observar, dentro de la gráfica de evolución temporal se ha incluido la monitorización de los valores del TCI a lo largo del día (parte de línea azul sobre fondo blanco) así como los valores de predicción durante el resto del día (horizonte de predicción 24 horas), correspondiente a la línea naranja sobre fondo amarillo. Los umbrales de congestión también se incluyeron como líneas

horizontales sobre la gráfica para facilitar su interpretación en un vistazo. Se proporcionan un par de capturas adicionales para representar diferentes valores y casuísticas de TCI y predicción:

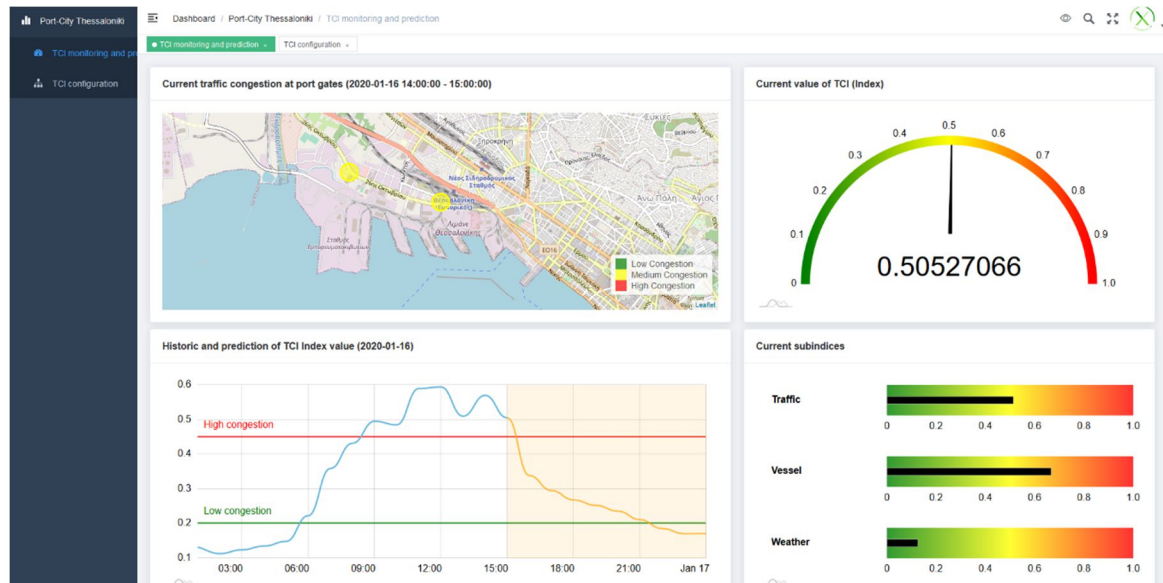


Figura 5.27: Resultado de visualización en escenario TCI (2)

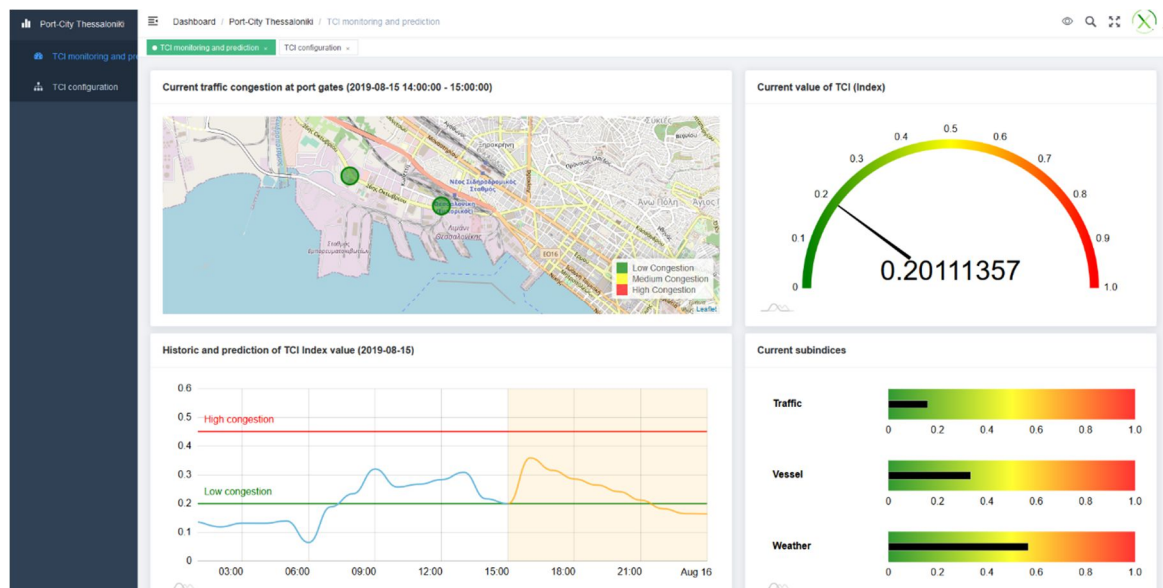


Figura 5.28: Resultado de visualización en escenario TCI (3)

En cuanto al impacto de aplicación de este escenario, el candidato (dentro del proyecto PIXEL) no tuvo suficiente capacidad ni recursos para involucrar a los agentes con poder de decisión o a la comunidad local, por tanto no pudo observarse un beneficio tangible en la ciudad durante la duración del experimento. No obstante, se concluyó que se trata de una herramienta perfectamente útil para aplicar políticas y acciones como las citadas en el contexto del escenario.

## 5.4. Instanciación *cloud* en puertos marítimos para análisis de impacto medioambiental

### 5.4.1. Contexto

La industria marítima en su conjunto, y las actividades portuarias como principal *hub* logístico, pueden suponer un gran impacto ambiental en las zonas donde se alojan. Para efectuar sus operaciones (carga, descarga, manipulación de mercancías, transporte, estiba...) emplean maquinaria pesada de diversa índole que, en la mayoría de los casos, es movida por motores de combustión. Esta maquinaria emite un abanico de elementos contaminantes a la atmósfera, desde gases de efecto invernadero, materia particulada, sulfuros y otras sustancias que pueden causar efectos nocivos, tanto a las personas como al resto de biodiversidad de su entorno [485]. Esta circunstancia adversa ha sido evidenciada tanto por comunidades locales como por entidades oficiales, que sitúan la contaminación atmosférica como la principal prioridad de atención medioambiental en el sector logístico portuario [486].

Durante los últimos treinta años, los impactos ambientales de las actividades portuarias han recibido una atención creciente por parte de la comunidad científica, las instituciones y los operadores económicos.

Observando este hecho, desde hace unos años se trata de dar respuesta a la pregunta: *¿cómo avanzar en esta reducción sin perder los beneficios del transporte marítimo mundial sobre el que se estructura buena parte de la economía mundial?* [487]. En este sentido, una serie de iniciativas han surgido en la última década y media proponiendo estrategias y metodologías desde diferentes puntos de vista (PPRISM [488], PORTOPIA [489], entre otras).

No obstante, para poder atacar este problema debe primero conocerse qué es lo que se desea reducir y en qué orden (categorización y priorización) y, consiguientemente, poder medirlo. En relación a la categorización, confluyen varias circunstancias. En primer lugar, los puertos marítimos son entornos vastos donde existen multitud de agentes (autoridades portuarias, terminales, operarios, prácticos...) que tienen diferentes intereses y preocupaciones en cuanto al impacto medioambiental, y cuyas acciones pueden repercutir de diferentes formas. En segundo, existe un abanico muy amplio de metodologías que se han publicado y utilizado para esta catalogación de aspectos medioambientales relevantes; por citar algunas de las más empleadas: ABP [490], ECOPORT [491], o INDAPORT [492] conviven con algunas propias de determinados puertos, como Algeciras, Livorno o Valencia [493]. En este sentido, según un informe del sector [494], existen unas tremendas diferencias entre unos y otros, haciéndolos incomparables.

Adicionalmente, algunos puertos, en sus esfuerzos por ser eficientes desde el punto de vista medioambiental, para lo cual gastan ciertos recursos, consideran estas cuestiones como confidenciales y son reacios a compartir información, como parte de su plan de negocio en un sector altamente competitivo.

La realidad es que no existe una metodología estandarizada para la identificación ni para la integración de los aspectos relevantes de todas las operaciones portuarias que pueda utilizarse para abordar el rendimiento medioambiental y de sostenibilidad del puerto en general.

El resultado de los aspectos anteriores ha conducido, hasta ahora a una situación en la que los puertos emiten informes anuales, frecuentemente basados en datos cualitativos, de encuestas o altamente sesgados, para reportar su impacto medioambiental. Es por ello que se afirma que no existen soluciones formales y promovidas por actores neutrales que permitan (i) una catalogación y priorización de estos impactos, (ii) una metodología para su medición no basada en informes y en análisis cualitativos, (iii) una tecnología para soportar lo anterior.

En vista de esta necesidad, surgió el proyecto PIXEL (en el que el candidato ha participado activamente durante más de tres años), cuyo objetivo principal consistió en diseñar y aplicar una plataforma IoT para abordar este problema y proporcionar una metodología cuantitativa, estandarizada y basada en pruebas que pueda aplicarse a todos los puertos para evaluar su impacto medioambiental.



**Figura 5.29:** Planteamiento contextual de PIXEL de cara al escenario PEI

El proyecto, y por ende, este particular escenario de validación de la tesis doctoral, partió de la base de que los problemas medioambientales suelen ser multidimensionales, por lo que a los responsables de la toma de decisiones les resulta muy difícil actuar sobre un gran número de datos inconexos provenientes de fuentes



de datos diferentes y heterogéneas. Les resulta mucho más fácil actuar sobre problemas ambientales complejos atendiendo a una única métrica que integre todos los impactos ambientales que tiene un puerto.

Esta única métrica fue denominada **PEI (del inglés *Port Environmental Index*)** [484], y conforma la génesis de los planteamientos que llevaron a iniciar esta tesis doctoral. La idea central del PEI es concebir una metodología completa, estandarizada y transparente que sirva para integrar todos los aspectos ambientales significativos de los puertos y los impactos relacionados en un solo valor (entre 0 y 1). Además, en relación con el escenario de validación, esta metodología es acompañada por una tecnología (basada en técnicas IoT) alimentada por datos en tiempo real, lejos de lo que alcanzan las iniciativas actuales.

El hecho de crear y popular este índice permitiría obtener una serie de ventajas con respecto a la pregunta lanzada al inicio de la sección: (1) mostrar cómo los puertos se comportan en términos de impacto medioambiental comparando unos con otros – siempre considerando la variedad geográfica, de tamaño y de restricciones dentro del propio índice, (2) clarificando el hecho de que el impacto medioambiental es responsabilidad de varios agentes, particionándola y pudiendo tomar medidas acordes, (3) permitiendo la comparación de “rendimiento” en función del tiempo y (4) ofrece información fácilmente interpretable para el proceso de toma de decisiones operativas y de inversión en caso de que se registren valores excesivos de impacto negativo en el medio ambiente.

Para entender el impacto del candidato (por tanto, de la solución) en la construcción de este índice, debe tenerse en cuenta la participación intensiva en:

- 1.** El diseño del índice y subíndices.
- 2.** El desarrollo del servicio de cálculo de los índices (véase sección 4).

**3.** El despliegue (piloto del proyecto) en el puerto de Tesalónica. Este elemento innovador (PEI) fue validado durante el proyecto en cuatro puertos europeos (Pireo, Tesalónica, Burdeos y Monfalcone). El candidato participó en todas estas actividades, no obstante, puede entenderse el segundo caso (Tesalónica) donde las actividades técnicas fueron desarrolladas por el candidato y donde se pudo realizar la validación de la solución con mayores garantías. Por tanto, este es el escenario que se expone en la tesis doctoral.



**Figura 5.30:** Puertos de validación PEI (Pireo, Tesalónica, Burdeos, Monfalcone)

Este escenario de validación supuso el mayor reto para el candidato desde varios puntos de vista. En primer lugar, el encaje de la arquitectura propuesta dentro del sistema global del proyecto (véase la siguiente subsección) fue un desafío, encontrando la manera de demostrar la flexibilidad y escalabilidad de la solución. En segundo lugar, las funcionalidades a validar fueron amplias, empezando con el servicio de índice compuesto, que fue revisado y evaluado por la Comisión Europea en dos ocasiones, y continuando por la necesidad de generación de reportes y emisión de recomendación de acciones sobre los valores obtenidos. En tercer lugar, por la gran variedad de datos a conectar sobre un mismo entorno, los problemas de disponibilidad de información y la complejidad de diseño del índice. En cuarto lugar, la utilización de la solución por los agentes interesados, que debía incluir la posibilidad de comparación entre diferentes entidades (único escenario de la tesis doctoral con este requerimiento). Por último, las restricciones de privacidad y compartición de la información, que sirvieron para justificar la validez de los módulos de seguridad y privacidad planteados (sección 3.6).

Además, este escenario añadió un componente clave que contribuyó a la validación de la arquitectura. Para que los índices entre diferentes puertos puedan ser comparables, debe existir un mecanismo que garantice la “fiabilidad” de dichos índices. Este aspecto es delicado y puede abordarse de diferentes maneras (véase sección 4.1.2.3), no obstante, se materializó en el establecimiento de un índice compuesto adicional, paralelo al PEI, en el que se modelaba el origen de cada fuente de información alimentando al valor comparado, permitiendo así contextualizarlo.

## 5.4.2. Implementación de la solución

A continuación se exponen los hallazgos, decisiones y resultados obtenidos durante el escenario de validación asociado al índice PEI.

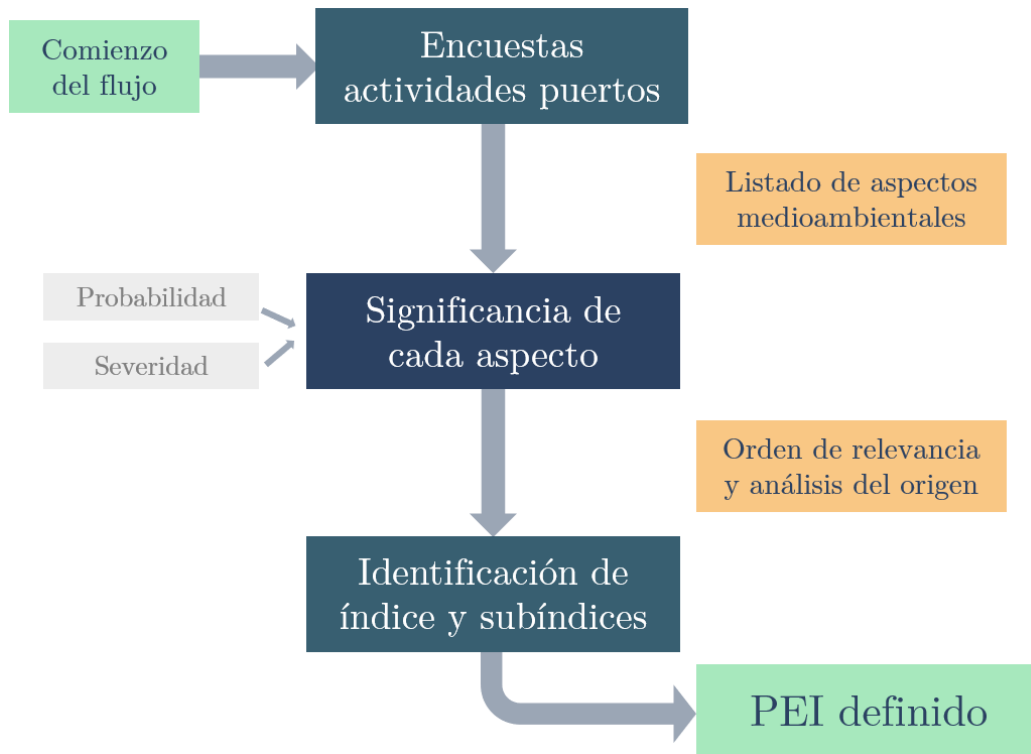
### *Fase I – Conceptual y servicio*

Al formar parte (esencial) de un proyecto de investigación europeo colaborativo, el índice PEI contó con la intervención de muchos agentes relevantes para su diseño. Como corresponde a una actividad dentro de una acción de investigación y desarrollo H2020 (RIA – del inglés, *Research and Innovation Action*), se abordó desde un enfoque principalmente científico, ahondando en la revisión de la literatura y las metodologías actuales.

En este sentido, se realiza a continuación una breve referencia de la metodología llevada a cabo para la (i) catalogación de los aspectos medioambientales relevantes en puertos marítimos y (ii) el diseño del índice y los subíndices pertinentes para la creación del índice PEI. Conviene destacar que esta acción no fue liderada por el candidato, sino participada con la idea del posterior desarrollo y validación del servicio de indicadores compuestos.



El flujo para la obtención de la estructura del índice PEI respondió al siguiente esquema:



**Figura 5.31:** Flujo de recopilación de literatura para el diseño del índice PEI

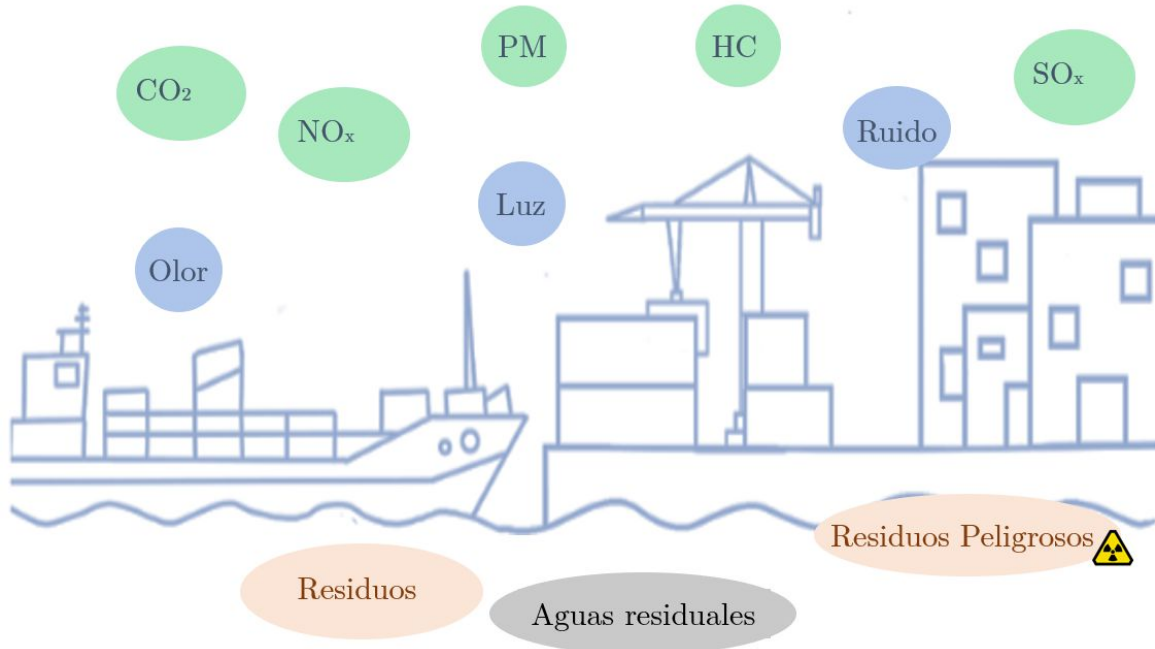
- 1) Comprensión de los aspectos medioambientales relevantes para los puertos:

A través de una serie de cuestionarios completados por los puertos del proyecto PIXEL (véase la subsección anterior), así como mediante la búsqueda en la literatura científica y técnica pertinente. Los aspectos medioambientales Los aspectos medioambientales que se consideraron significativos fueron las emisiones a la atmósfera emisiones de aguas residuales, emisiones acústicas, producción de residuos, contaminación lumínica contaminación lumínica y los niveles de olor. Más información sobre estas encuestas y sus resultados puede encontrarse en el informe D5.1 del proyecto PIXEL [495].

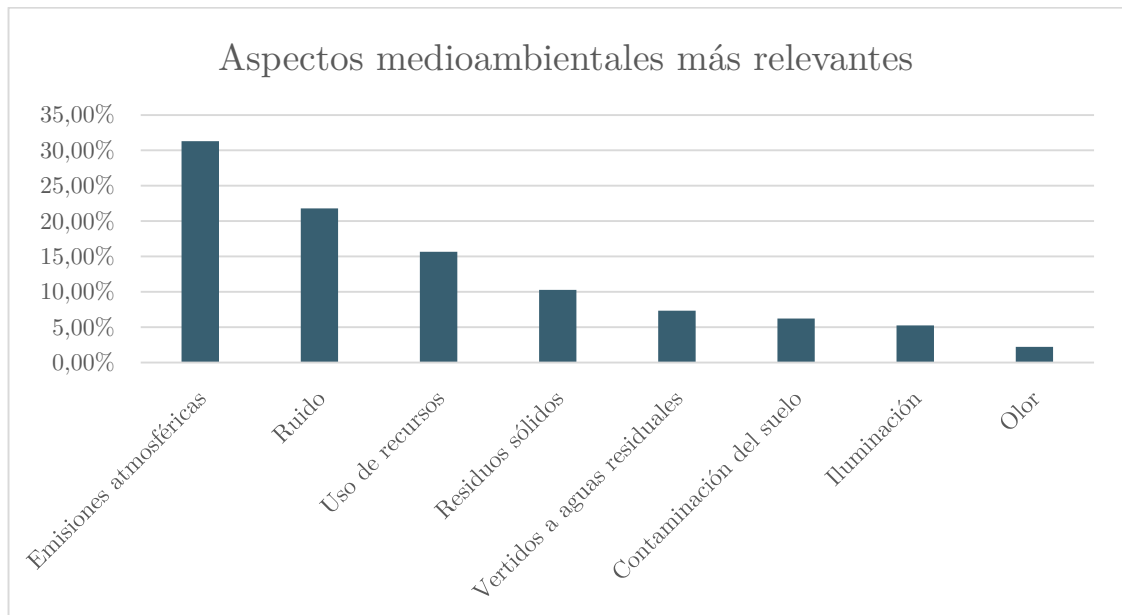
- 2) Asignación de la relevancia de cada uno de dichos aspectos:

Este aspecto fue realmente importante para la validación de la solución, ya que constituyó la base para la ponderación de los *cKPIs*). Tras la aplicación de un valor de “relevancia” en el caso (probabilidad) de que uno de estos elementos se manifestara en el puerto (significancia), se concluyó que las emisiones atmosféricas eran el elemento más relevante, seguido del ruido, residuos sólidos, vertidos a aguas residuales e

iluminación. Según una serie de argumentos expuestos, el uso de recursos pudo asociarse a residuos energéticos (aumentando la ponderación) y la contaminación del suelo y el olor fueron descartados de la construcción del índice.



**Figura 5.32:** Aspectos medioambientales en un entorno marítimo portuario

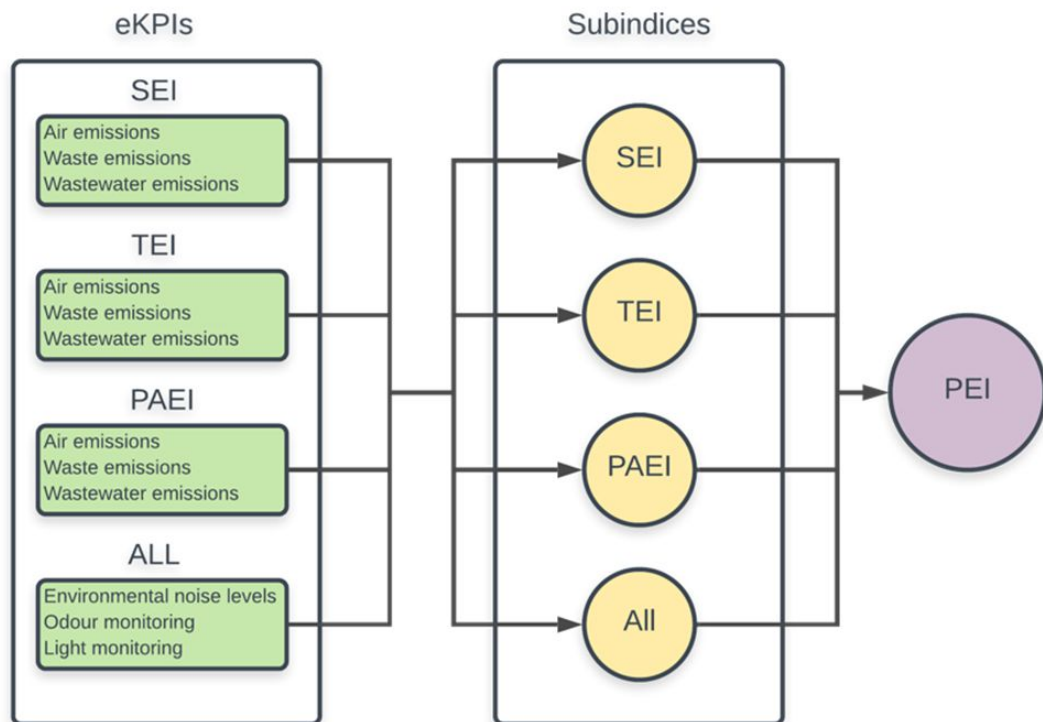


**Figura 5.33:** Gráfica que muestra los aspectos medioambientales más relevantes para los puertos marítimos según encuestas del proyecto PIXEL

3) Identificación de elementos primitivos y construcción de índices:

A partir de la información anterior, se realizó un análisis de qué elementos individuales componen cada uno de dichos aspectos relevantes (los *eKPIs* – del inglés *environmental Key Performance Indicators* – equivalentes a *cKPIs* en la solución de la tesis doctoral). Se identificaron considerando su importancia para cada aspecto anterior, la capacidad de ser medidos sistemas de información, por su representatividad y por su correlación. La tabla completa de los indicadores (*eKPIs*) diseñados se expone en el artículo publicado por la revista *Ecological Indicators* (Elsevier) titulado “*A novel approach for assessing the ports’ environmental impacts in real time – The IoT based port environmental index*”. Se trata de 43 elementos individuales que se categorizaron según dos ejes. El primero, directamente desde los tipos de aspectos medioambientales que cubrían (análisis anterior), como emisiones atmosféricas, etc. El segundo, según el tipo de origen dentro del puerto; es decir, según el tipo de actividades y a quién debían ser atribuidas. Según esta clasificación, se podían dividir en: provenientes de los barcos, de la terminal, de la Autoridad Portuaria o no ser atribuibles a ningún elemento de manera unívoca. Algunos de ellos se repiten para varios orígenes (por ejemplo, tanto los barcos como las actividades de las grúas generan contaminantes como CO<sub>2</sub>).

Partiendo de lo anterior, la estructura de cálculo se hizo evidente y se refleja en la siguiente figure:



**Figura 5.34:** Estructura de subíndices e índice en el escenario PEI

Siendo SEI (*Ship Environmental Index*) el subíndice creado para representar los eKPIs asociados a barcos (buques dentro del área del puerto, tanto atracados como maniobrando), TEI (*Terminal Environmental Index*) aquél asociado a las actividades de terminal logística (carga/descarga de mercancía, apilamiento, almacenamiento...), PAEI (*Port Authority Environmental Index*) modelando el impacto de la gestión del puerto (muy variable entre casos) y, por último, GEI (*Global Environmental Index*).

Contrariamente al escenario de validación anterior, en este caso fue en este punto cuando se procedió al análisis de los datos. Esto fue así debido a (i) el plan de trabajo del proyecto y (ii) a la necesidad de establecer un marco común para varios puertos para después proceder a instanciar la metodología en todos ellos. Igualmente, ya se había realizado un primer barrido de la información durante la actividad de encuestas para la captura de los aspectos medioambientales más significativos.

A continuación se muestran las fuentes de datos que se estudiaron para la instanciación y cálculo de los índices. Conviene recordar que solo se ha considerado el despliegue en el puerto de Tesalónica, donde el candidato intervino con más intensidad y eligió el escenario para la validación de la solución.

#### **Información sobre los barcos procesados (y a procesar) por el puerto:**

Apreciando que el mismo actor de interés participó en este escenario de validación (Autoridad Portuaria de Tesalónica), se hizo uso de la misma fuente de datos provista y abierta para el doctorando.



**Figura 5.35:** Fuente de datos sobre barcos a procesar por el puerto escenario PEI

Estos datos servidos por el departamento de IT del Puerto a través de una API REST. Internamente, estos datos son gestionados por un TOS (para los portacontenedores) y el software *Statistics DB* (para mercancía convencional). En esta fuente se pueden consultar los datos tanto de las operaciones pasadas de los buques (se dispone del historial de 2015 a 2019 y de 2020 hasta ahora) como de los buques que llegarán al puerto durante la próxima semana (escalas de buques). Al ser el PEI un cálculo retrospectivo, el cálculo en este escenario de validación sólo fue utilizando los datos pasados.

En cuanto a la información aportada por esta fuente, se encuentran: (i) la carga operada (carga/descarga) por esos barcos, (ii) información sobre los buques

(IMO, descripción, tipo de carga – contenido) y (iii) información temporal (fecha de llegada, fecha de partida, fecha de atraque, fecha de salida de muelle).

**Anexos MARPOL-II facilitados por la Autoridad Portuaria (AP):**

Dentro de la misma API expuesta por la AP, se ofrecieron datos de vertido de residuos.



**Figura 5.36:** Fuente de datos vertido de residuos escenario PEI

El funcionamiento de esta fuente de datos es razonablemente similar a la anterior. Cabe mencionar que, cuando un barco se dispone a atracar en un puerto para operar mercancía o para hacer escala (mediante un “anuncio”), informa al puerto receptor de si desea verter residuos que el puerto debe gestionar. Este tipo de residuos puede ir desde plásticos, aceites, basura del barco hasta deshechos de pesca o materia inerte (si es el caso). Esta información se le proporciona al puerto a través de los llamados anexos MARPOL [496]. En el puerto de Tesalónica existe un sistema digitalizado que recupera esta información y la expone a través de esta API una vez por mes (datos de los residuos gestionados durante el mes anterior).

**Informes de consumo eléctrico y otras actividades:**

Una forma relevante de calcular el indicador de uso de recursos y generación de residuos por parte del puerto es a través de su consumo eléctrico. Existe una relación directa entre la utilización de energía en el puerto y el impacto de sus actividades. De este modo, se aprovecharon los datos de informes y facturas de consumo energético y de gas para ser incluidas en el cálculo del CI de este escenario. En la Figura 5.39: Reportes anuales de ruido escenario PEI se puede observar el tipo de información aportada. Este formulario web se preparó e incluyó dentro de la interfaz de la solución para la inserción de información que no se encontraba disponible en el API (Figura 5.37: Formulario web datos de consumo eléctrico PEI).

**Figura 5.37:** Formulario web datos de consumo eléctrico PEI

### Herramienta de reporte de generación de residuos y vertidos en aguas residuales:

De nuevo, la API expuesta por la AP incluye información sobre los residuos generados por la terminal. Este punto de acceso devuelve información del momento en que se produjo esta generación de residuo, la cantidad en kilogramos y su catalogación siguiendo el estándar EWC (*European Waste Catalogue*) [497].

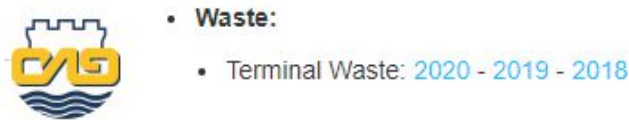


Figura 5.38: Gestión de residuos terminal portuaria escenario PEI

Del mismo modo que la fuente de datos anterior, esta información es actualizada una vez al mes por parte del puerto.

### Informe de ruido:

Aunque durante el escenario de validación se seleccionó una fuente de datos en tiempo real para la obtención de valores de ruido (y de iluminación – ver a continuación), se incluyeron también una serie de informes sobre el ruido en el puerto para permitir un cálculo (menos precisos – ver “índice de fiabilidad”) retrospectivo del PEI de años anteriores para el puerto de Tesalónica. Esta fuente de datos consistió en informes anuales hechos durante 24 horas una vez al año en diferentes puntos del puerto.

Se trata de mediciones fiables pero poco representativas para el transcurso de un año, únicamente orientativas sobre datos medios.

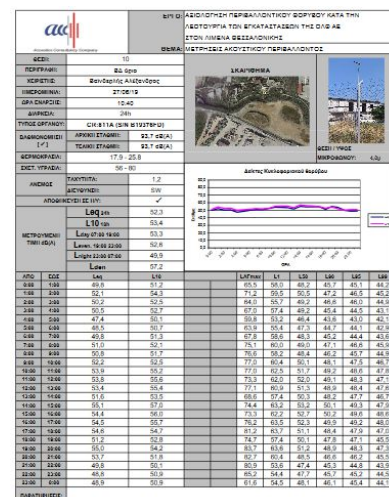


Figura 5.39: Reportes anuales de ruido escenario PEI

### Estación de sensores de ruido e iluminación en tiempo real:

Dentro de las actividades del doctorando en este escenario se incluyó la búsqueda de sensores para la captura de la información correspondiente al resto de indicadores (ruido, iluminación, olor). En este sentido, se exploraron opciones como U-Sound de Urbiotica, IoTSoundSensor de SensorTeam, enLight Noise, IoTSens, In\_noise de Intelkia o Kunak Noise para obtener datos de ruido y otros dispositivos como CitySensePlus de Tvilight, WS1 Pro de Ubibot o IoTLRW de NCD. Aunque el olor se descartó, formó parte de la búsqueda inicial, explorando opciones como los modelos XP-329IIIR o V819 de New Cosmos o MSS de MSSForum. No obstante, finalmente se decidió por la adquisición del sensor SmartSpot de la empresa española HOPU[498]. Las razones principales por las que se eligió fueron: (i) alta



capacidad de customización y adición de módulos software y hardware, (ii) incluye varios sensores dentro del mismo dispositivo, entre ellos sonido e iluminación, interesantes para este escenario y (iii) rango de precios competitivo (alrededor de 2.000€).



**Figura 5.40:** Estación sensores ruido y luz para escenario PEI

En cuanto a sus características técnicas, posee un encapsulado IP65 y ofrece conectividad WiFi, Ethernet y GPRS. Equipa un sensor de sonido que capta señales entre 40 y 115 dB con una desviación media de 0,1 dB. Incluye también un sensor de luz OPT3001 de Texas Instruments [499] capaz de proveer datos de iluminación entre 0,01 y 83k lux, con una desviación de 0,01 lux operando entre temperaturas desde -40°C hasta 85°C.

Un aspecto relevante dentro de este escenario fue el punto de instalación de la estación. Si bien es cierto que para el caso de las emisiones atmosféricas no fue posible contar con sensores (los niveles de contaminantes en un punto del tiempo pueden deberse a una multitud de factores no relacionados unívocamente con las actividades portuarias y requieren la replicación de condiciones e ideales o la introducción de compensación – por tanto, se estarían tomando medidas y acciones sesgadas [500]), el aspecto sonoro es diferente. Durante el desarrollo de este experimento, se descubrió que podría elegirse un punto óptimo de instalación de los sensores para que los valores proporcionados por éstos fueran válidos. Para ello, se realizó un estudio de dispersión de ruido del área del puerto de Tesalónica. Utilizando la herramienta *Predictor Lima* [501], y tras la introducción manual de información del puerto tales como terreno, edificios, fuentes de generación de ruido y pantallas, se procedió a ejecutar varias simulaciones. El resultado fue un mapa de dispersión del ruido a partir del cual se seleccionó un punto óptimo para su instalación. El sensor funcionando en el Puerto de Tesalónica se muestra a la izquierda en la Figura 5.40: Estación sensores ruido y luz para escenario PEI.

A continuación se muestra una tabla resumen de los datos que se utilizaron para este escenario de validación. Esta información fue determinante a la hora de desarrollar los conectores (siguiente subsección) y para definir los métodos matemáticos que guiarían el cálculo del CI.

**Tabla 5.10:** Tabla resumen de datos en el escenario PEI

| Fuente de datos     | Conectividad       | Parámetros relevantes                   | Frecuencia de muestreo                  | Unidades                | Histórico  |
|---------------------|--------------------|---|---|-------------------------|------------|
| Barcos procesados   | REST API (HTTP)    | IMO<br>Mercancía<br>Toneladas<br>Fechas | Tiempo real (cuando se anuncian barcos) | Toneladas<br>DD/MM/YYYY | Desde 2015 |
| Anexos MARPOL       | REST API (HTTP)    | Contenido anexos                        | 1 vez al mes                            | Toneladas               | Desde 2016 |
| Consumos eléctricos | Formularios web    | Energía<br>Gas                          | 1 vez al mes                            | Kwh<br>l                | Desde 2012 |
| Reporte de residuos | REST API (HTTP)    | Residuos por tipo                       | 1 vez al mes                            | Códigos EWC<br>Kgs      | Desde 2018 |
| Informes de ruido   | Reportes estáticos | Ruido en varios puntos                  | 1 vez al año durante 24h                | dB                      | Desde 2019 |
| Estación sensores   | MQTT               | Lden, Lnight<br>Iluminación             | 1 por segundo                           | dB<br>lux               | Desde 2021 |

Realizando una reflexión sobre los datos disponibles, se trata de uno de los casos más optimistas entre los puertos que formaron parte del proyecto (la digitalización en puertos, especialmente pequeños y medianos, dista mucho de lo que cabría esperar [502], quedando todavía un largo camino por delante). En este escenario, la mayoría de valores podían ser extraídos de forma automática, disponiendo incluso de fuentes IoT como el sensor de luz y sonido. No obstante, la información no es suficiente ya que muchos datos (como se documenta en la siguiente subsección) tuvieron que realizar una serie de asunciones y obtención de información de bases de datos externas para ser completadas.

Basado en lo anterior, tal y como se especifica en la metodología, debió decidirse la frecuencia con la que deseaba calcularse el servicio de cálculo del índice compuesto. Acorde a los datos, la entrada más restrictiva hubiera sido el reporte de ruido (una vez al año). No obstante, esta medida solo se utilizó para las primeras anualidades (2016, 2017) y se asumió un ruido constante durante los meses del año. En la aplicación real del índice, esta fuente fue sustituida por el sensor en tiempo real, por tanto, el PEI en Tesalónica se planteó como una herramienta retroactiva que muestra el impacto medioambiental del pasado mes.

Por último, para completar esta fase, se realizó una reflexión sobre cuáles debían ser los métodos matemáticos a escoger de entre los disponibles en cada paso (véase sección 4.1.2).



**Tabla 5.11:** Elección de métodos matemáticos para el escenario PEI

| Gestión de datos ausentes:  | Media | <i>Unconditional mean</i> |   |                        |
|---|-------|---------------------------|---|------------------------|
| <p>En relación a los datos ausentes, este aspecto fue considerado como de menor relevancia en comparación con los otros métodos. La opción <i>hot deck</i> a través de la imputación del último pudiera ser válido pero fue descartado en favor de la media incondicional. Esto se decidió debido a que, en general, se tratará de <i>sets</i> de datos lo suficientemente grandes (serie temporal con muchos registros) como para afirmar que el valor medio no afectará a la interpretación del CI final, no obstante si el valor último hubiera sido un valor atípico esto podría cambiar.</p>   |       |                           |   |                        |
| Normalización:  | 1     | Por cargo                 | 2 | Distancia a referencia |
| <p>El escenario de validación PEI es uno de los más especiales ya que ha sido el único en que la efectiva comparación del resultado de los índices entre varias entidades equivalentes ha sido necesario. Esto añade una complejidad adicional al cálculo del CI en cuanto a normalización se refiere. En los escenarios individuales, los valores obtenidos de indicadores (<i>cKPIs</i>) pueden ser directamente normalizados siguiendo el método elegido (su rango de valores será similar o, en todo caso, sus diferencias deberán plasmarse equivalentemente entre 0 y 1). Esto sucede porque la entidad siempre es la misma. No obstante, las entidades en este caso son heterogéneas, y los valores prenormalizados de los indicadores serán muy variables. Este hecho se ilustra mejor con un ejemplo. El puerto del Pireo genera muchos más residuos (por su tamaño) que el puerto de Monfalcone. Por tanto, no sería correcto normalizar los valores de estos indicadores (por ejemplo, 5.000 toneladas de vertidos de un puerto grande versus 1.000 toneladas de vertidos de un puerto pequeño). Como se ha comentado en la sección 4, para que índices sean comparables tienen que haber sido normalizados contra el mismo valor y con el mismo criterio. Por tanto, se hizo necesaria la introducción de un paso previo para eliminar este sesgo y homogeneizar los valores previamente a la normalización. Según el informe EIAP (del inglés <i>Environmental Impact Assessments of Ports</i>) [503], esta situación se ha planteado anteriormente, y los criterios que suele utilizarse es normalizar los puertos según toneladas de mercancía manipulada, según TEUs (los dos anteriores son valores equivalentes), número de empleados o número de operaciones procesadas. En el caso de este escenario de validación, se decidió realizar una prenormalización según toneladas procesadas por el puerto.</p> <p>En cuanto a la normalización propia del servicio, se siguió el siguiente razonamiento: (i) el método de estandarización (<i>Z-scores</i>) conducía a valores negativos y positivos de los índices, lo que dificultaba su interpretabilidad. (ii) El método de reescalado (<i>min-max</i>), aunque permite la comparación entre entidades se desvía del objetivo de tener una referencia común sobre la que basar la interpretación. Esto significa que, si los <i>cKPIs</i> cambian con el tiempo, los valores del IPE no serán comparables a lo largo del tiempo, ya que la escala de los <i>cKPIs</i> habrá cambiado. Por tanto, se escogió el método de distancia a una referencia. Para implementar esta normalización, se hace uso del módulo específico contemplado en la arquitectura de la solución propio para este servicio (véase sección 4.2.2). A partir de una información inicial extraída del análisis de los</p> |       |                           |   |                        |

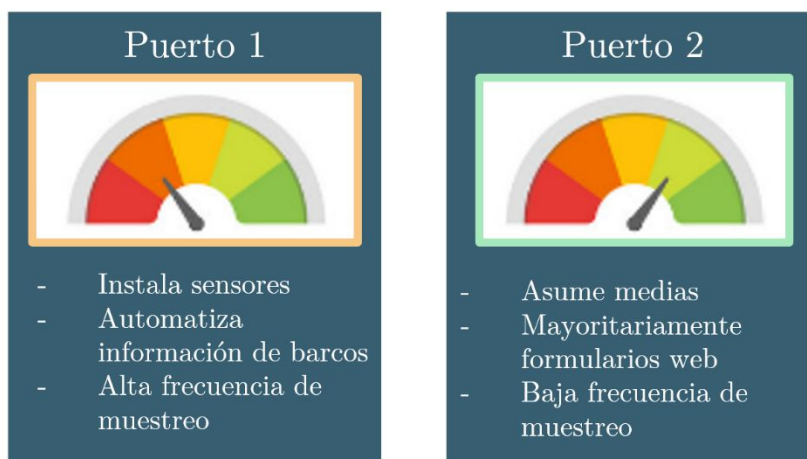
|   |                   |                                     |
|---|-------------------|-------------------------------------|
| <p>datos, se estableció un valor de referencia por cada indicador (<i>cKPI</i>, en este caso <i>eKPI</i>), y se normalizaba contra el mismo. Con el paso del tiempo, para mantenerse válida, esta referencia debía actualizarse, por tanto se adquiría el valor mínimo obtenido en cada <i>eKPI</i> de todos los puertos en el último periodo y se establecía como la nueva referencia.</p>   |                   |                                     |
| <b>Ponderación:</b>   | <i>PEI</i> -----> | <i>Custom</i> : Según significancia |
|   | <i>RR</i> ----->  | <i>BAM</i> – Según expertos         |
| <p>Al comienzo de esta subsección se ha explicado el procedimiento seguido para encontrar qué aspectos medioambientales eran más relevantes para los puertos, acompañados de una revisión de la literatura. El resultado de este procedimiento fue la obtención de unos porcentajes de relevancia (significancia) para cada uno de ellos. Esta información se utilizó directamente en este proceso de ponderación. Aunque un método de pesos igualitarios hubiera sido factible (el análisis de sensibilidad e incertidumbre lo corroboraba), se apostó por la utilización de estos “factores de relevancia” para ponderar los subíndices asociados (emisiones atmosféricas, ruido, etc.). Los pesos de primer nivel (sobre indicadores – <i>eKPIs</i>, se mantuvieron iguales). Debido a su naturaleza (miembros pertenecientes al sector otorgaron un peso/relevancia a cada aspecto, se consideró este método como equivalente al de asignación de presupuesto (BAM).</p> <p>En este punto conviene destacar que se realizó un segundo ejercicio de ponderación. Esta vez sí, utilizando el método de asignación de presupuesto (BAM), se generó un nuevo juego de pesos para el cálculo del índice de fiabilidad. La explicación de esta actividad se encuentra en la siguiente página.</p> |                   |                                     |
| <b>Agregación:</b>  | Aditiva           |                                     |
| <p>La introducción del método geométrico, según el estudio realizado, conducía a un CI matemáticamente indefinido en algunos casos. Observando la fórmula para este método, se concluye que la expresión da un resultado válido si el parámetro <i>c</i> es impar. En este caso, no se puede garantizar este hecho. Además, en caso de que pudiera emplearse (<i>c</i> impar), los resultados empleando agregación geométrica conducirían a una difícil interpretación, ya que su estructura lleva a que (si se utiliza la normalización por referencia), los valores más cerca del óptimo fueran más pequeños, descompensando exageradamente la existencia de indicadores peores. En otras palabras, aún con un Sistema de pesos equilibrado, la existencia de un solo indicador cercano al óptimo (referencia) eclipsaría el efecto negativo de otros parámetros (aunque estos fueran más).</p>   |                   |                                     |

Para la toma de estas decisiones (realizada por el candidato junto con otros equipos de entidades de investigación) se siguió una estrategia de análisis comparativo de combinaciones, precedido de un PCA (del inglés *Principal Component Analysis*), un análisis de incertidumbre ante las posibles combinaciones y un análisis de sensibilidad de los datos. Todas estas evidencias (cuyo alcance excede los objetivos de esta tesis) pueden encontrarse en el entregable D5.3 del proyecto PIXEL.

Como elemento adicional de este escenario de validación, se detectó la necesidad de incluir un índice de fiabilidad del resultado del índice.

La idea del PEI es basarse cuanto más posible en datos automatizados y en tiempo real (concepto y técnicas IoT) desde fuentes precisas. No obstante, no todos los puertos tienen el mismo nivel tecnológico ni equipos de, por lo que es más difícil obtener y observar los mismos parámetros. Como se explica en la sección 3, la solución está preparada para aceptar indicadores desde diferentes fuentes de datos, pero las mismas no pueden/deben ser consideradas iguales a la hora de interpretar el resultado del índice. En aras de la claridad, permítase al lector concebir un ejemplo: no es lo mismo recuperar la cantidad de aguas residuales de una red de sensores instalados en la red de tuberías (mucho mejor) que introducir a través de un formulario los valores estimados de dicho vertido (mucho peor)

A la hora de comparar...



¿Cómo representar de una forma cuantitativa estos hechos?

**Figura 5.41:** Contextualización de índices

Para ello, tecnológicamente se hizo uso del módulo de contextualización de índices diseñado y desarrollado según la sección 4.2.3. Como preparación, debió encontrarse un mecanismo para establecer cuán fiable/cercano al nivel de automatización deseado es cada fuente de las posibles utilizadas para cada tipo de dato.

En este sentido, el candidato realizó un ejercicio de entrevista/encuesta con una serie de expertos en diferentes campos para que aportaran su opinión al respecto utilizando precisamente la metodología BAM, como sigue:

- Se seleccionó a 15 expertos tanto en aspectos tecnológicos (IoT, Big Data, sistemas de información) como dentro del sector (portuario, logístico, transporte...).
- Se preparó un cuestionario dividido en dos fases:

- [1] *¿Cómo de importante es cada uno de los siguientes aspectos/características en una fuente de datos con respecto a las otras en un entorno portuario?*

Los expertos debían otorgar un “presupuesto” a cada una de ellas teniendo un total de 20€ a repartir. Las opciones dadas y los resultados obtenidos fueron:

**Tabla 5.12:** Resultado primera pregunta BAM en escenario PEI

| Característica                             | Presupuesto medio |
|--|-------------------|
| Cercanía a tiempo real                     | 2,9               |
| Precisión                                  | 3,7               |
| Automatización                             | 3,1               |
| Bajo coste                                 | 3,1               |
| No necesidad de intervención humana        | 3                 |
| En conformidad con las políticas de puerto | 3,6               |
| Validada por usos previos (consolidada)    | 2,2               |
| Desacoplada de efectos en cascada          | 2,3               |

- [2] *¿Cómo de valioso es cada tipo de obtención de información para la fuente de datos [X] en comparación con el resto?*

Cada experto debió rellenar una pregunta por cada fuente de datos [X] (por ejemplo, valores de ruido) – un total de 14. Los expertos debían otorgar un “presupuesto” a cada una de ellas teniendo un total de 20€ a repartir. Las opciones dadas (formas de obtención) fueron: sensor, API en tiempo real, API periódica, herramienta propia de PIXEL intermedia, datos externos, formulario manual o valor extraído de la literatura. Se estableció una relación de cuánto cada una de esas opciones respondía a las prioridades establecidas tras la pregunta 1, y conforme a los resultados 14 preguntas, se establecieron los pesos en cada indicador y subíndice para el cálculo del índice de fiabilidad. Para la obtención de estos pesos, se asumió que, para cada dato necesario (por ejemplo, iluminación), el tipo de obtención con el presupuesto medio más alto (por ejemplo, API en tiempo real) tendría un peso de 1. En el Apéndice B puede encontrarse una serie de reflexiones realizadas por el candidato y compartidas para la realización del BAM.

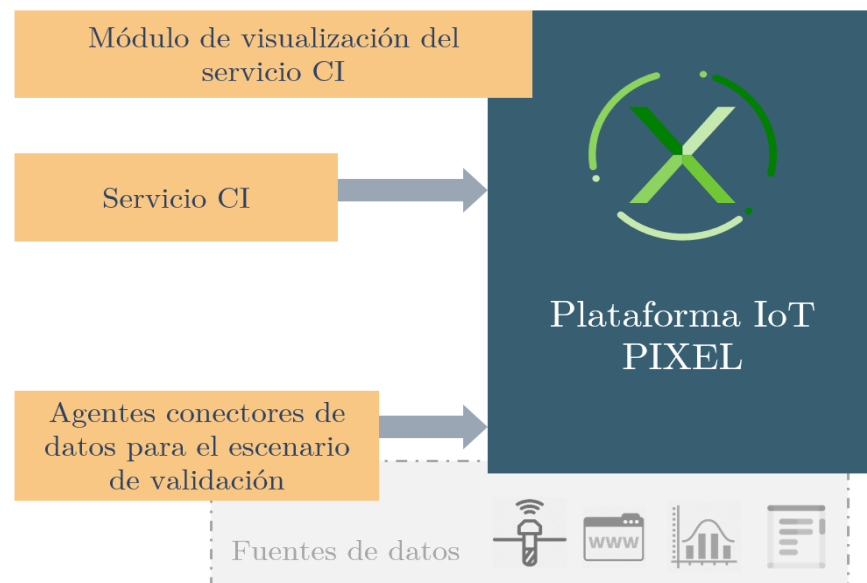
Las entrevistas/encuestas realizadas para esta actividad cumplieron todas las consideraciones de privacidad de datos, tratamiento de datos personales y aspectos éticos siguiendo la normativa de la Unión Europea y ciñéndose a los mecanismos aprobados dentro del proyecto PIXEL.

*Fase II – Arquitectura y conectores*

Para comprender el papel de la solución de esta tesis dentro del desarrollo del PEI (por tanto, dentro de la validación en el proyecto PIXEL), deben considerarse las siguientes cuestiones:

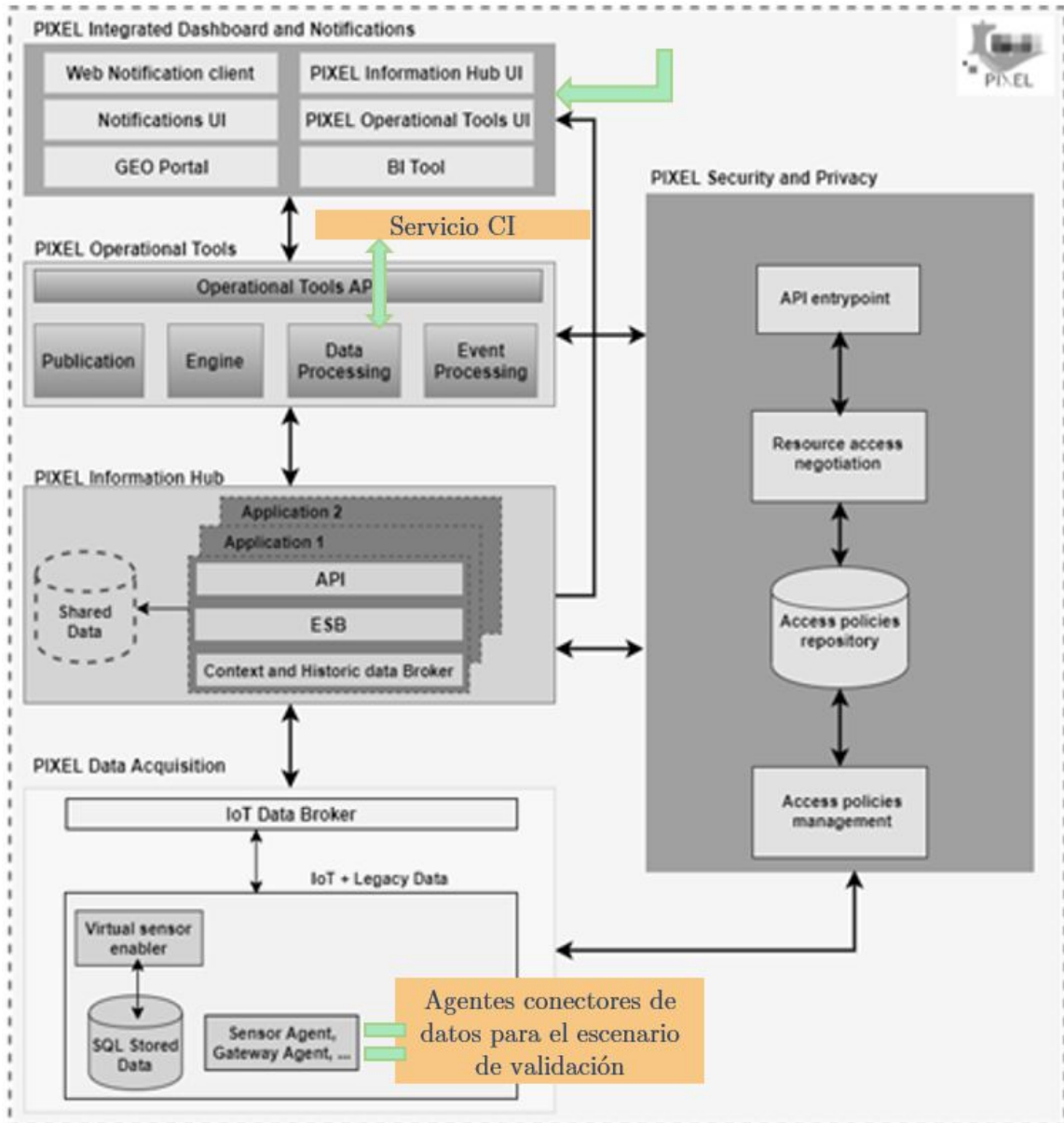
- El candidato participó activamente en el diseño y desarrollo de la plataforma IoT del proyecto PIXEL.
- La plataforma PIXEL es más compleja e intrincada que la solución propuesta en esta tesis.
- Algunos módulos de la arquitectura de la solución del doctorando formaron parte intrínseca de la plataforma global de PIXEL (por ejemplo, la visualización del PEI, objeto de estudio de esta sección).
- El diseño de la plataforma PIXEL se realizó previamente al comienzo de esta tesis doctoral, por tanto la solución propuesta se encuentra fuertemente inspirada por ésta.
- El modelo PEI (cálculo del índice compuesto medioambiental del proyecto) fue desarrollado por el doctorando y sirvió como base para el desarrollo del servicio (más general) expuesto en la sección 4.
- La ejecución de uno de los pilotos (de un total de 5) en uno de los puertos (fueron 4) fue lo que puede considerarse como caso de validación de la solución de esta tesis.

Por todo lo anterior, puede decirse que para este escenario de validación solo puede considerarse que se realizó una instalación parcial de la solución. Por ello, en este escenario no pudo completarse toda la metodología expuesta en la sección 5.2. Un reflejo de la inserción parcial de la solución dentro de la plataforma PIXEL se ilustra en la siguiente imagen:



**Figura 5.42:** Encaje de la solución sobre plataforma PIXEL escenario PEI

Analizado desde otro punto de vista, en la siguiente figura se muestra una visión de la arquitectura de la plataforma PIXEL, indicando los puntos donde los módulos de esta tesis doctoral fueron empleados/insertados/integrados.



**Figura 5.43:** Integración de la solución sobre plataforma PIXEL escenario PEI

Por tanto, a la hora de evidenciar la validación de esta tesis en este escenario, es conveniente enfocarla desde los tres puntos de interacción:

En primer lugar, describiendo la implementación de los agentes de conexión con datos requeridos para el escenario PEI:





**Agente de obtención de datos de residuos sólidos y líquidos para el subíndice SEI (asociado a barcos):**

Desde la misma API, este conector de datos se encarga de introducir 14 indicadores asociados a los residuos procesados por el puerto generados por los barcos operados.

Los datos se ofrecen en un formato JSON específico, con una entrada por cada barco que ha requerido al puerto procesar residuos. En este sentido, la figura al margen derecho representa un ejemplo de la información recibida. Estas entradas contienen datos sobre el barco en particular (nombre, descripción) y exponen después una serie de campos ordenados alfabéticamente que representan los residuos según los anexos Annex I, Annex II y Annex V de la especificación MARPOL. No es el objetivo de este trabajo profundizar sobre este contenido, por tanto baste comentar que incluyen información relativa a, por ejemplo, residuos plásticos, de comida, residuos domésticos, aceites, aguas de lastre sucias, aguas residuales, sustancias nocivas, etc.

La lógica efectuada por el agente consisten en recupere los datos de esta API web, filtrar los datos para el periodo de tiempo correspondiente (según acordado en fase I, mensualmente), analizar el significado de cada campo asignándolos a las filas y columnas de referencia, hacer la suma de ciertos residuos para ese periodo específico, recuperar la información de la mercancía procesada por el puerto en ese periodo (prenormalización) e imputar este dato para su inserción como indicador. La fórmula utilizada es la siguiente:

$$E_T = \sum \frac{C_i}{t} \tag{19}$$

Siendo  $C$  la cantidad del residuo (en toneladas) y  $t$  el periodo.

**Figura 5.45:** Formato de datos residuos de barcos escenario PEI

```

"5": {
  "A": 6,
  "B": "MSC",
  "C": "MSC INGE",
  "D": "01-08-18",
  "E": "01-08-18",
  "F": null,
  "G": "1.500",
  "H": "0.300",
  "I": "0.100",
  "J": "0.400",
  "K": null,
  "L": "0.100",
  "M": "1.500",
  "N": null,
  "O": null,
  "P": null,
  "Q": null,
  "R": null,
  "S": null,
  "T": null,
  "U": null,
  "V": null,
  "W": null,
  "X": null,
  "Y": null,
  "Z": null,
  "AA": null,
  "AB": null,
  "AC": null,
  "AD": null
},
    
```

**Agente para procesar consumo eléctrico y gas de terminal y autoridad portuaria para índices de emisiones atmosféricas en PAEI:**

Se trata de una agente que, con una periodicidad mensual, provee 9 indicadores partiendo de la información de facturación eléctrica y de gas. Siguiendo una serie de indicaciones según el proyecto PIXEL, existe una serie de mecanismos para inferir las emisiones atmosféricas de unas instalaciones y maquinaria (terminal y Autoridad Portuaria) a partir de su consumo de recursos (energía y gas). En este sentido, Tesalónica tiene una serie de subestaciones manejando ambas energías, cuyos datos se encuentran igualmente expuestos en la API global disponible.



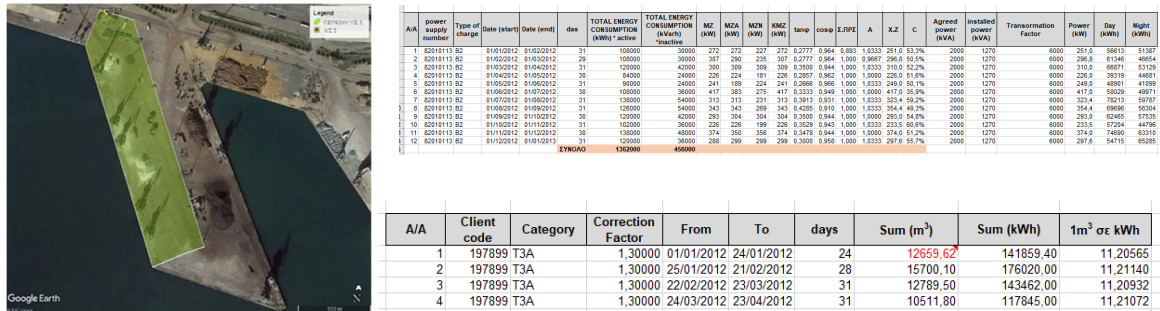


Figura 5.46: Datos de consumo eléctrico y gas para emisiones PEI

El procedimiento es equivalente al anterior agente, interpretando la información de todas las subestaciones para el periodo de tiempo determinado, obteniendo la información de cada campo y aplicando las siguientes fórmulas:

$$E = Pob * Potencia * Actividad * LF * EF \quad (20)$$

Para el caso de la energía eléctrica,  $E$  son las emisiones en gramos,  $Pob$  representa la población de maquinaria en el puerto generando ese consumo, Potencia media de los motores eléctricos usados, horas de actividad y los factores de emisión según unas tablas concretas.

$$C = C_1 \cdot \left(\frac{dim_1}{td_1}\right) + \dots + C_n \cdot \left(\frac{dim_n}{td_n}\right), \quad E_{ind} = AR_{consumo} \cdot EF_{ind} \quad (21)$$

En el caso del gas, se relaciona el consumo por subestación con los factores de emisiones de cada elemento contaminante (representado como indicador).

### Agente asociado a los indicadores de residuos generados por la terminal – TEI:

Se trata de un agente que, mensualmente, genera 3 indicadores relacionados con el vertido de residuos por parte de la terminal y Autoridad Portuaria. Como se ha mencionado en las evidencias de la fase I, la información (también desde el API expuesto) viene categorizada por los códigos EWC y expresada en kilogramos. En este caso, no debe aplicarse ninguna fórmula especial para popular los indicadores, ya que se trata de una asociación directa (siempre prenormalizando sobre la cantidad de mercancía operada por el puerto en dicho periodo).

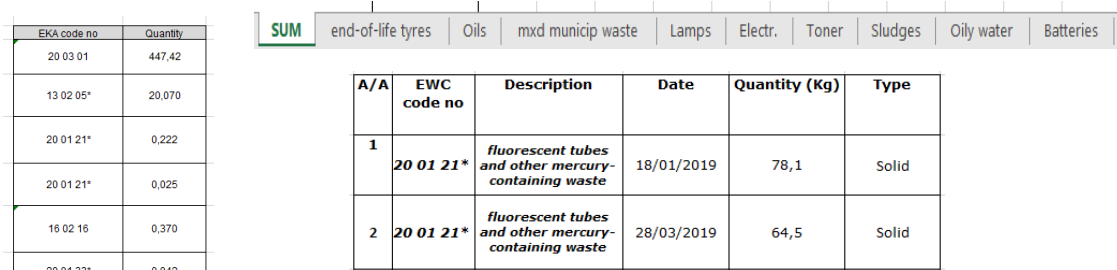


Figura 5.47: Datos de residuos generados por terminal en escenario PEI

**Agente para obtención de datos de ruido y luz desde estación de sensores:**

Como último agente necesario para la validación del escenario, se realizó un software para conectar los datos de la estación de sensores con la plataforma IoT en el formato esperado para el cálculo del CI.

Se analizaron dos opciones, y se procedió a la conexión MQTT con el sensor utilizando uno de los protocolos más empleados para la conectividad de sensores IoT. La siguiente figura muestra ambas opciones, siendo la primera referida al cloud web ofrecido por el proveedor del sensor (protocolo LwM2M [504] posteriormente conectado a una interfaz web) y la segunda conectividad directa mediante MQTT.



**Figura 5.48:** Opciones de conectividad para implementar el agente de ruido y luz

Es conveniente destacar que se realizó una conexión directa con los datos, que proveían una frecuencia de actualización de 1 segundo. Estos datos se conectaron siguiendo el modelo de datos *NoiseLevelObserved* propuesto por FIWARE. No obstante, la frecuencia que se decidió para el envío del indicador fue diaria. Para ello, debía aplicarse la siguiente fórmula que convierte desde los tres valores ofrecidos por la estación  $L_{day}$  (entre las 7h las 19h),  $L_{evening}$  (entre las 19h y las 23h) y  $L_{night}$  (entre las 23h y las 7h) a  $L_{den}$  (valor objetivo).

$$L_{DEN} = 10 \cdot \log \left[ \frac{12}{24} \cdot 10^{\frac{L_{day}}{10}} + \frac{4}{24} \cdot 10^{\frac{L_{evening}+5}{10}} + \frac{8}{24} \cdot 10^{\frac{L_{night}+10}{10}} \right] \quad (22)$$

En cuanto a los valores de iluminación (luxes) estos son directamente interpretados hacia el indicador.

Para este agente, un *broker* MQTT (implementado con *Mosquitto* [505]) se integró dentro de la implementación y, a partir de la suscripción del código a los *topics* determinados, se construyen los indicadores necesarios.

En la Figura 5.48: Opciones de conectividad para implementar el agente de ruido y luz puede apreciarse cuáles son los *topics* en los que se emite la información.

En segundo lugar, el servicio desarrollado en la sección 4, fue ejecutado en el seno de la plataforma PIXEL a través de un módulo avanzado propio de esta arquitectura llamado *Operational Tools* (OT).

El módulo OT es, en esencia, un orquestador de servicios, materializados en PIXEL en modelos de simulación y algoritmos predictivos. Se trata de un software desarrollado dentro de dicho proyecto que define un conjunto de herramientas de alto nivel (publicación, edición, ejecución y programación) sobre estos servicios, facilitando así su uso y la gestión de eventos asociados como alarmas y notificaciones. Tecnológicamente, el software (desarrollado en Java) interactúa con la API de Docker del entorno donde se ejecuta (*host*), siendo así capaz de ejecutar contenedores desde imágenes (previamente disponibles en DockerHub [506], [507]) a partir de unas reglas y una nomenclatura específica. Las opciones y funcionalidades de este módulo son muy vastas, pero quedan fuera del alcance de este trabajo. Existe documentación que puede encontrarse en repositorios públicos [507].

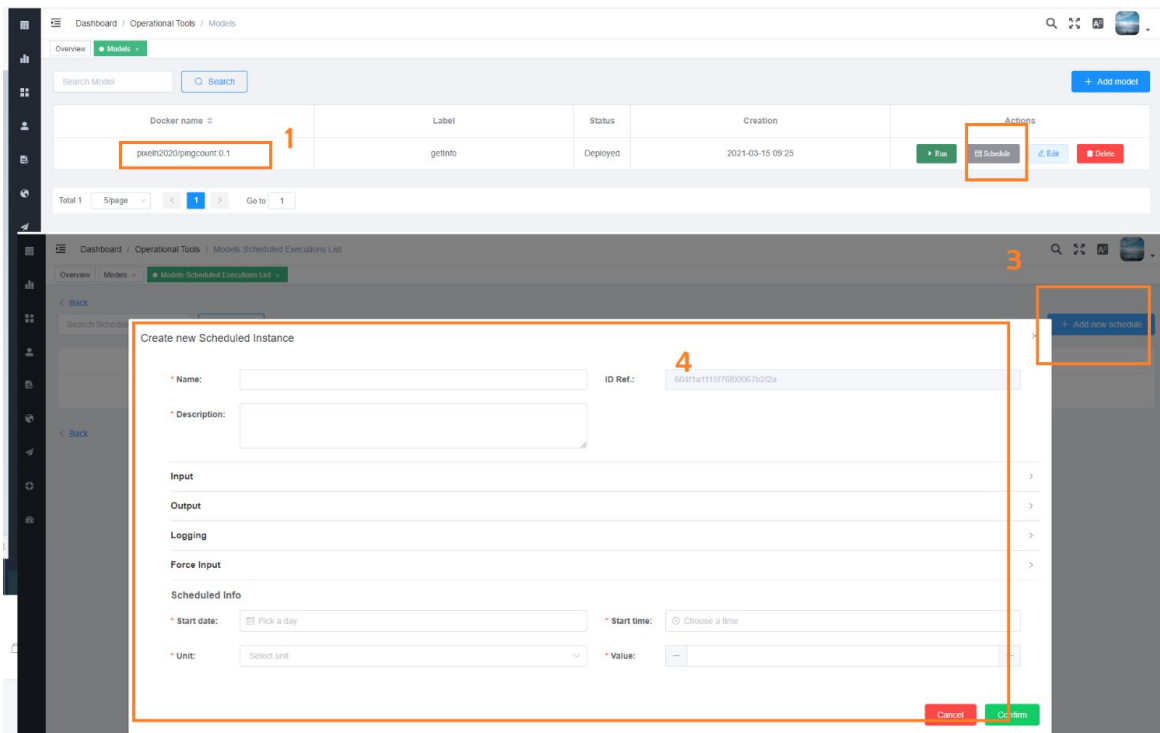
Las OT se encargan, pues de ejecutar “modelos”, que son piezas de software encapsuladas como imágenes Docker que pueden llamarse desde su API. Es así como ha encajado el servicio de cálculo de CIs de la tesis doctoral con la plataforma PIXEL. Siendo distribuido como una imagen encontrándose la misma alojada en dicho repositorio público podría ejecutarse desde este punto. Conviene destacar que para que esto fuera posible, hubo que realizar unas ligeras modificaciones en el código del servicio de la solución, debido a unos ajustes para interpretar parámetros de las variables de entorno requeridas por el módulo OT.

El funcionamiento de esta interacción fue el siguiente (véase Figura 5.49: Integración de la solución sobre plataforma PIXEL escenario PEI):

1. La imagen se encuentra en DockerHub y se publica a través de las OT. Este software posee un módulo adaptador que interpreta la estructura del servicio y obtiene información sobre el tipo de ejecución, información del modelo, etc. Esta información se encuentra en un archivo llamado *getInfo*.
2. Una vez publicado, el PEI puede ser ejecutado pasando como argumento de entrada un fichero JSON llamado *Instance* con parámetros como de dónde recuperar los datos, los valores de normalización o dónde almacenar los resultados. Este punto fue clave para validar la solución de esta tesis doctoral, ya que se asemeja enormemente al diseño planteado para la ejecución de servicios en la arquitectura propuesta (véase la sección 3.4.2).
3. No obstante, el módulo OT requiere un paso adicional para la efectiva ejecución del servicio. A través de un formulario web se debió introducir información para que este módulo pudiera planificar la ejecución periódica de dicho modelo. Para este escenario de validación

(PEI en el puerto de Tesalónica), coincidiendo con las conclusiones de la fase I, fue seleccionada una periodicidad de un mes.

4. Por último, este módulo también requería la indicación del índice donde almacenar los resultados de la ejecución del modelo.



**Figura 5.49:** Integración de la solución sobre plataforma PIXEL escenario PEI

En tercer lugar, se requirió la integración de la visualización desarrollada. Debido a que la tecnología de desarrollo usada fue la misma (Vue.js) y a que la estructura del *Dashboard* de PIXEL lo permitía (con una estructura ordenada y escalable), pudo integrarse de una manera rápida y fácil. Hubieron de hacerse una serie de ajustes (URLs de llamada para peticiones a base de datos, etc.) que se solventaron con rapidez dentro del proyecto. Los resultados del uso de la interfaz se muestran en la siguiente subsección.

### **Software y equipamiento utilizado:**

Por último, se concluye esta subsección con la indicación de los detalles técnicos del despliegue. En este caso, todos los módulos se desplegaron en la misma máquina, excepto las datos ofrecidos por proveedores externos. En particular, se denomina a esta instalación “*cloud*” ya que se encontraba instanciada en un servidor central dedicado (aunque se trata de un servidor *on-premise*). El equipamiento fue provisto por la Autoridad Portuaria de Tesalónica. Una vez más, cabe destacar que en este escenario de validación, únicamente se instaló parcialmente la arquitectura propuesta (existía una instalación de base de la plataforma IoT de PIXEL sobre la que se integraron módulos específicos).

Por tanto, los datos que se ofrecen a continuación corresponden a todo el entorno PIXEL. Esto se realiza para contextualizar de una manera realista sobre los entornos en los que se han validado los módulos de la arquitectura. Se señala aquellas partes específicas que corresponden a esta tesis.

**Tabla 5.13:** Equipamiento *hardware* virtualizado utilizado en escenario PEI

| Elemento                            | Especificación técnica          |
|-------------------------------------|---------------------------------|
| Máquina virtual 1 ( <i>core</i> )   | 4 cores x 16GB RAM x 300GB HDD  |
| Máquina virtual 2 ( <i>public</i> ) | 4 cores x 16GB RAM x 200 GB HDD |
| Sensor de ruido y luz               | SmartSpot Station HOPU          |

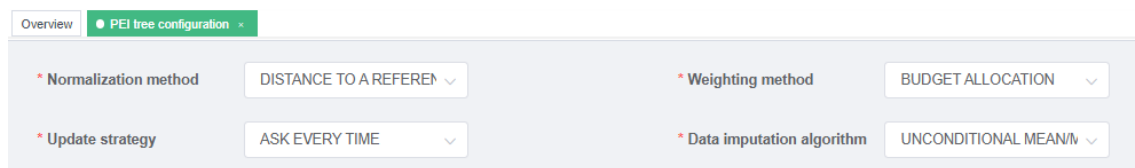
Además, tanto las especificaciones de software de los módulos preconstruidos (componentes FIWARE y servidor Apache) como las de las tecnologías utilizadas para desarrollar y ejecutar los demás módulos se detallan en la siguiente tabla:

**Tabla 5.14:** Tecnología *software* utilizada en el escenario de validación TCI

| Elemento                          | Tecnología / versión      |
|-----------------------------------|---------------------------|
| Sistema operativo del servidor    | Ubuntu Server 18.04.4 LTS |
| Java (usado para la solución)     | OpenJDK 1.8.0_252         |
| Vue.js (interfaz módulo solución) | 3.0.0-beta23              |
| Python (usado para la solución)   | 3.9.1                     |
| Docker + Docker Compose           | 19.03.10 + 1.25.3         |
| PIXEL Operational Tools           | 0.5                       |
| PIXEL Dashboard                   | 3.3.0                     |
| FIWARE Orion                      | 2.4.0                     |
| PIXEL Information Hub             | 1.6                       |
| PIXEL DAL Proxy                   | 1.1.1                     |

### *Fase III – Configuración y resultados*

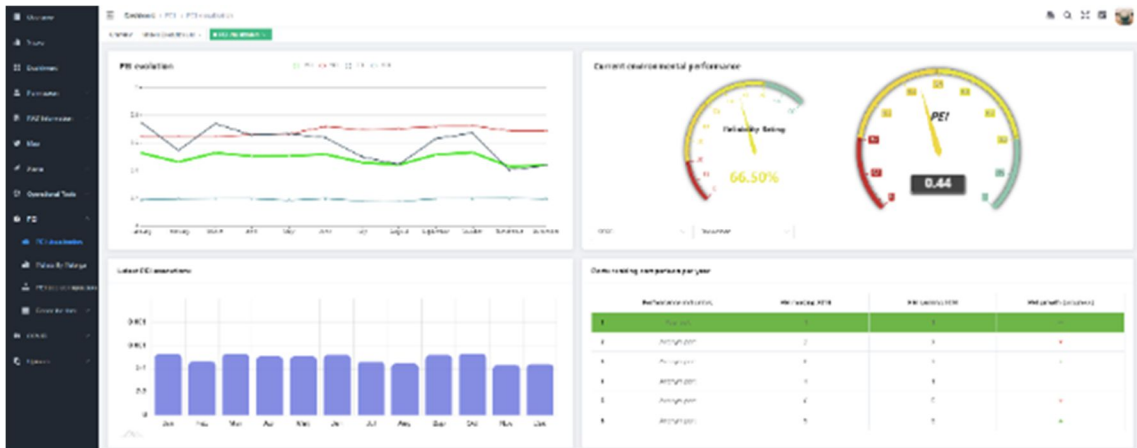
Por último, se procedió a ejecutar la fase III de la metodología, orientada a la configuración de parámetros y a la observación de los resultados. La selección de métodos matemáticos según fase I se corresponde con la siguiente imagen:



**Figura 5.50:** Selección de métodos matemáticos para cálculo del CI PEI







**Figura 5.52:** Ejecución del servicio CI en el escenario PEI (muestra 2020)

En la captura mostrada, el resultado para el mes de diciembre fue de 0.44 (normalizado), uno de los valores más bajos de la anualidad, lo que permitió al puerto analizar la situación y tomar medidas apropiadas.

Dashboard / PEI / Reliability Ratings

Overview | PEI tree configuration | **Reliability Ratings**

| Data origin    | Subindex     | Piece of data                         | Optimal retrieval way           | Current retrieval way         | Reliability Rating | Aggregated RR (subindex) | Aggregated RR (origin) |         |
|----------------|--------------|---------------------------------------|---------------------------------|-------------------------------|--------------------|--------------------------|------------------------|---------|
| Ships          | Air Emission | IMO number and gross tonnage of ships | Real time API                   | Periodic API                  | 80.56 %            | 67.96 %                  | 83.47 %                |         |
|                |              | Main and auxiliary engine of ships    | Periodic API                    | Average value from literature | 36.84 %            |                          |                        |         |
|                |              | Berth and maneuvering time of ships   | Sensors                         | Periodic API                  | 86.49 %            |                          |                        |         |
|                | Waste        | MARPOL annexes about waste            | Pixel proxy tool                | Periodic API                  | 97.14 %            |                          |                        | 97.14 % |
|                |              | Wastewater                            | MARPOL annexes about wastewater | Pixel proxy tool              | Periodic API       |                          |                        | 85.29 % |
| Terminal       | Wastewater   | Wastewater produced by the terminals  | Pixel proxy tool                | Handtyped forms               | 21.95 %            | 21.95 %                  | 21.95 %                |         |
| Port authority | Wastewater   | Wastewater produced by the PA         | Pixel proxy tool                | Handtyped forms               | 9.76 %             | 9.76 %                   | 9.76 %                 |         |

**Figura 5.53:** Ejecución del servicio CI índice de fiabilidad en el escenario PEI

En cuanto a la validación del CI de índice de fiabilidad, la ejecución fue satisfactoria, y como muestra se incluye la información para el puerto de Tesalónica en este escenario de validación. Un valor del 65% se razonablemente bueno (especialmente en comparación con otros puertos), pero existe margen de mejora.

En la siguiente página se incluye (en 3 partes) un ejemplo de obtención de reporte (en PDF) para el puerto de Tesalónica. Como puede observarse, el valor resultante del índice, todos los subíndices y los indicadores (incluyendo valor recibido y normalizado) son incluidos, así como una serie de recomendaciones que

el servicio propone basado en los resultados y en la configuración del motor de reglas. Se han ocultado los valores sensibles debido a la política de protección de datos, pero la información mostrada es suficiente para corroborar la validación de la solución.

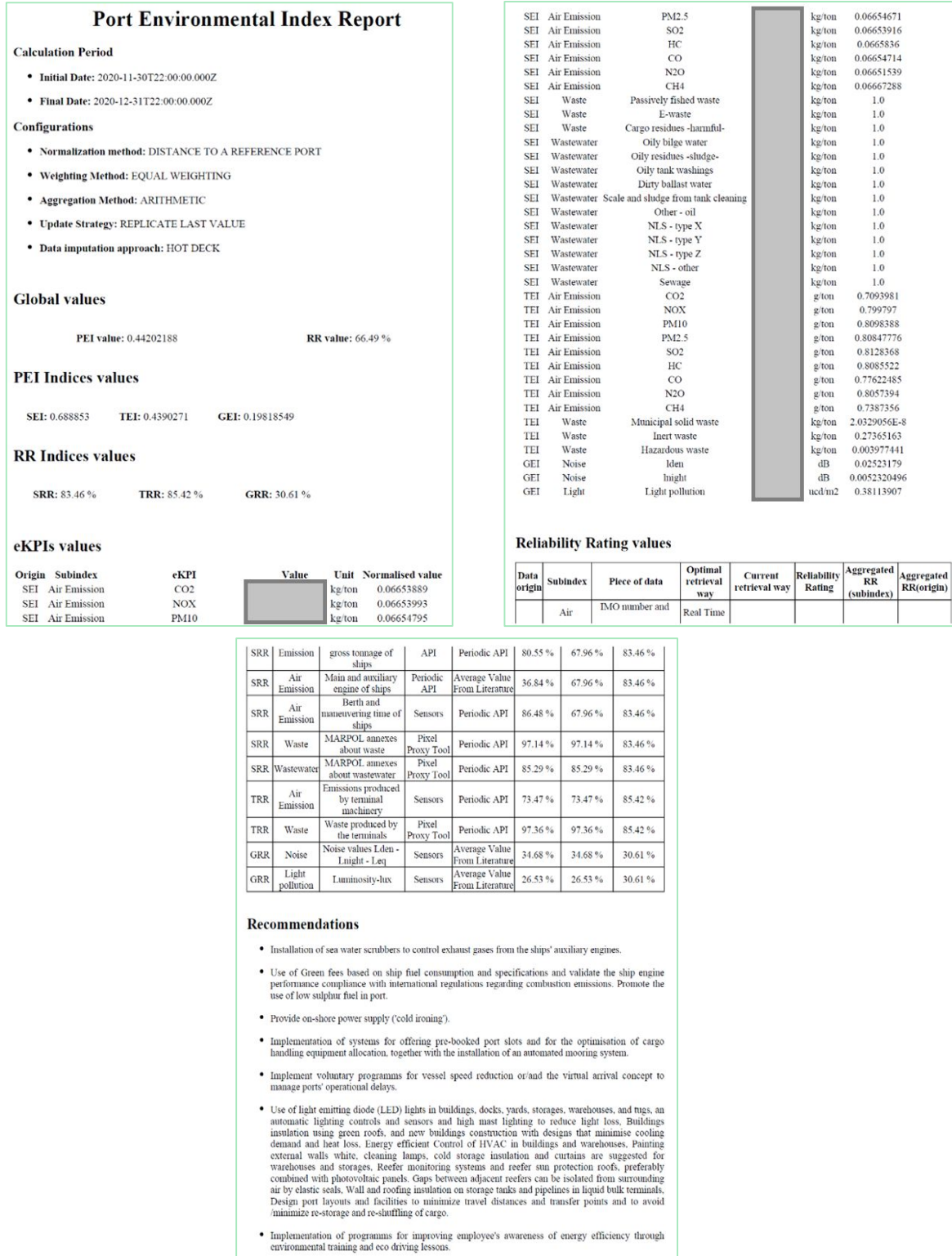


Figura 5.54: Resultado de generación de reporte con recomendaciones del PEI



Para finalizar la explicación de este escenario de validación, se muestra a continuación una imagen de un operario portuario consultando la información obtenida de la ejecución del servicio PEI integrado en la plataforma PIXEL.



**Figura 5.55:** Utilización del servicio CI en escenario PEI en puerto

El objetivo final de aplicación es ayudar a la reducción del impacto medioambiental creando un círculo virtuoso de mejora: adquirir información sobre la situación actual; facilitar la búsqueda de vías de mejora y permitir la estimación del impacto; comprobar el impacto después de la aplicación. En este sentido, en este caso se puede hacer una serie de reflexiones sobre la utilización de este servicio en el puerto de Tesalónica (más información se puede encontrar en el entregable D8.4 del proyecto PIXEL – acceso restringido).

Se ha evidenciado, de 2019 a 2021 (fecha de fin de este escenario), una reducción de alrededor del 5% del coste asociado al impacto ambiental y sonoro en el puerto, gracias a las acciones empleadas parcialmente motivadas por la información provista por parte del servicio.

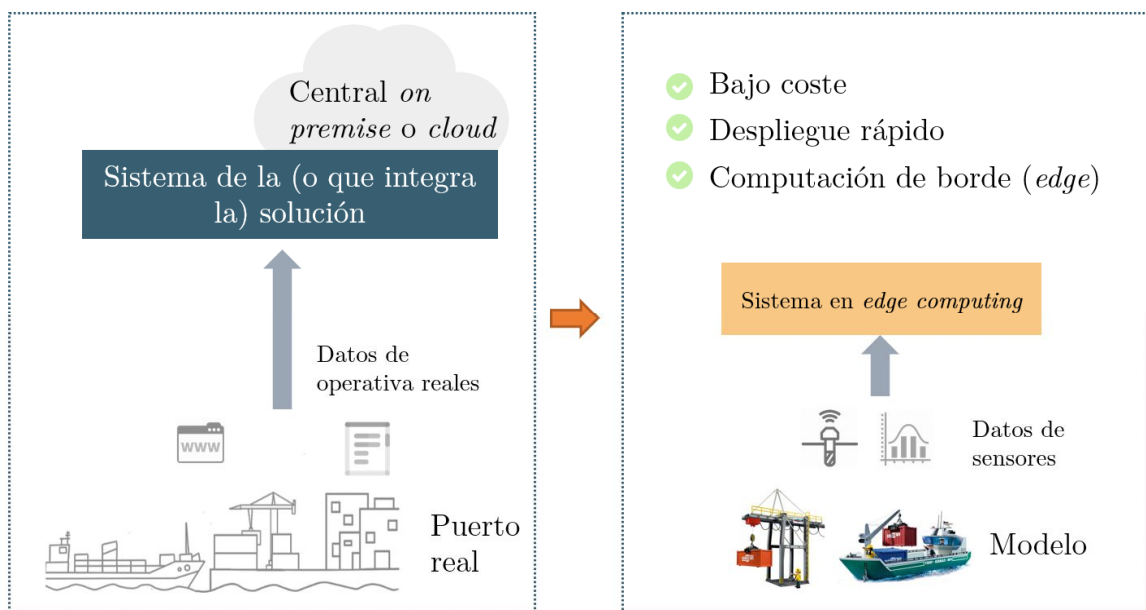
Además, a través de una serie de encuestas a operarios de la Autoridad Portuaria y otros agentes interesados (entrevistas gestionadas por el proyecto a 66 personas), se concluyó que la plataforma en general, y el servicio de CI en particular, mejoró la integración de la planificación puerto-ciudad a través de acciones coordinadas para abordar los impactos negativos del puerto y un 68% de los encuestados coincidieron en que el hecho de adoptar este CI (PEI) contribuyó a mejorar “significativamente” la aceptación del puerto como actor ambientalmente responsable al compartir los resultados del PEI. El 16% de ellos consideró, además, que dicha mejora era más que significativa (“grande”).

## 5.5. Instanciación *edge* para la evaluación rápida de impacto medioambiental a través de un *kit* IoT

### 5.5.1. Contexto

Hasta este punto, los escenarios de validación en los que se demostró la arquitectura planteada pueden ser considerados “de gran alcance”. Incluidos (o no) dentro de un proyecto mayor (PEI en PIXEL), en general han requerido de concienzudos y largos procesos de análisis, despliegue y demostración. En ambas ocasiones, además, han requerido de cierto equipamiento computacional para poder ser utilizados (denominado “cloud”). En cambio, con este escenario de validación se plantea el reto del prototipado, desarrollo, despliegue y demostración rápida de la arquitectura haciendo uso de elementos de computación más ligeros (equiparables a equipamiento habitual en borde – “*edge computing*”). La sección 2.3 expone un análisis profundo de las bondades de la computación en borde en entornos de Internet de las Cosas, así como las herramientas, arquitecturas y tecnologías habituales para ponerlo en marcha. El objetivo de este escenario es trasladar esos conocimientos a la práctica y comprobar, si es posible, que la arquitectura y metodología de despliegue diseñadas son lo suficientemente escalables para soportar este nuevo planteamiento.

El contexto operacional de este escenario de validación consiste en el despliegue de sensores ambientales en un escenario controlado e integrando la solución de la tesis doctoral sobre un escenario simulado haciendo uso de un modelo “a escala” de una interfaz mar-tierra de un puerto marítimo.



**Figura 5.56:** Contexto del planteamiento del escenario de validación  $\mu$ PEI

En cuanto al servicio de indicador compuesto, se planteó hacer uso del conocimiento adquirido durante el escenario de validación anterior. De una manera simplificada, se diseñó un sistema que pudiera obtener datos relevantes para el cálculo de un índice de impacto medioambiental de la llegada, atraque, operación de carga y partida de un barco a un puerto marítimo. Se aporta más información en el estudio de la fase I.

La idea fue también recrear este escenario en un entorno interactivo de cara a su exposición en una feria o evento. Esto se planteó como una posible actividad de diseminación de la investigación por parte del candidato canalizada a través del proyecto PIXEL. Sin embargo, por las restricciones experimentadas a causa de la pandemia por COVID-19, esta actividad no pudo llevarse a cabo. Sin embargo, puede consultarse la demostración de este experimento en un vídeo de YouTube alojado en el canal del proyecto [508].

El diseño funcional de esta demostración consistió en la observación en tiempo real de los resultados de un servicio de CI ( $\mu$ PEI) sobre los datos ambientales recogidos con sensores con una mayor frecuencia de actualización que en los dos casos anteriores. Los objetivos avanzados son: (i) aplicar modelos sobre los datos recogidos aprovechando la arquitectura de la tesis doctoral, (ii) mostrar cómo desplegar fácilmente (de forma *plug-and-play*) un sistema IoT sobre los sensores recogidos. Replicar escenas habituales en puertos marítimos con un modelo a escala (barco, grúa, contenedores) para comprobar cómo cambian los valores ambientales en un escenario portuario “simulado” a lo largo del tiempo y en función de la posición/flujo de los diferentes elementos.

En términos de relevancia para la tesis doctoral, esta aplicación:

- Es la más cercana al tiempo real y a la concepción clásica de IoT de todas en las que se ha validado arquitectura, por lo que queda más patente la utilización de ciertos mecanismos del Internet de las Cosas (por ejemplo, publicación/suscripción a un *broker* MQTT o utilización de dispositivos proveedores de conectividad – WiFi en este caso) y en los dispositivos de recogida de datos en un entorno de prototipo.
- Se observa de una manera clara y directa cómo (debido a la ejecución del servicio diseñado en la sección 4) cambios en los elementos físicos representan un cambio en el índice de modelización de impacto ambiental. El valor añadido queda patente en un simple vistazo.
- Se trata de un despliegue muy customizado y con gran influencia de los conceptos de *edge computing* aprendidos por el doctorando (véase fase II).
- Esbozo de un concepto de explotación de la solución (véase sección 6) denominado “*CIC kit*” que podría ofrecer claras perspectivas innovadoras, facilitando un despliegue rápido de una solución IoT para la medición y predicción de indicadores compuestos.

- Por último, uno de los puntos más relevantes podría ser el uso de equipos de bajo coste y bajo consumo de recursos de computación sobre una simulación de escenario ciudad-puerto. Una de las principales preocupaciones en el panorama de despliegue de soluciones IoT es el coste inicial para poner en marcha pruebas de concepto innovadoras (como las expuestas en las secciones 5.3 y 5.4). Con el sistema propuesto (“*CIC-kit*”), se podrá realizar una evaluación real y valiosa utilizando equipos ampliamente disponibles, baratos y razonablemente precisos, válidos para pruebas a gran escala.

### 5.5.2. Implementación de la solución

A continuación se exponen los hallazgos, decisiones y resultados obtenidos durante el escenario de validación asociado al índice  $\mu$ PEI.

#### *Fase I – Conceptual y servicio*

En cuanto al servicio de indicador compuesto, se planteó hacer uso del conocimiento adquirido durante el escenario de validación anterior.

Durante el proyecto PIXEL se averiguó una manera consolidada de representar el impacto ambiental de un puerto marítimo a través de una serie de indicadores, subíndices e índice final (PEI), proviniendo de una vasta recopilación de referencias en la literatura. Es conveniente destacar que la búsqueda exhaustiva de referencias en la literatura para la creación (tanto del índice anterior como en este caso) no es una actividad objetivo de esta tesis y no es una tarea que fuera llevada a cabo principalmente por el candidato. Muchos de los datos requeridos en aquel caso conllevan una fuerte carga administrativa y únicamente pueden ser provistos por entidades portuarias (por ejemplo, facturas energéticas, informes de residuos, etc.), haciéndolos inviables para su incorporación en un despliegue rápido. Además, la existencia de tal cantidad de indicadores necesarios (43), con la variabilidad y heterogeneidad en la recogida de estos, llevó a dichos escenarios a requerir una periodicidad de ejecución muy baja (cada semana, cada mes...), lejos de los objetivos de demostración cercana al tiempo real que se buscaba con este índice ( $\mu$ PEI).

Por tanto, se procedió a aprovechar el conocimiento sobre aquellos indicadores que sí pudieran considerarse válidos para este planteamiento. Como se ha visto, los indicadores de ruido e iluminación fueron aquellos que pudieron completarse desde información directa de sensores. Su alta disponibilidad en el mercado y su bajo coste hicieron de estos indicadores candidatos perfectos para ser incluidos en este índice rápido. Por otro lado, se decidió también incorporar sensores de calidad del aire para cumplimentar los subíndices de contaminación atmosférica. En este caso, esto sí pudo realizarse de esta manera debido a que el sistema se desplegó en unas condiciones completamente controladas (sala aislada en el laboratorio del grupo de

SATRD en el edificio 4D de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la UPV).

Así, se definió que el índice a calcular en este escenario fuera una simplificación del PEI,  $\mu$ PEI. Observando los potenciales datos a obtener como fuentes de sensores, el diseño de la estructura del índice fue sencillo. Los datos de iluminación y sonido eran asociados, según acuerdo, a un subíndice global (GEI) y la generación de gases por parte de los barcos a la inclusión de un indicador de contaminación atmosférica por parte de la terminal (TEI) asociado a las emisiones producidas por la llegada del barco y la operación de “descarga del contenedor”. Esto se realizó de esta manera ya que el alcance del demostrador se centraba en los aspectos IoT, dejando la puerta abierta a una interpretación propia de la asignación de *cKPIs*.

En vista de lo anterior, el doctorando (junto con compañeros del grupo de investigación) procedió a una búsqueda de los sensores más apropiados para el despliegue del prototipo en este escenario.

De manera análoga al resto de escenarios, en la siguiente figura se muestran los sensores que fueron seleccionados. Las características de unos y otros pueden encontrarse en sus respectivos *datasheets* (evidencias del montaje de la estación se encuentran en la fase II).



**Figura 5.57:** Sensores seleccionados para el escenario de validación  $\mu$ PEI. De izquierda a derecha: (a) sensor de CO2 MG-811, (b) sensor de ruido KY-038, (c) sensor de iluminación KY-018

Se contó también desde la primera fase con la necesidad de incorporar elementos físicos para simular el escenario de llegada, atraque, operación de descarga de contenedor y partida de un barco. Para ello, se han usado los artefactos de la siguiente figura:



**Figura 5.58:** Artefactos físicos para la simulación de demostración escenario  $\mu$ PEI

Por último, para completar esta fase, se realizó una reflexión sobre cuáles debían ser los métodos matemáticos a escoger de entre los disponibles en cada paso (véase sección 4.1.2).

**Tabla 5.15:** Elección de métodos matemáticos para el escenario  $\mu$ PEI

|   |                                |              |
|---|--------------------------------|--------------|
| <b>Gestión de datos ausentes:</b>   | <i>Hot deck imputation</i>     | Último valor |
| Se realizaron pruebas sobre el montaje de la estación de sensores (ver fase II) y se constató que siempre se recibían datos. De este modo, al estar en un entorno muy controlado, no se esperaban valores ausentes. No obstante se configuró para tomar el último valor disponible.   |                                |              |
| <b>Normalización:</b>   | Re-escalado ( <i>Min-Max</i> ) |              |
| En este caso, al tratarse de una simulación de un puerto, la comparación con datos de referencia no es una opción (no existe un puerto “ideal” similar con el que compararse). Analizando los métodos expuestos en la sección 4.1.2.2, se decidió proceder con el re-escalado, cogiendo el mínimo y el máximo de los valores de la serie temporal de los diferentes indicadores y aplicando la fórmula adecuada. Aquí, la estación de sensores fue expuesta a varias condiciones antes del experimento para adquirir un rango de valores suficiente para dotar de utilidad a este método. |                                |              |
| <b>Ponderación:</b>   | Igualitaria                    |              |
| Al no disponer de todos los indicadores analizados en el escenario PEI, se decidió proceder con un reparto de pesos igualitario tanto a nivel indicador (nodos raíz) como subíndices de segundo nivel.  |                                |              |
| <b>Agregación:</b>  | Aditiva                        |              |
| Por último, se podría haber utilizado un método no compensatorio (como una matriz de <i>outranking</i> ) si el indicador tuviera como objetivo comparar las entidades que ejecutan el indicador entre sí; sin embargo, éste no era el caso (ya que se trata de un escenario de validación rápida para solo un puerto simulado) y esta elección habría hecho que el sistema fuera demasiado complejo desde el punto de vista computacional para el alcance del experimento.  |                                |              |

### Fase II – Arquitectura y conectores

El caso de aplicación “*CIC kit*” o  $\mu$ PEI resulta especial ya que se ideó desde cero partiendo de las bases de indicador anterior con el objetivo de demostrar varias innovaciones de la arquitectura propuesta (escalabilidad, *edge computing*).

Por tanto, el diseño para el despliegue modular de la arquitectura pudo establecerse de la manera más oportuna para favorecer estas premisas. Los componentes esenciales de la arquitectura fueron incluidos, sin la necesidad de añadir otros adicionales (como la normalización externa o la predicción). Al no precisar de una visualización particular específica, tampoco fue necesaria la



adaptación de este componente. Por tanto, se debió incluir lo siguiente: adquisición de datos, persistencia (con conversor incluido), ejecución del servicio, seguridad y visualización.

El aspecto más importante a tener en cuenta durante el diseño de la arquitectura de despliegue fue la necesidad de ejecutar la mayor parte de módulos cuanto más cerca posible de la fuente de datos, para asemejar a un modelo de *edge computing* en un entorno real. Por tanto, consideraciones como la utilización de bases de datos más ligeras fueron tenidas en cuenta.

De esta manera, el despliegue de la arquitectura fue el siguiente:

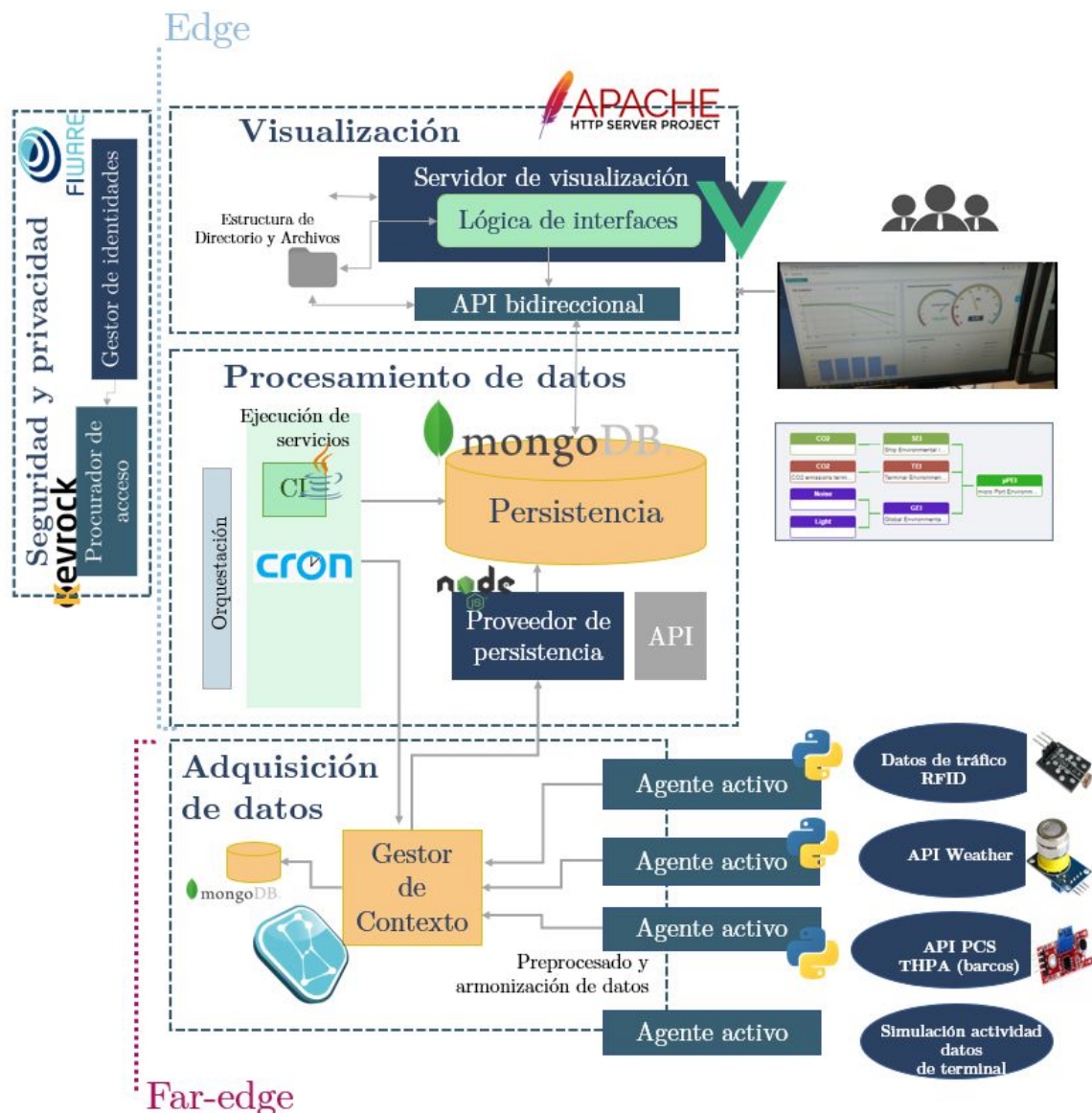


Figura 5.59: Diseño de arquitectura de despliegue para el escenario  $\mu$ PEI

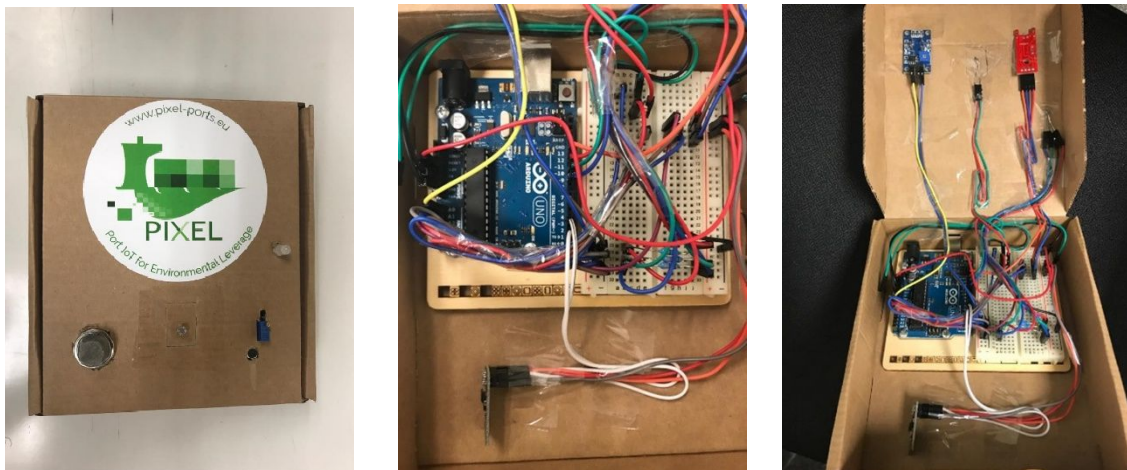
Previo al análisis de la implementación de los módulos, conviene evidenciar el montaje realizado para el escenario de validación. No puede comprobarse la instalación de agentes, del gestor de contexto o el funcionamiento de la visualización si no se dispone de datos para realizar la misma (al menos, datos que no sean

únicamente sintéticos). Por tanto, los sensores identificados en la fase I fueron adquiridos y preparados para alimentar al sistema del experimento.

Cada uno de los sensores poseía unas características diferentes, en especial de calibración y ajuste para proporcionar los valores apropiados (varios niveles de complejidad). En el caso del sensor de iluminación, una serie de cálculos fueron también necesarios ya que las condiciones en especificadas en la ficha técnica no se correspondían con las del entorno de prueba. Usando una placa de prototipado rápido (*proto-board*), se instalaron los sensores y los datos (de ambos tres sensores al mismo tiempo) eran capturados por diferentes pines analógicos de un Arduino UNO [509] que fue el primer elemento de computación incluido para recoger estos datos. En esta placa de desarrollo, se cargaron varios ficheros software compilados para la recogida de los mismos y su envío al punto de comienzo de la arquitectura de la solución.

En este momento, para dotar de conectividad a la placa (por ende, a los sensores), se tuvo que contar con la incorporación de un componente hardware ESP-01 [510], basado en sistema *en chip* ESP-8266 para conectar dicho Arduino a una red WiFi que dio cobertura a todos los elementos del experimento.

Una vez lograda dicha conexión, se estableció que los tres sensores, junto con el procesador de recogida de datos y el elemento de conexión, debían ser integrados en un mismo elemento físico (de forma rápida), lo que puede observarse en la siguiente figura. Esta decisión se alineó con la idea de avanzar en la validación de la arquitectura de la tesis doctoral como “preparada para escenarios *edge computing*”, ya que este dispositivo debía tener una naturaleza ubicua. También, por ello, se incluyó una batería para alimentar los diferentes elementos electrónicos.



**Figura 5.60:** Montaje de estación de sensores para escenario de validación  $\mu$ PEI

Habiendo asegurado la conexión (a nivel 3, de red [511]), se procedió a diseñar la conexión de los datos (capa de aplicación, ya que el transporte es garantizado TCP/IP gracias al ESP-01). Se decidió utilizar MQTT ya que se trata del protocolo más utilizado en IoT para la conexión de datos, por tanto, robusto y con gran



soporte. Desde la lógica dentro del procesador de Arduino, se cargaron ficheros software suficientes para garantizar el envío periódico (cada segundo) de información de los tres sensores mediante su publicación en tres *topics* MQTT diferentes. Es conveniente, también, destacar que este envío (publicación) siguió un formato específico de encriptación de la información de tal manera que en caso de captura del tráfico WiFi estos datos no fueran interpretables (previniendo problemas relacionados con la seguridad).

Para que esta transmisión pudiera realizarse, otro elemento dentro de la misma red debía actuar como concentrador (*broker*) MQTT, exponiendo la información de dichos *topics* a potenciales suscriptores. Este *broker* se implementó con la tecnología Mosquitto y el sistema que lo alojó fue la Raspberry Pi 4 Model B que constituyó el elemento principal *edge* del experimento. En esta Raspberry Pi se encontraba también el concentrador de datos de la solución de esta tesis doctoral, evitando por tanto la necesidad del envío de datos a otros entornos. En este sentido, los datos no podían ser movidos directamente desde un *broker* (MQTT) al posterior (gestor de contexto), sino que requerían conectores de datos.

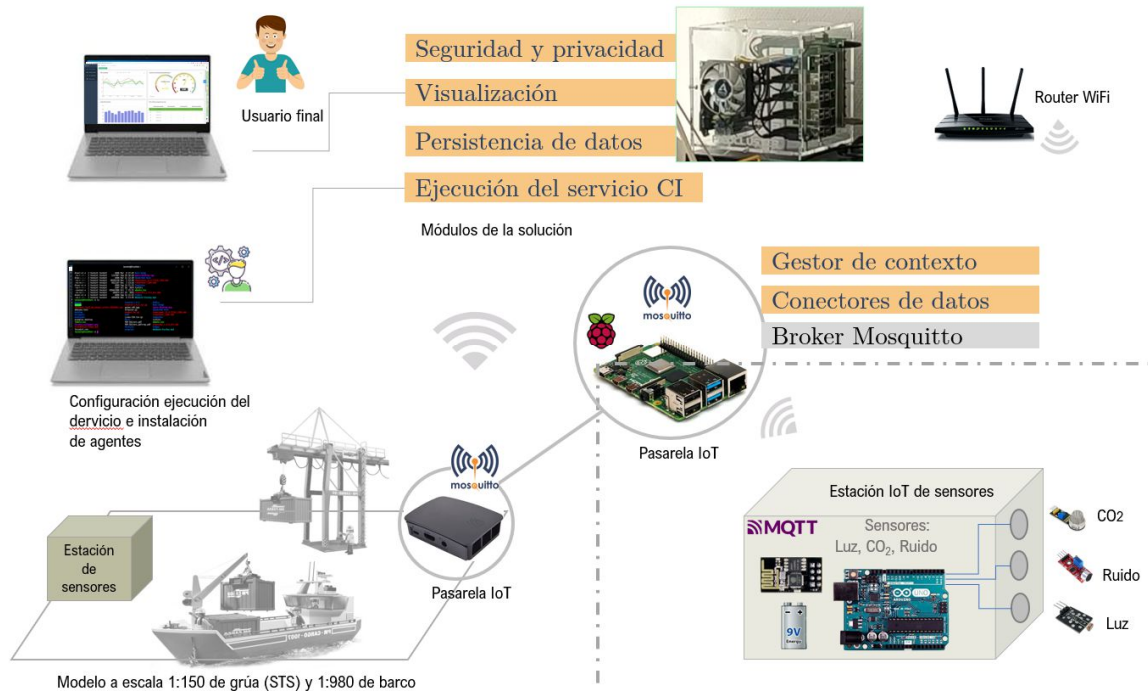
De manera equivalente a lo que se ha reportado en el resto de escenarios, conectores de datos (agentes) fueron necesarios para convertir la información primitiva en indicadores (*cKPIs*) preparados para alimentar el servicio de índices compuestos. La lógica en cada uno de ellos (un total de 4) consistió en la conversión al formato esperado y la aplicación de las mismas fórmulas definidas para el caso de validación PEI, cuando fue necesario. Además, se implementó una simulación de contaminación atmosférica asociada a la maquinaria de la terminal (una grúa a escala). Ya que no se trata de un elemento real, se utilizaron datos en un rango equivalente para ajustar los resultados del índice.

En ese momento, todos los datos adquiridos (recibidos como actualizaciones de entidades), se procesan en las instalaciones (en la pasarela), adaptándose a un modelo de datos común y con la periodicidad acordada para el cálculo del CI.

Retomando el flujo anterior, se ha comentado que el gestor de contexto propuesto por esta tesis doctoral se encontraba igualmente en ejecución en el entorno *far-edge* del despliegue, conectado vía WiFi con la estación de sensores y preparado para proseguir con la arquitectura hacia el módulo de persistencia.

Finalmente, tanto dicho módulo (persistencia) como el resto de componentes esenciales de la arquitectura se han instalado en un entorno de computación edge. Estos son: (i) ejecución del servicio de CI, (ii) persistencia de datos, (iii) visualización y (iv) seguridad y privacidad. Se decidió contar con este elemento adicional (siempre dentro de la computación de borde) debido a que la instalación de todos los elementos de la arquitectura en un dispositivo como la pasarela escogida hubiera supuesto una sobrecarga excesiva, dificultando la escalabilidad del sistema (especialmente a medida que el número de registros en la serie temporal de cada indicador hubiera ido incrementándose).

En la siguiente figura puede observarse un esquema del montaje final escogido y desplegado en este escenario:



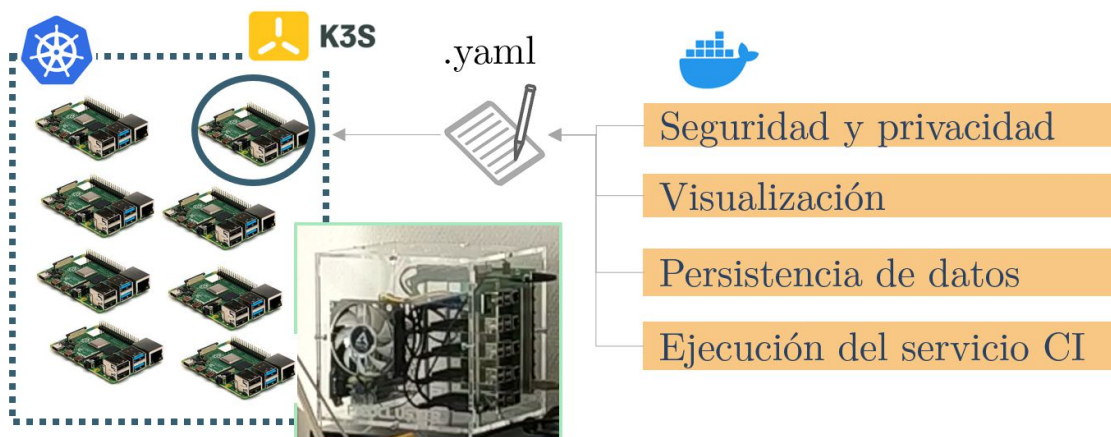
**Figura 5.61:** Montaje y componentes del escenario  $\mu$ PEI

En la siguiente lista se lista los elementos que lo conforman, considerando el flujo expuesto en la página anterior:

- Una maqueta “portuaria”, compuesta por una grúa *Ship-to-Shore* (STS) a 1:150 y un buque portacontenedores a 1:980. Ambos elementos se han montado sobre un espacio asimilable tierra firme que simula un patio de terminal portuaria.
- Una estación IoT que equipa tres sensores IoT (CO<sub>2</sub>, ruido y luz) y un Arduino para capturar datos y un módulo ESP-01 para conectarse vía Wi-Fi a la pasarela. Las placas realizan la recogida de datos en tiempo real de los sensores que se conectan a ellas, y luego publican estos datos en una pasarela IoT.
- Un elemento de computación *far-edge* SBC (del inglés *Single Board Computing*), materializado con una Raspberry Pi Model 4 B. Este dispositivo contiene esencialmente (i) una pasarela IoT implementada, primero con un servidor Mosquitto que permite la recepción de los datos de la estación de sensores bajo un modo publicación-suscripción, (ii) un gestor de contexto, moviendo un elemento relevante de la computación a un entorno cerca de la fuente de datos, implementado siguiendo la arquitectura con el módulo ORION de FIWARE y (iii) elementos auxiliares (agentes) para conectar los datos entre ambos *brokers*.

- Un elemento de computación *edge*, con mayor capacidad de procesamiento que el anterior pero todavía considerado de borde al tener características de tamaño, rendimiento, configuración, flexibilidad y conectividad como las resaltadas en la revisión del estado del arte de estas tecnologías (véase sección 2.3).

El objetivo de este elemento es cubrir el resto de módulos de la solución, tales como almacenamiento, seguridad o ejecución del índice  $\mu$ PEI. Este elemento *edge* se trata de un clúster de Raspberry Pi 4 Model B (10 unidades) ensambladas sobre un elemento de soporte, conexión y ventilación. No obstante, para que todos estos módulos se ejecutaran de una manera alineada con el IoT de Nueva Generación (NGIoT), y para flexibilizar, mejorar y escalar la arquitectura, tuvo que analizarse cuál sería la mejor forma de desplegar la arquitectura. Como se ha dicho está compuesto por 10 dispositivos de procesamiento, donde todos ellos disponían de un sistema operativo Linux (Raspbian OS). Podría haberse optado por la instalación de los diversos módulos (seguridad, bases de datos de persistencia) en dichos elementos del clúster de forma separada, analizando requerimientos de computación, software, consumos, etc. No obstante, este procedimiento no hubiera seguido los patrones de escalabilidad y DevOps planteados por NGIoT. Por tanto, se procedió al despliegue de dichos módulos usando el *framework* de orquestación Kubernetes. En particular, se decidió utilizar K3s [512] ya que se trata de la distribución más recomendada para este tipo de dispositivos (sobre arquitectura ARM [377]). Se eligió una de las Raspberry como *master* del clúster y se unieron todas las demás como *worker nodes*. De este modo, a través de la creación de un manifiesto `.yaml` refiriendo a las imágenes de los módulos de la arquitectura, se consigue abstraer todas las Raspberry como una sola, desplegando así todos los módulos con una simple orden de ejecución (*kubectl*). Esta decisión resulta especialmente interesante ya que, de este modo, se simplifica también enormemente las configuraciones de puertos, seguridad y otras conexiones que son gestionadas directamente por el motor de Kubernetes a partir de dicho manifiesto.



**Figura 5.62:** Despliegue de la arquitectura usando Kubernetes escenario  $\mu$ PEI

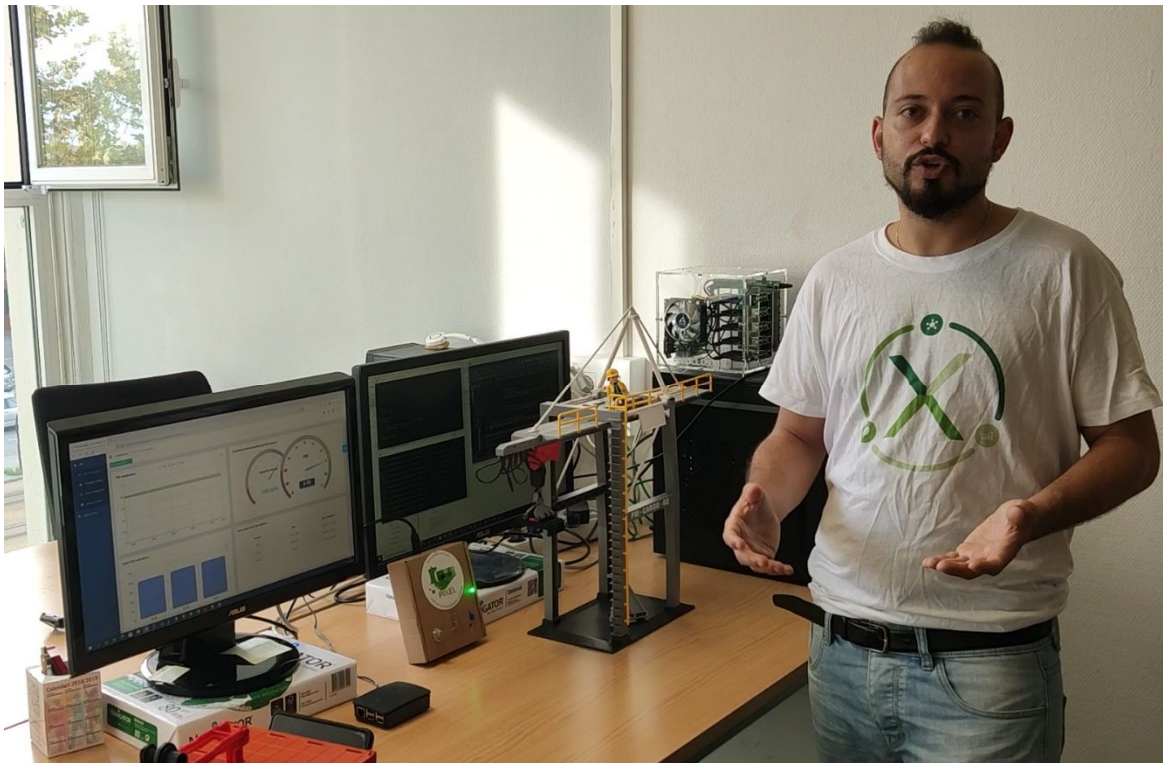
- Elemento de interfaz: un ordenador conectado a través de la URL escogida a la interfaz expuesta por el sistema.

- Un router WiFi para facilitar la comunicación entre todos los componentes. La conexión se establecerá mediante MQTT (estación-*far-edge*) y HTTP sobre TCP/IP (*far-edge-edge*)

Como nota final, todos los componentes utilizados en el escenario de validación comparten los mismos principios fundamentales: dispositivos de bajo coste y fáciles de conectar y tecnología software abierta. Una vez más, los procedimientos anteriores deberían servir para validar la facilidad de despliegue de la solución propuesta en esta tesis doctoral.

### *Fase III – Configuración y resultados*

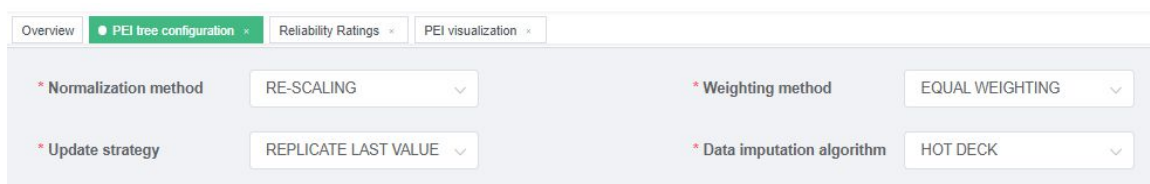
El escenario de validación fue desarrollado por el candidato y demostrado durante el mes de septiembre de 2021. En la siguiente imagen se observa al doctorando realizando una explicación del proceso para el seminario web emitido en directo por el proyecto PIXEL. En la imagen se pueden apreciar los diferentes elementos del montaje, desde los elementos a escala hasta la estación de sensores, la Raspberry Pi *far-edge* y el clúster *edge*.



**Figura 5.63:** Despliegue de la arquitectura usando Kubernetes escenario  $\mu$ PEI

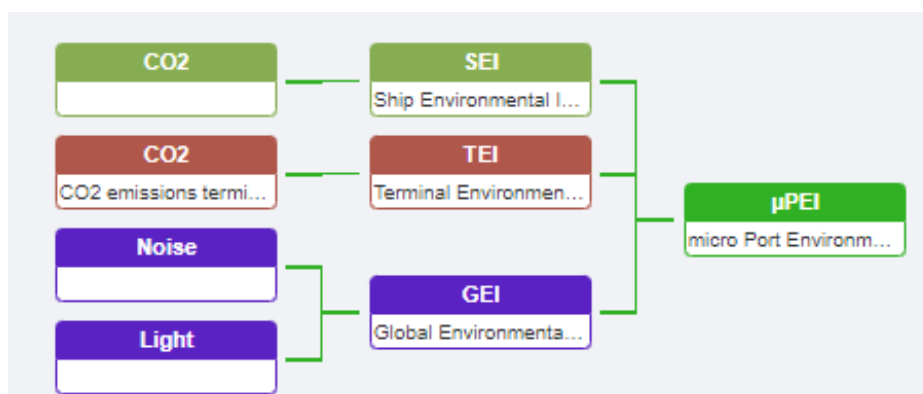
En las siguientes páginas se exponen las particulares configuraciones y los resultados del experimento.

En primer lugar, la selección de métodos matemáticos según fase I se corresponde con la siguiente imagen:



**Figura 5.64:** Selección de métodos matemáticos para cálculo del CI  $\mu$ PEI

En cuanto a la estructura de cálculo establecida a través del árbol en la interfaz, se estableció como sigue (al tratarse de una ponderación igualitaria, no se indican los pesos):



**Figura 5.65:** Estructura del árbol para el cálculo del  $\mu$ PEI

Una vez completada la configuración, tanto el servicio como la arquitectura estuvieron listas para la validación.

Considerando el entorno global de simulación y de demostración de prototipo, se trataron de recrear ciertos eventos para ilustrar posibles operaciones en un puerto que tendrían influencia en el valor de los índices y subíndices realizados.

En este sentido, se realizaron acciones como reproducir un sonido que actuó como bocina del barco, producir una respiración de emisiones de CO<sub>2</sub>, cambiar los valores de luz simulando noche/día o modificar la posición y flujo del contenedor de una grúa. Además, gracias al mecanismo motriz del artefacto de la grúa, se simuló una operativa de descarga, mientras el barco permanecía atracado. En la Figura 5.67: Operaciones simuladas para la recreación del escenario  $\mu$ PEI se propone un resumen.

Durante estos movimientos, se hicieron patentes los cambios en los cálculos del indicador compuesto siguiendo la periodicidad determinada. A pesar de tratarse de un entorno en gran medida simulado, bastó para observar que, en caso de proveer

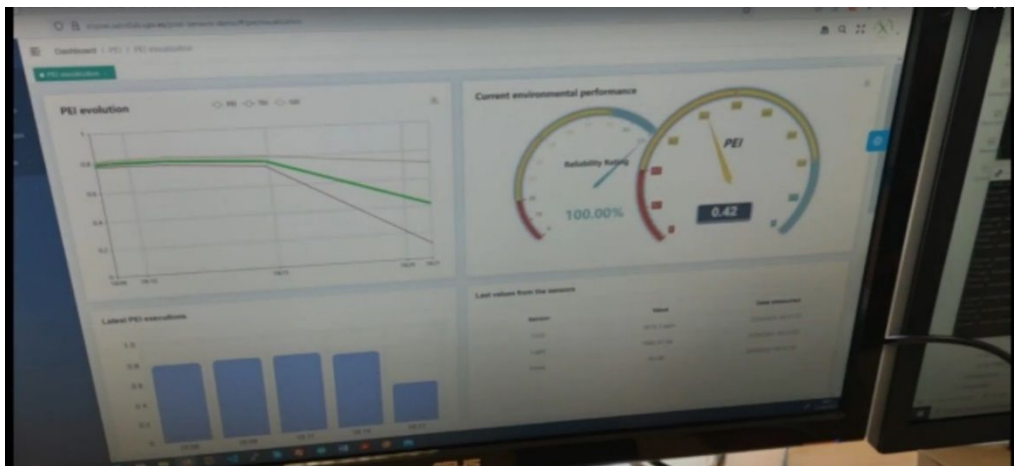


información en tiempo real, la arquitectura está preparada para ofrecer información sobre los indicadores definidos usando los componentes diseñados.



**Figura 5.67:** Operaciones simuladas para la recreación del escenario  $\mu$ PEI

Por último, los resultados observados reflejaban unos índices variantes en función de los eventos simulados (que podían ser captados por los sensores), reflejando una correspondencia con dichos eventos.



**Figura 5.68:** Resultado ejecución índice  $\mu$ PEI

Como conclusión del caso de uso, se ha conseguido validar una serie de requisitos propuestos para la arquitectura a la vez que se ha observado que es posible aprovechar equipos de bajo coste y técnicas de IoT para resolver problemas complejos y costosos, como la monitorización en tiempo real del impacto ambiental de un puerto.

En concreto, las principales innovaciones han sido (a) al uso de dispositivos y equipos de sensores de bajo coste, bajos recursos y agnósticos para producir resultados exitosos de monitorización medioambiental, (b) el bajo nivel de dificultad para desplegar dicho sistema (físico y lógico) y su integración en una solución de la tesis doctoral, manteniendo la seguridad de los datos. También ha resultado clave para fortalecer la perspectiva de la creación de un dispositivo y software (**CIC kit**) para un despliegue rápido del calculador de índices compuestos, lo que podría ser realmente interesante de cara a una futura explotación del producto (sección 6).

## 5.6. Instanciación *cloud* en un entorno de terminal logística para predicción de distancia social en época de COVID-19

Este caso de uso tiene como objetivo la simulación de escenarios en terminales logísticas portuarias en términos de operaciones esperadas para operar un barco de mercancías, tratando de predecir potenciales situaciones donde la distancia social y las medidas de protección asociadas a la COVID-19 no fueran a ser cumplidas. Se trata de un escenario de validación considerablemente secundario en la tesis doctoral. La principal utilidad de cara a la validación de la solución fue la demostración de una rápida capacidad de adaptación y despliegue, así como la facilidad de integración con otros servicios. Este escenario presta menos atención a la construcción del índice compuesto (es un valor sencillo que no requiere comparación entre entidades y solo debe servir como referencia estimativa). De hecho, la inclusión de este caso en la tesis radica en la rapidez de instalación y cobertura de un problema en un corto espacio de tiempo. Este sistema se diseñó (a partir de la arquitectura de la tesis) y desplegó exitosamente en el puerto de Monfalcone (Italia), incluyendo la conexión de datos heterogéneos y la alimentación del servicio, en un plazo menor de 1 mes.

### 5.6.1. Contexto

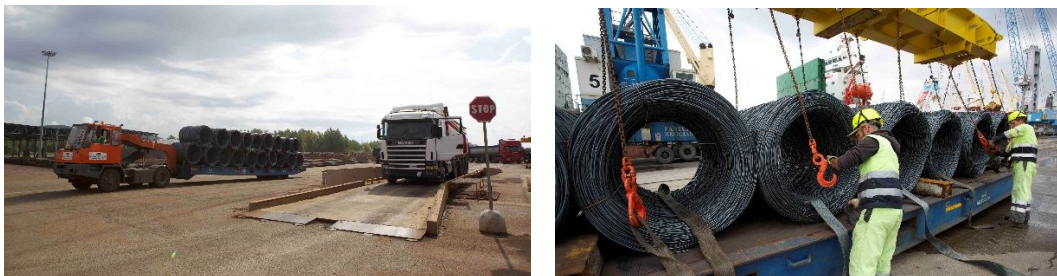
El brote de COVID-19 que golpeó a todos los países del mundo desde la primavera de 2020 obligó a los gobiernos y a las autoridades nacionales e internacionales a tomar medidas sin precedentes, como el cierre y la restricción de la circulación de personas (y trabajadores) para comprobar y controlar la propagación exponencial de la pandemia. El sector del transporte de mercancías y de la logística estuvo en primera línea desde el principio de la crisis del COVID-19, haciendo frente a estas restricciones sin detener su actividad, clave para el comercio mundial.

La mayoría de los países restringieron o detuvieron los vuelos internacionales; mientras que, en el sector del transporte marítimo, los buques fueron puestos en cuarentena durante semanas antes de que se les permitiera entrar en los puertos. Los contenedores de mercancías y los camiones fueron retenidos en los puertos y en las fronteras estatales para ser revisados, afectando especialmente a los escenarios de conectividad intermodal. El aumento de los controles fronterizos y de la normativa aduanera provocó, también, un aumento de los tiempos de espera. Asimismo, un mermado cumplimiento de la primera y última milla creó desafíos extremos para el transporte de mercancías y especialmente para el transporte por carretera. En este sentido, los puertos, con sus terminales logísticas a la cabeza, son nodos esenciales en la cadena de suministro multimodal global, y, como se ha visto, la crisis de COVID-19 repercutió significativamente en ellos. Al margen de la

mencionada ralentización del transporte, se debieron aplicar rigurosas medidas de seguridad sanitaria como el establecimiento de distancias sociales entre las personas y, por tanto, también entre los trabajadores de la terminal (por ejemplo, limitando la densidad en espacios tanto interiores como exteriores). Al margen de las restricciones especiales, existe también un reciente reglamento europeo sobre el Paquete de Movilidad (*Mobility Package*) [513] que expone que las áreas de estacionamiento de camiones deben estar seguras y protegidas, tomando como base de partida unas condiciones de comodidad, espacio y seguridad de los conductores de los camiones en caso de que se formen colas de espera en la entrada de los puertos. Debido a la situación expuesta, muchas zonas de aparcamiento (por ejemplo, en el puerto de este escenario de validación: Monfalcone) estaban abarrotadas, incurriendo en potenciales incumplimientos de esta normativa. Este hecho no sólo resulta extremadamente peligroso en términos de seguridad vial, sino también muy incómodo y preocupante para la salud de los trabajadores.

Todos estos problemas llevaron, dentro del proyecto PIXEL (participado por el doctorando) a plantear una solución rápida donde pudiera aprovecharse en desarrollo de una plataforma IoT para ayudar a evaluar cómo las medidas de distanciamiento social de COVID-19 podrían introducirse en la programación de las operaciones portuarias evitando la aglomeración de trabajadores y conductores de camiones en una zona de la terminal durante un espacio de tiempo. Además, se incluyeron también en este despliegue ágil los impactos en la productividad, medioambiente y eficiencia energética que las medidas de restricciones de distancia social en la terminal podrían ocasionar al puerto. Ayudar a los gestores portuarios a comprender cómo y cuándo (y si) podría existir un riesgo de violar las restricciones de distancia social en función de la mercancía a operar en un horizonte a corto plazo se identificó como un objetivo interesante que podría impulsar a un operador de la terminal a tomar las medidas adecuadas para prevenirlo. Por ejemplo, podría estructurarse la secuencia de operaciones de maquinaria en diferentes áreas de una manera alternativa, o podría derivarse parte la mercancía o camiones a espacios fuera del puerto (como, por ejemplo, a un nodo logístico cercano).

Por último, debido a estas observaciones preliminares, el Puerto de Monfalcone, junto con SDAG, como terminal interior, estaba interesado en evaluar cómo las medidas de distanciamiento social COVID-19 pueden introducirse en la programación de las operaciones portuarias y simular qué impactos en términos de productividad y eficiencia energética tendrían esas medidas.



**Figura 5.69:** Contexto de escenario CEDI - Puerto de Monfalcone



## 5.6.2. Implementación de la solución

A continuación se exponen los hallazgos, decisiones y resultados obtenidos durante el escenario de validación asociado al índice CEDI.

### *Fase I – Conceptual y servicio*

Similarmente a otros casos, se contó con la colaboración estrecha de agentes directamente involucrados en el proceso para definir el índice compuesto. En este caso, el índice compuesto no fue la principal atención del escenario de validación. No obstante, existió la necesidad de condensar información de varios ámbitos (dimensiones) que fueran brindados conjuntamente con el resultado principal esperado. Por ello, el diseño del CI en este caso fue muy relajado y no se estableció la necesidad de seguir ninguna metodología ni formalidad en su definición, se trata de un parámetro interno para la muestra conjunta en una interfaz enfocada a la simulación de operaciones de terminal. En este escenario, la prioridad fue integrar la información, ofrecer una solución rápida y facilitar la toma de decisiones de los operarios de la terminal.

Según la definición del caso de uso, el índice compuesto CEDI representa el estado de densidad de trabajadores de la terminal en un área particular en un momento determinado. Al contrario que en otros casos, el CEDI no está orientado a representar un impacto retroactivo a partir de una serie de entradas, sino a simular un escenario futuro en función de la ejecución de servicios adicionales en base a información de buques, operaciones, maquinaria y plan de la terminal.

El objetivo del escenario (donde encaja el índice CEDI) se sustenta sobre cuatro pilares fundamentales:

- Definición de un nivel de densidad de ocupación por zonas para garantizar el distanciamiento físico de acuerdo con las disposiciones de la OMS y representación de su cumplimiento, en conjunto con otros parámetros, condensado en índice compuesto.
- Monitorización en tiempo real de dichos niveles.
- Creación de alertas en caso de que se vaya a exceder dicho nivel de densidad.
- Estimaciones de ocurrencia en función de la mercancía a operar (buques a procesar) y de la planificación habitual de las operaciones y maquinaria de la terminal.

En este sentido, la creación del CEDI fue concebida para tratar de representar dicha densidad en conjunto con:

- Los parámetros de productividad de la terminal.
- La energía consumida (estimada) para la ejecución de dichas operaciones.

- El impacto medioambiental (a nivel emisiones a la atmósfera de dichas actividades).

Esto resulta especialmente relevante ya que la intención es incluir (de cara al usuario final) la opción de seleccionar dos modos de simulación:

- a) Sin aplicación de las restricciones: Se trataría de una estimación de dicho índice, densidad, etc., ofreciendo información de cuál sería el escenario de operación según los parámetros de funcionamiento de la terminal predefinidos.
- b) Aplicando las restricciones: Estimación de índice y resto de parámetros forzando a que la densidad de trabajadores en un área no pueda superar el umbral máximo (4 metros cuadrados por trabajador). Este caso es una planificación supeditada a que este valor no se supere. Esta modalidad implica una reestructuración de las operaciones (se realiza por parte de un servicio externo al índice compuesto), que tendría directa afectación al CEDI y al resto de mediciones, ya que la operación de la mercancía se alargaría en el tiempo, requiriendo más recursos y con potencial mayor impacto en el medioambiente.

El diseño del CEDI está orientado a proponer estas medidas y a representar en un valor único dichos escenarios. Como se ha dicho, la formalización de los métodos matemáticos y la construcción del índice no fueron la prioridad en este escenario. Sin embargo, se realizó una pequeña reflexión sobre cómo plantear dicho índice:

**Tabla 5.16:** Reflexión sobre formalización del índice CEDI

|  |                                |
|--|--------------------------------|
| <b>Normalización:</b>  | Re-escalado ( <i>Min-Max</i> ) |
| No se disponía de ningún elemento de referencia y se trata de un índice para monitorizar varias simulaciones. Por tanto, decidió escogerse un método de reescalado. Se realizaron varias ejecuciones previa integración de todo el sistema (series de registros previos a entorno de preproducción). De esas series, se extrajeron los valores mínimos y máximos para cada indicador y se utilizaron para realizar la normalización. |                                |
| <b>Ponderación:</b>  | Equitativa                     |
| Ya que se trata de un despliegue rápido (objetivo del escenario), y no se dispuso de más información para planter un análisis estadístico, se seleccionó la ponderación más sencilla y utilizada: equitativa.  |                                |
| <b>Agregación:</b>   | Aditiva                        |
| La agregación geométrica se excluyó teniendo en cuenta la gran variación de las unidades de medida y la dependencia mutua. Los datos seleccionados fueron: consumo energético, impacto medioambiental (CO <sub>2</sub> ) y tiempo de procesamiento de operaciones. Si bien es cierto que, para la monitorización de dichos parámetros, puede no existir una gran correlación, en los escenarios de                                   |                                |

simulación sí existe ya que los primeros indicadores (energía y contaminación) son calculados directamente a partir de los resultados del mismo servicio (PAS – *Port Activity Scenario*). Por tanto, la combinación de ambos métodos resultó en una agregación de media aritmética.

### *Fase II – Arquitectura y conectores*

De igual modo que en el resto de casos de uso, esta fase comenzó con un análisis de las necesidades modulares. No obstante, previo a estas decisiones debió tenerse en cuenta las siguientes consideraciones particulares de este caso de uso:

- A nivel tecnológico, se desea un sistema IoT de despliegue rápido con capacidad de añadir nuevas fuentes de datos si se considerara necesario en el futuro.
- A nivel de servicio, existe una aplicación de simulación (PAS) que debe ser ejecutada por el sistema a desplegar, teniendo que combinarse con el servicio CI para proveer la funcionalidad esperada.
- Se requiere la realización de una alternativa de visualización con unas funcionalidades específicas (véase fase 3).
- En lugar de una visión retrospectiva, debe ofrecerse información prescriptiva para la toma de decisiones.
- En cuanto al despliegue, debido a que se realiza bajo las restricciones del puerto adoptante dentro del proyecto de investigación PIXEL, no pudo elegirse la modalidad de despliegue. En este caso, la descentralización (*edge computing*) no pudo ser un foco de atención, y la instalación y ejecución de todos los módulos de la arquitectura (incluyendo los agentes conectores) se realizó en un entorno (*on premise*) centralizado.

Tras estas consideraciones, se avanzó hacia el diseño modular de la solución. Si bien es cierto que este puerto se encontraba inmerso en la integración de una plataforma IoT de mayor alcance (PIXEL [19]), ésta se encontraba en una fase muy temprana, requiriendo un mayor análisis y tiempo de despliegue. Por lo tanto, se trató de la oportunidad ideal para demostrar la agilidad y facilidad de adaptación de la solución propuesta (MVP). Así, se identificó la necesidad de incorporar los siguientes módulos de la solución:

**Agentes conectores de datos:** Se deseó incorporar este módulo para insertar datos de diferentes fuentes del IoT (se exponen en las siguientes páginas). Se necesita un componente que los recupere y luego los traduzca a un formato de datos unificado (sección 3.3.1).

Se requiere la existencia de un gestor contexto (componente habitual del IoT – véase sección 3.3.2) cuyo objetivo es recoger datos en tiempo real de diferentes fuentes en un elemento centralizado. Este elemento permite controlar y validar el

formato de la información recibida, por lo que también fue requerido en este escenario como un módulo de control de calidad de datos.

Al tener que aplicarse cálculos de simulación sobre datos históricos, el despliegue requería un módulo de persistencia (véase sección 3.4.1). Además, parte del procesamiento está supeditado a la definición de una serie de formulario a rellenar por el usuario en el puerto. Por tanto, esto implica la existencia de un módulo de visualización con formularios de introducción y un almacenamiento para recoger la información introducida.

Por último, el sistema debe ser capaz de ejecutar un servicio (encapsulado en Docker) y aprovechar la salida del mismo para ejecutar el cálculo del CI (también empaquetado en imagen Docker). Por tanto, se requiere cierto nivel de orquestación basado en metodologías claras, transparentes y configurables (véase – sección 3.4.2).

Con todo, un esquema de la arquitectura desplegada puede observarse en la siguiente figura:

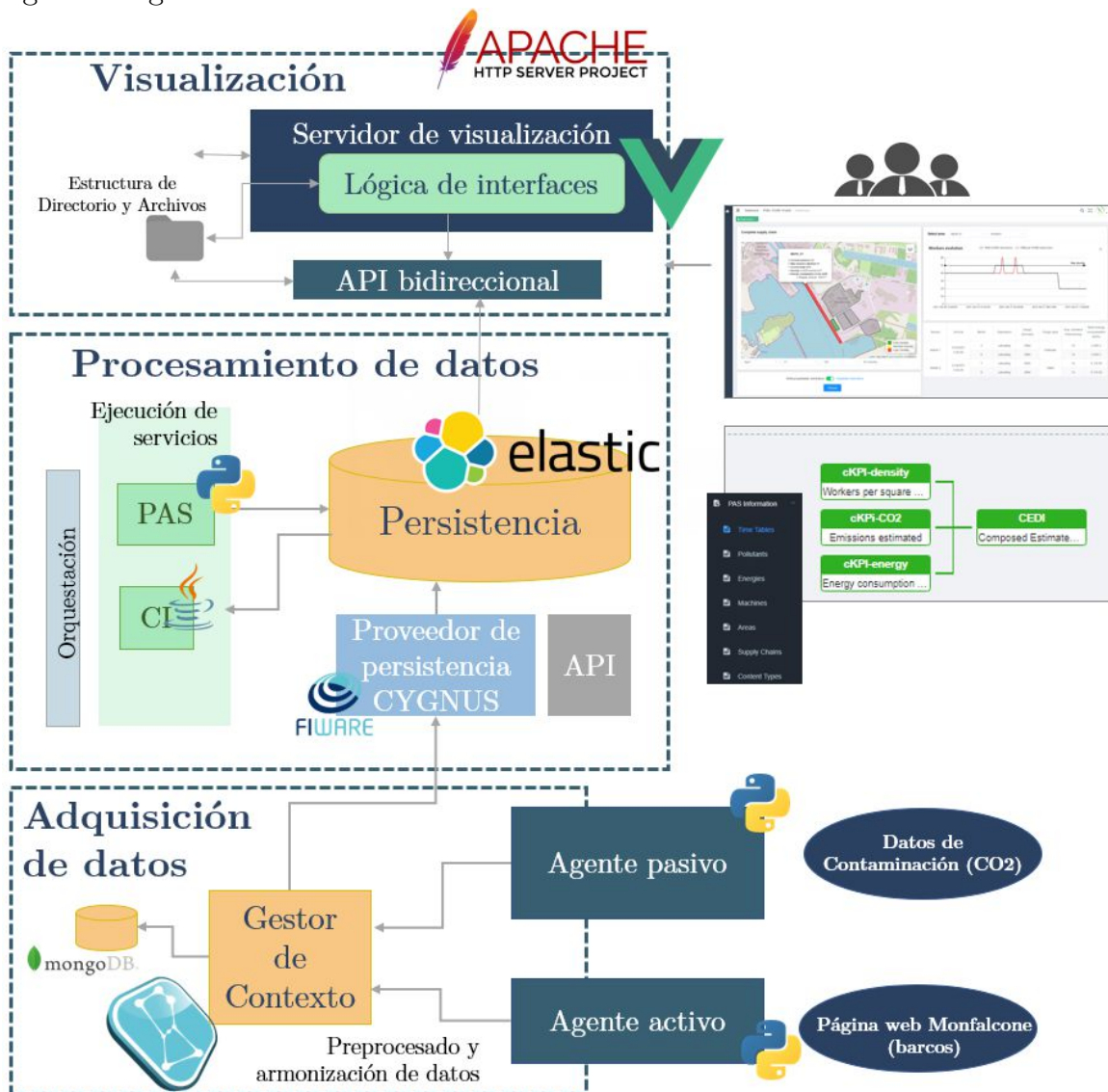


Figura 5.70: Diseño de arquitectura de despliegue para el escenario CEDI

En este momento, se procedió a instalar los elementos esenciales de la plataforma, llevando a cabo las comprobaciones indicadas en la metodología (sección 5.2.3).

No obstante, la arquitectura carece de uso si no se le conectan datos IoT para dar valor y poder ejecutar dichos servicios. A continuación, se expone una pequeña referencia sobre los datos en tiempo real, y la implementación de los agentes asociados. Cabe destacar que para su desarrollo software se empleó la librería *pyngsi*.

### Contaminación atmosférica: CO<sub>2</sub>

En el contexto de una terminal portuaria marítima, el estudio de la evolución de los valores de CO<sub>2</sub> (aunque no puedan atribuirse directamente a las operaciones de la terminal únicamente) podría ayudar a caracterizar los efectos de las restricciones en la huella medioambiental. Para obtener estos datos, es preferible el uso de sensores. En este sentido, se realizó la conexión con el mismo sensor que se utilizó en el escenario PEI (véase sección 5.4). En este caso, se equipó esta sensorización y se llevó a cabo un proceso análogo de integración de la información.

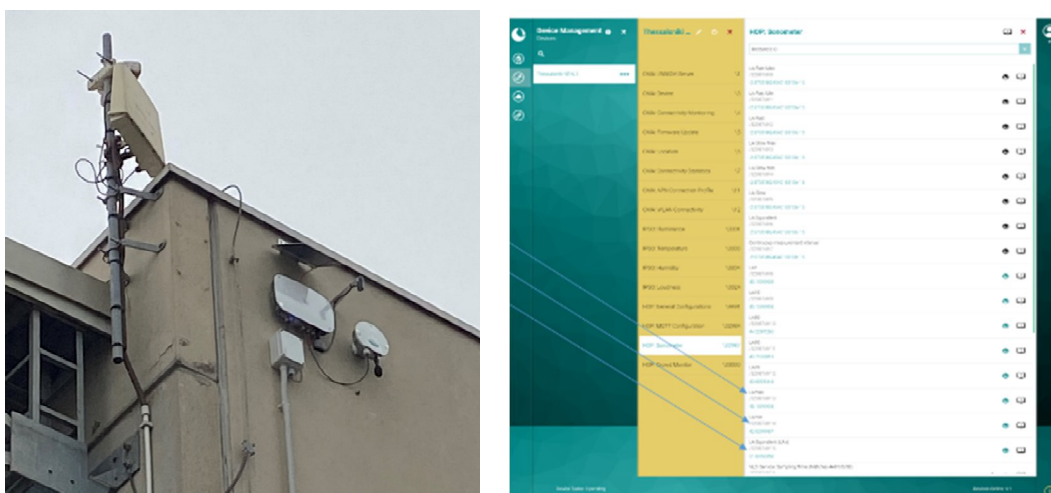


Figura 5.71: Sensorización CO<sub>2</sub> escenario CEDI

### Llamadas de los buques para su operación (*vessel calls*)

Se trata de la información esencial requerida para realizar este escenario de validación. Los capitanes de los buques deben informar con cuatro días hábiles de antelación antes de llegar a un puerto de destino mediante la llamada notificación ANOA (*Advanced Notices of Arrival*). Desde el punto de vista de los puertos, conocer el número, la naturaleza, el tonelaje bruto, la hora prevista de llegada y de salida y otros detalles del buque, permitirá simular las operaciones de la terminal durante una semana, prediciendo así el número de trabajadores necesarios por zona y franja horaria (este es el valor que aporta el servicio intermedio -PAS- documentado en las siguientes páginas).

Esta información puede llegar por diferentes vías, siendo la más habitual la conexión a un servidor remoto (web-HTTP, FTP, vía APIs u otras). Sin embargo,

existen otras formas de recoger estos datos en caso de que las llamadas no estén disponibles. Aquí entra de nuevo en juego el IoT, ya que, por ejemplo, datos AIS [514] pueden considerarse un proveedor de datos bajo la comprensión del IoT. Mediante el uso de equipos específicos (transceptores y convertidores), el AIS podría utilizarse para rastrear los buques en la zona marítima cercana de un puerto para incluirlos en la simulación de la herramienta. No obstante, en este caso no fue necesario, ya que el puerto de Monfalcone ofrece de forma pública los anuncios de los barcos próximos a procesar (a través de su página web):

*Crowd monitoring.* El mercado ofrece varias opciones para el seguimiento de las personas en una determinada zona terrestre de una ciudad, destacando el uso del análisis de recuento basado en smartphones [4]. Para el caso propuesto en este trabajo, esta información podría ser útil para ser comparada con las predicciones según los resultados del servicio PAS y para fomentar la capacidad de los gestores portuarios de reaccionar rápidamente en caso de que no se cumplan las restricciones. Se destaca que, en el momento de validación de la solución en este escenario, el puerto de Monfalcone se encontraba en fases de análisis para la incorporación de este tipo de datos de manera sensorizada. No obstante, finalmente no se llegó a implementar un agente de conexión en tiempo real de esta información.

### Servicio intermedio: PAS – *Port Activity Scenario*

Este caso de uso está constreñido al uso de un software (encapsulado como imagen Docker) que fue desarrollado dentro del proyecto PIXEL. Este programa, conocido como Escenario de Actividad Portuaria (*Port Activity Scenario - PAS*) [515][516], consta de varios bloques que, en conjunto, modelan la programación y el rendimiento de las operaciones de la terminal necesarias para operar eficazmente un buque (carga/descarga) y los procesos implicados, incluyendo el tiempo asignado, la energía consumida, la maquinaria empleada y el personal dedicado.

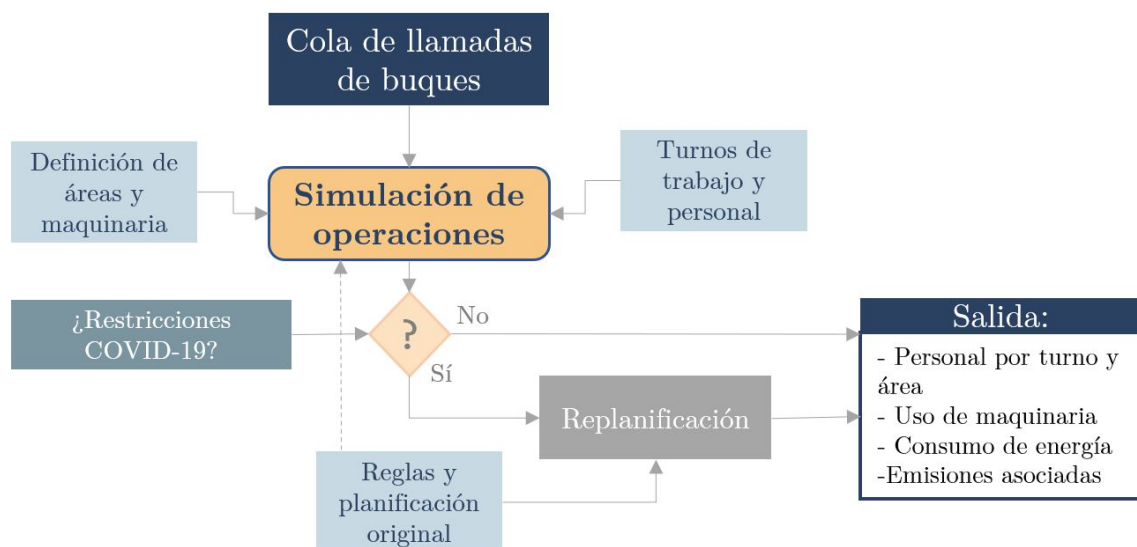


Figura 5.72: Esquema funcionamiento servicio PAS

Para la utilización de este programa en la arquitectura propuesta en este trabajo, es necesario realizar algunos ajustes. Integrado como modelo en la capa de aplicación del sistema, el PAS fue mejorado para incluir las restricciones de COVID-19 en sus cálculos para obtener una simulación y permitir el seguimiento de la densidad de trabajadores por zona y por hora (fuera del alcance de la tesis). El procedimiento a seguir fue crear dichas restricciones e incluirlas en la definición de la cadena de suministro de las operaciones del puerto en el que se desea desplegar la herramienta. Esto se realiza a través de la configuración de los parámetros del puerto (mediante interfaz facilitada por el módulo de visualización). Por diseño, la configuración del PAS permite definir un número máximo de trabajadores por zona, proceso y máquina utilizados en la terminal.

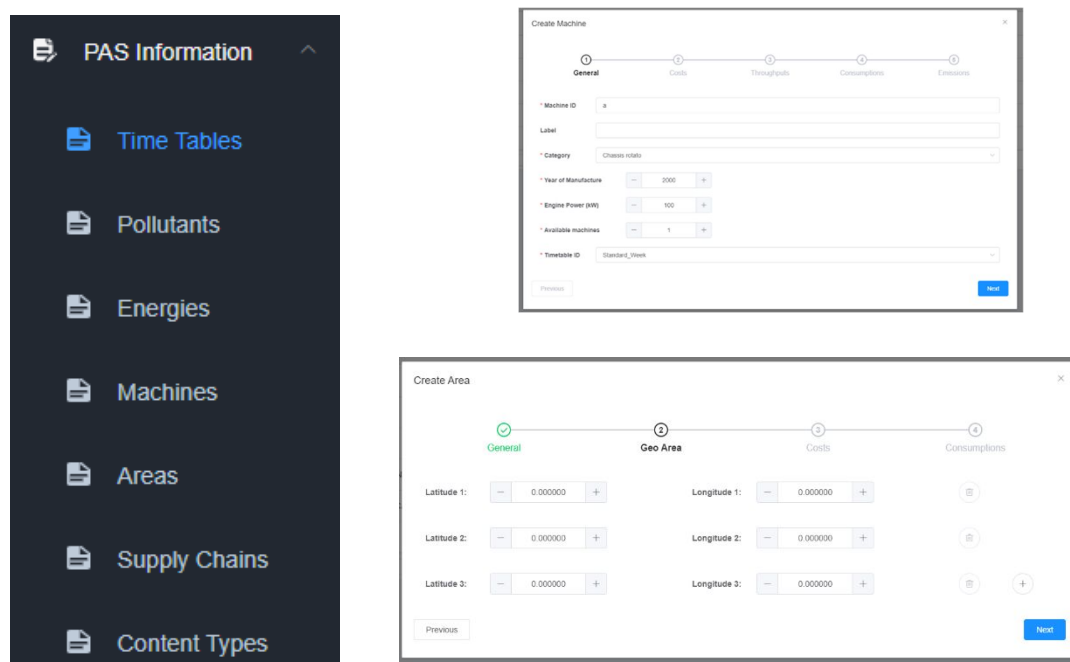


Figura 5.73: Formularios de introducción de datos - PAS

De este modo, dado un plan de buques diario/semanal, este servicio intermedio calculará los resultados sobre las operaciones (*supply chain*): (i) según la programación habitual, basada en la operativa y las prioridades de la terminal, (ii) teniendo en cuenta esas limitaciones, lo que dará lugar a una programación específica que diferirá de la habitual.

Tecnológicamente, el modelo PAS se compone de varios scripts Python personalizados que interactúan entre sí para simular la programación de la terminal. Para la integración, se utilizó Docker para containerizar la aplicación, facilitando el despliegue dentro del esquema de la arquitectura.

Por último, es destacable el flujo que debe seguirse entre servicios para garantizar un correcto funcionamiento del despliegue de la solución. Una vez ejecutado (orquestrado por el módulo correspondiente de la arquitectura – sección 3.4.2), los resultados son almacenados en el índice indicado de la base de datos.



Tras su persistencia en dicho índice/documento, el servicio CI es iniciado especificando los datos de entrada, que provienen de dicho índice. Para ello, un agente intermedio fue generado para garantizar que la información se encuentra en el formato esperado por el CIC.

El flujo se ilustra brevemente en la siguiente figura:

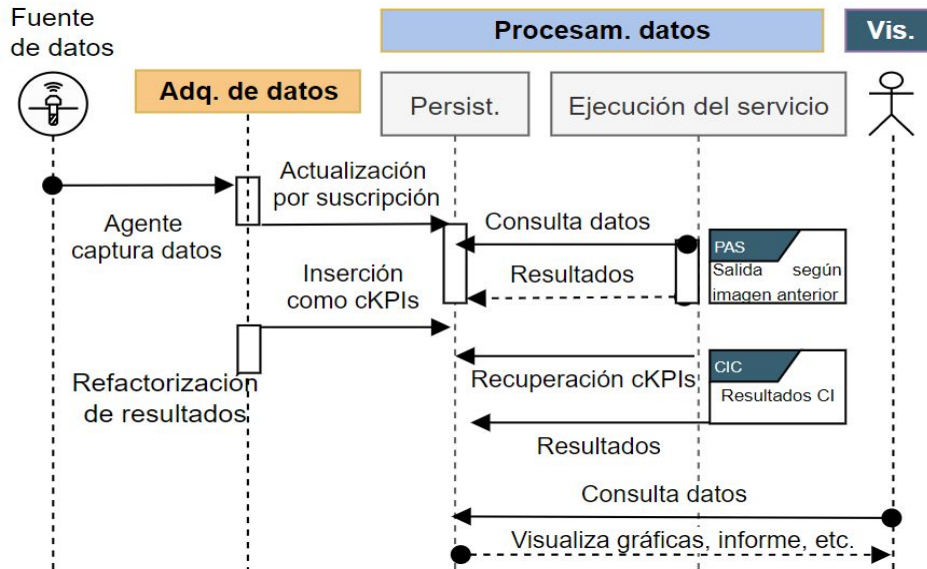


Figura 5.74: Flujo de servicios en escenario CEDI

Por último, el módulo de visualización representa la información (utilizando unas vistas diferentes a lo definido para la solución global).

### *Fase III – Configuración y resultados*

En este punto, se realizó la configuración del índice compuesto tal y como se muestra en la siguiente figura. Como se anunció, se compone de la información de simulación de consumo energético, impacto medioambiental y densidad de trabajadores en las áreas concretas. El trabajo de disección de la información desde el PAS para su correcta inserción como *cKPIs* en el elemento de persistencia es responsabilidad del agente de conversión mencionado. En concreto, se planteó la ejecución del servicio CIC una vez para cada área que desee ser estudiada desde el punto de vista del CEDI y representada en la interfaz.

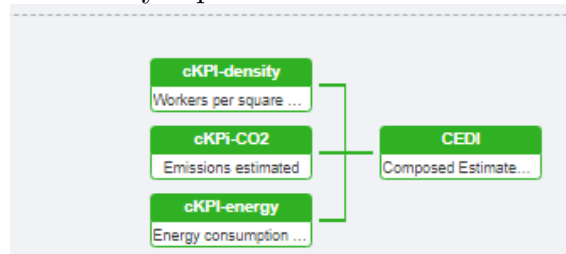
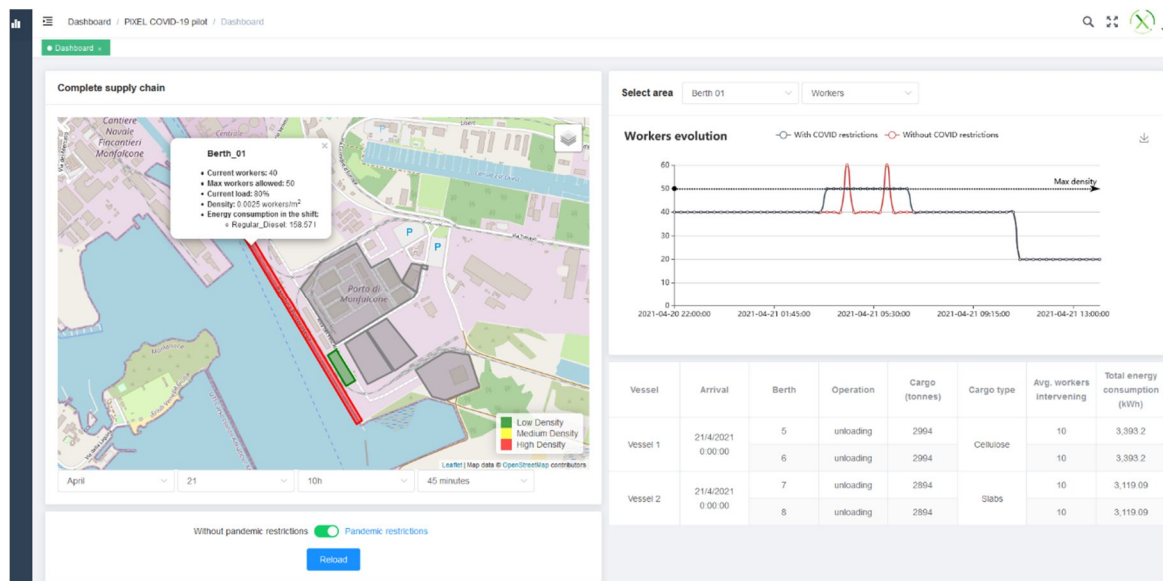


Figura 5.75: Estructura del árbol para el cálculo del CEDI



Como se ha mencionado, se destaca que debió realizarse un desarrollo adicional sobre el módulo de visualización prediseñado en la solución (sección 3.5) para reflejar las características particulares de este despliegue. En concreto la interfaz incluye (véase Figura 5.76: Interfaz específica de visualización - CEDI): (i) una vista de mapa de la terminal portuaria, detallando la ocupación de los trabajadores, (ii) un gráfico multilínea para monitorizar la evolución en el tiempo de dicha densidad, el consumo de energía (en kWh), el CEDI y las emisiones de CO2 por área (suponiendo con/sin restricciones de la pandemia,) (iii) una tabla informativa que proporciona estimaciones asociadas a cada buque a operar y (iv) un código de colores por umbral y área que sirve como sistema de alerta cuando el CEDI supera dichos valores en esa área y periodo de tiempo.

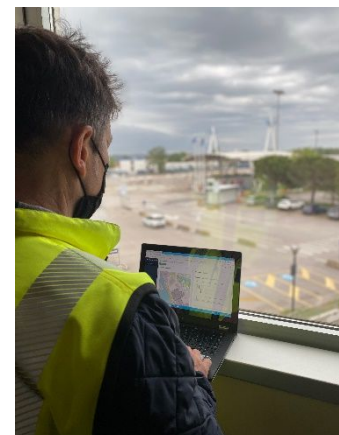


**Figura 5.76:** Interfaz específica de visualización - CEDI

Técnicamente, se ha utilizado la librería *Leaflet.js* (como se hizo en el escenario de validación TCI – sección 5.3), combinada con datos de *OpenStreetMap*. En cuanto a las tablas y las líneas de evolución, se ha empleado la librería *Apache Echarts* [517].

Idealmente, este despliegue podría utilizarse para realizar una replanificación automática de las operaciones de terminal para asegurar el cumplimiento de las normativas, sin embargo, el alcance de este piloto ha sido sólo proporcionar una simulación (herramienta de apoyo a la decisión) basada en los datos reales de los buques que llegan y las operaciones previstas en la terminal.

**Figura 5.77:** Utilización de la simulación planteada – escenario CEDI



## 5.7. Instanciación híbrida para la recomendación de asignación en transporte logístico por carretera

Este escenario de validación se centra en la recomendación de asignación del tándem conductor-camión en una orden de transporte de un contenedor de mercancía desde/hacia el puerto de Valencia. En particular, trata de ofrecer un valor sobre cuán óptima es una selección a través del índice compuesto.

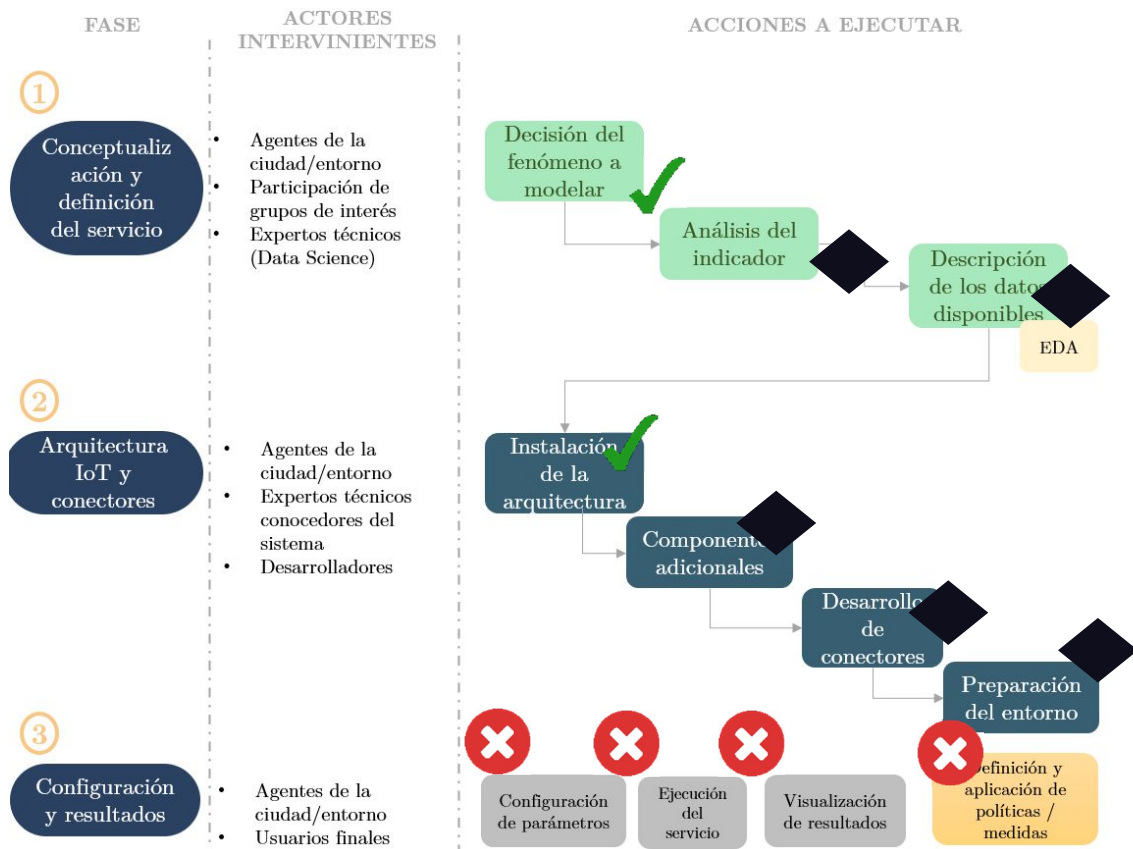


Figura 5.78: Fases completadas en el escenario OAI

Conviene destacar (véase la figura anterior) que este escenario de validación aún no ha sido completamente finalizado. El motivo es que se encuentra enmarcado en el proyecto de investigación ASIGNAT, aún en fases iniciales/intermedias de su ejecución (finalización esperada: octubre 2023). No obstante, ha decidido incluirse esta actividad (que está siendo llevada a cabo por el candidato y otros miembros del equipo del proyecto) para evidenciar dos aspectos: (1) cómo la arquitectura planteada encaja perfectamente para actuar como base de un Gemelo Digital (*Digital Twin*, DT) y (2) la versatilidad de la solución en general para adaptarse a diferentes casuísticas y problemas, demostrando que la conjunción de IoT e índices compuestos resulta de gran utilidad en una amplia variedad de escenarios.

### 5.7.1. Contexto

Enmarcado dentro del proyecto de investigación ASIGNAT, el doctorando descubrió la situación actual de la asignación de transporte de mercancías tras una orden de pedido desde el puerto de Valencia. El caso particular que el doctorando analizó corresponde a una empresa local (colaboradora en dicho proyecto), pero la casuística observada puede extrapolarse a otros casos en Europa y el mundo e incluso a otros puntos de la cadena logística, ya que el transporte terrestre conectado con un entorno portuario es uno de los casos de uso más complejos dadas las particularidades de las operaciones y el número y tipo de agentes que intervienen.

En este sentido, las órdenes son recibidas a través de herramientas informatizadas de gestión (PCS – *Port Community System*), incluyendo información sobre el contenedor, la mercancía, etc. El ente logístico publica estas órdenes para que empresas de servicio asociadas puedan proponer ofertas para efectuar el transporte necesario. En este sentido, dichas empresas son las encargadas de analizar cada orden y asignar a conductores y camiones de entre su plantilla (o externalizado) y su flota. En los casos habituales, los conductores envían periódicamente información al departamento de tráfico de su empresa acerca de su ubicación y el tiempo estimado para llegar al siguiente punto de un viaje (*Estimated Time of Arrival, ETA*), así como posibles incidencias en ruta. En algunos casos existen herramientas automatizadas que gestionan estas comunicaciones, pero es usual que se realicen a través de llamadas telefónicas o mensajería instantánea. Además, es también una práctica común en este tipo de empresas gestionar la asignación de conductor y camión a una orden a partir de información no registrada, tal como experiencia, conocimiento de casuísticas, sensaciones, etc. Puede realizarse esta afirmación ya que la empresa con la que se ha colaborado posee una flota de más de 180 vehículos (no se trata de una empresa pequeña).



Figura 5.79: Representación del contexto del escenario OAI

Observando esta realidad, el objetivo que se planteó fue innovar mediante la implementación de un sistema IoT que sirviera para la recopilación de datos heterogéneos (útiles para dicha asignación) de forma automática y en tiempo real, siendo mostrados al usuario y con un funcionamiento de Gemelo Digital. Un gemelo digital es una representación virtual de un elemento existente físicamente (sea este un objeto, un sistema o, en el caso de este escenario de validación, el estado en un momento determinado de una flota y del proceso de decisión para la asignación de transportista), cuyo estado y su comportamiento son monitorizados y simulados. Esta copia digital es única y se actualiza para reflejar los cambios del mundo real, generalmente de forma periódica (incluyendo también opciones bajo demanda).

Disponiendo de lo anterior, las empresas de servicio podrían ser capaces de automatizar la información relacionada previa a la asignación (que es un proceso que seguirá siendo supervisado por humanos), pudiendo establecer ciertas reglas para la recomendación de la mejor opción (tándem camión-conductor) en cada caso. Como estas distinciones pueden implicar varias dimensiones, y se quiere ser capaces de dotar de cierta complejidad (diferente ponderación, etc.), se plantea el uso de indicadores compuestos para guiar la toma de decisiones en la asignación de tándem camión-conductor a una orden de pedido.

### 5.7.2. Implementación de la solución

A continuación se exponen los hallazgos, decisiones y resultados obtenidos durante el escenario de validación asociado al índice OAI.

#### *Fase I – Conceptual y servicio*

Dentro de ASIGNAT, el objetivo del subsistema del Gemelo Digital es el de representar en todo momento el estado actual de la flota de una empresa de transporte, juntamente con las órdenes que deben de procesarse y todos aquellos parámetros que influyen en la decisión de asignación de conductor y vehículo. No obstante, los Gemelos Digitales no buscan únicamente ofrecer información correcta y actualizada, sino también modelar/simular el comportamiento de dicho objeto/elemento acorde a cómo sería en el mundo real, apoyado con mediciones y a través, también, de herramientas matemáticas.

En este sentido, se especificó que debía existir algún tipo de funcionalidad adjunta a las competencias del Gemelo Digital del sistema global IoT a implementar que fuera capaz de efectuar una “recomendación” de asignación justificada, basada en el comportamiento habitual de dicho proceso de decisión.

Para exponer la utilización del servicio CIC desarrollado en esta tesis dentro del escenario de validación en ASIGNAT, resulta necesaria la introducción de:

#### **Las fuentes de datos disponibles:**

- ERP de transporte: que proporcionará las ordenes de transporte sobre las que se deben realizar las asignaciones, los datos de conductores,

vehículos, empresas, puntos de carga/descarga, asignaciones reales

- Aplicación móvil portada por los conductores de los camiones que se encarga del tracking por GPS, determinando el ETA a cada uno de los puntos relevantes del viaje y ubicando su posición real.
- Tacógrafo Digital: El tacógrafo digital de los camiones registra información valiosa sobre los tiempos de conducción de los conductores. No obstante, los sistemas más extendidos de descarga de información del tacógrafo digital no permiten la descarga de información en tiempo real, sino que se suele hacer de forma manual, y los que permiten una integración en tiempo real son escasamente utilizados.
- Información interna (informatizada o no) de la empresa de transportes, incluyendo la flota, la información relacionada con los conductores, sus condiciones, restricciones o preferencias particulares, etc.

#### **Los elementos informacionales clave:**

A la hora de plantear el desarrollo de un índice compuesto para recomendar una asignación, se realizó un ejercicio de entender los datos (concretos) más importantes de acara a su construcción. Los puntos siguientes fueron el resultado de un análisis tras la tarea de definición de los modelos de datos:

- *TransportOrder*: representa la orden de transporte recibida (del ERP del transportista, de una plataforma PCS, etc.).
- *TransportCargo*: contiene la información relativa al contenedor de la orden de transporte.
- *TransportInstructions*: describe las instrucciones del viaje: fechas, puntos de carga y descarga.
- *POI*: Puntos de interés. Describe los puntos (empresas, lugares de carga y descarga) donde debe ir el vehículo con la mercancía (por ejemplo, un contenedor).
- *Company*: contiene los diferentes agentes que participan en una orden de transporte.
- *Vehicle*: almacena todos los vehículos, ya sean cabezas o remolques.
- *Driver*: representa la información relativa a los conductores.
- *Condition*: clase para categorizar las diferentes variables de un conductor o vehículo que deben tenerse en cuenta a la hora de realizar una asignación.

#### **El proceso de servicios que culmina con la utilización del CI:**

El sistema IoT planteado en este escenario debe incorporar lo que fue denominado como: planificador de asignaciones. Debía consistir en un recomendador enfocado a la toma de decisiones en la asignación para aumentar la productividad, eficiencia y competitividad a largo plazo. Para ello, se estableció que

esta recomendación debía formalizarse a partir del establecimiento y automatización de una serie de reglas y objetivos. Lo que hasta el momento residía en elementos no trazables como experiencia, información en papel o comunicaciones sin registrar, se encontraría según esta propuesta embebido dentro del gemelo digital. Este sistema debía ser capaz, mediante un motor de reglas, de generar asignaciones recomendadas. Cabe mencionar que dentro de los planteamientos de dicho proyecto de investigación existía también un módulo de aprendizaje automático con un objetivo análogo (con la diferencia que este último debía realizar un entrenamiento de un modelo a partir de un histórico, mediante sus propios algoritmos de *Machine Learning*).

Por tanto, el primer elemento (servicio) a incluir dentro de dicho sistema fue un motor de reglas con información sobre las condiciones de las diferentes entidades que permitiera cribar qué conductores/camiones podían ser recomendados para procesar una orden de pedido de transporte terrestre.

No obstante, en la mayoría de las ocasiones, más de un conductor existe que puede cumplir las condiciones de reglas creadas: transporte nocturno, de fin de semana, de mercancías peligrosas, con refrigeración (freezer)... Se antojó entonces un mecanismo para refinar la recomendación y proponer el mejor candidato de entre los disponibles. Según lo estudiado en secciones anteriores (1.1 y 4), el servicio desarrollado para el cálculo de CIs resultó una opción ideal para abordar esta “priorización” acorde a unos KPIs de negocio expresados por el actor de interés. Éstos pueden concebirse como *cKPIs* a alimentar el índice planteado (OAI – *Optimal Assignment Index*), que serían alimentados por la información existente en el Gemelo Digital. Resulta una decisión acertada ya que, como se ha visto en la sección 4.1.1, un CI (si se encuentra bien diseñado e implementado) sirve para introducir componentes de ponderación y comparación entre diferentes elementos (conductores/camiones) de cara a un fenómeno de la vida real (un mejor rendimiento a nivel de negocio al realizar la asignación a alguno de ellos de una orden de transporte).

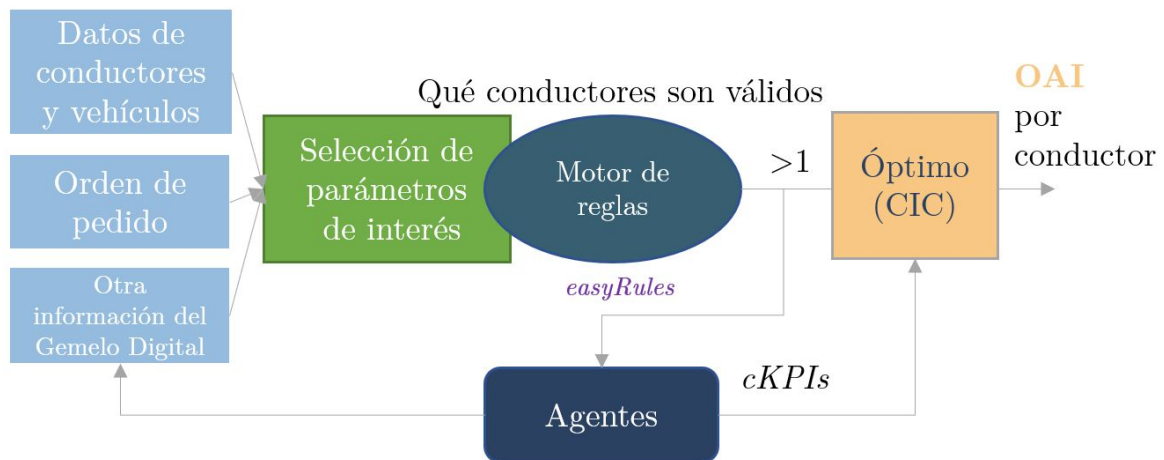


Figura 5.80: Ciclo de servicios en el escenario OAI

Por tanto, desde un punto de vista de ciclo de servicios, la plataforma IoT debería recoger la información, ejecutar un motor de reglas a partir de la información introducida y, de los conductores que resultaran elegibles (los que satisficieran dichas reglas), obtendrían un índice (OAI) con valor entre 0 y 1 que representaría cuán óptima es su asignación en función de los *cKPIs* estimados asociados a dicha dupla.

Como se ha indicado al comienzo de la sección, este escenario se encuentra en fases iniciales, por tanto aún no se ha realizado un estudio completo de cómo debería construirse el **OAI**, las implicaciones que existen, un posible EDA de los datos comentados, correlaciones entre ellos, consulta en la literatura de propuestas relacionadas/alternativas o validación de concepto por parte de actores involucrados. No obstante, se ha realizado una primera tentativa de implementación tecnológica para la validación del concepto, apoyada en la definición de los siguientes indicadores iniciales:

- *Distancia de viaje*: El valor de la distancia total necesaria para procesar la orden, que puede jugar un factor determinante para otros aspectos del negocio.
- *Posición relativa*: cuán lejos se encuentra el conductor del punto de recogida de la orden del pedido. Este cKPI estará alimentado por fuentes de IoT como los datos GPS en tiempo real de la posición de los camiones.
- *Tiempo de descanso*. A partir de los datos GPS y de reporte por aplicación de los conductores, este valor podrá ser de interés a la hora de recomendar la asignación de esta dupla.
- *Tiempo de conducción*: Análogo al anterior.
- *Conductor propio/no propio*: Valor cuantificable según escenario y ponderable dinámicamente para especificar cómo se valora que la orden sea procesada por un miembro de la plantilla de la empresa o subcontratado a un conductor autónomo o a otra empresa.
- *Facturación acumulada*: Dato relativo a la dupla en cuestión.
- *Cobertura de facturación*: Porcentaje de vehículos de la flota que obtiene una facturación igual a la media  $\pm$  una desviación típica, medido en un intervalo mensual. Se calcula dinámicamente asumiendo que dicha dupla fuera a ser asignada.
- *Consumo estimado de combustible*: Se trata de un indicador proveniente de varias entidades de datos, por tanto, no es directo y requiere agente.
- *Rendimiento del servicio*: Facturación media de los viajes respecto a los kilómetros recorridos, medida en un intervalo temporal (día, semana, mes)

Se han dividido de forma tentativa en tres categorías, tal y como perfila (Figura 5.81: Árbol tentativo de construcción de índice OAI) una primera introducción en la interfaz de la herramienta instalada. Atendiendo a las primeras



definiciones de indicadores iniciales propuestas, se han clasificado (nivel subíndice) en geográficos, temporales y de lógica de negocio.

La introducción de esa información (*ckPIs*) deberá ser realizada por agentes (recomendado: *pyngsi*) que implementarán la lógica necesaria para convertir de datos de entidades (véase al inicio de esta subsección) en indicadores relevantes.

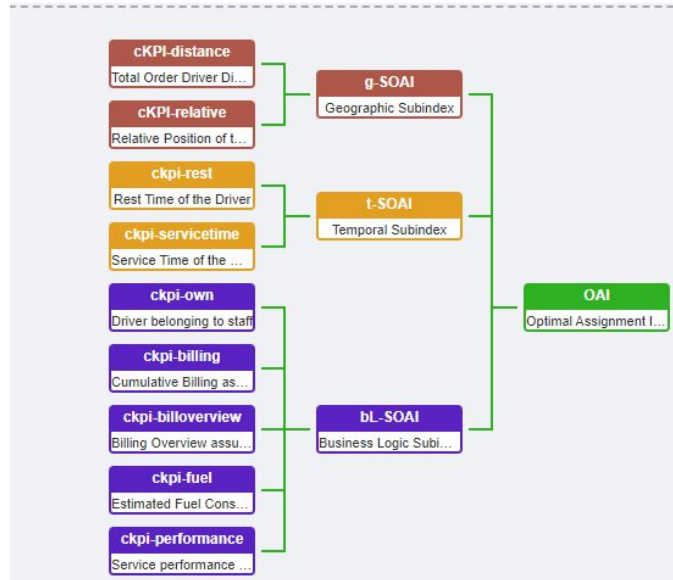


Figura 5.81: Árbol tentativo de construcción de índice OAI

Está previsto el correspondiente estudio y formalización del índice compuesto siguiendo la metodología expuesta en la sección 4.1.2.1 y su despliegue acorde a los pasos establecidos en esta tesis (sección 5.2).

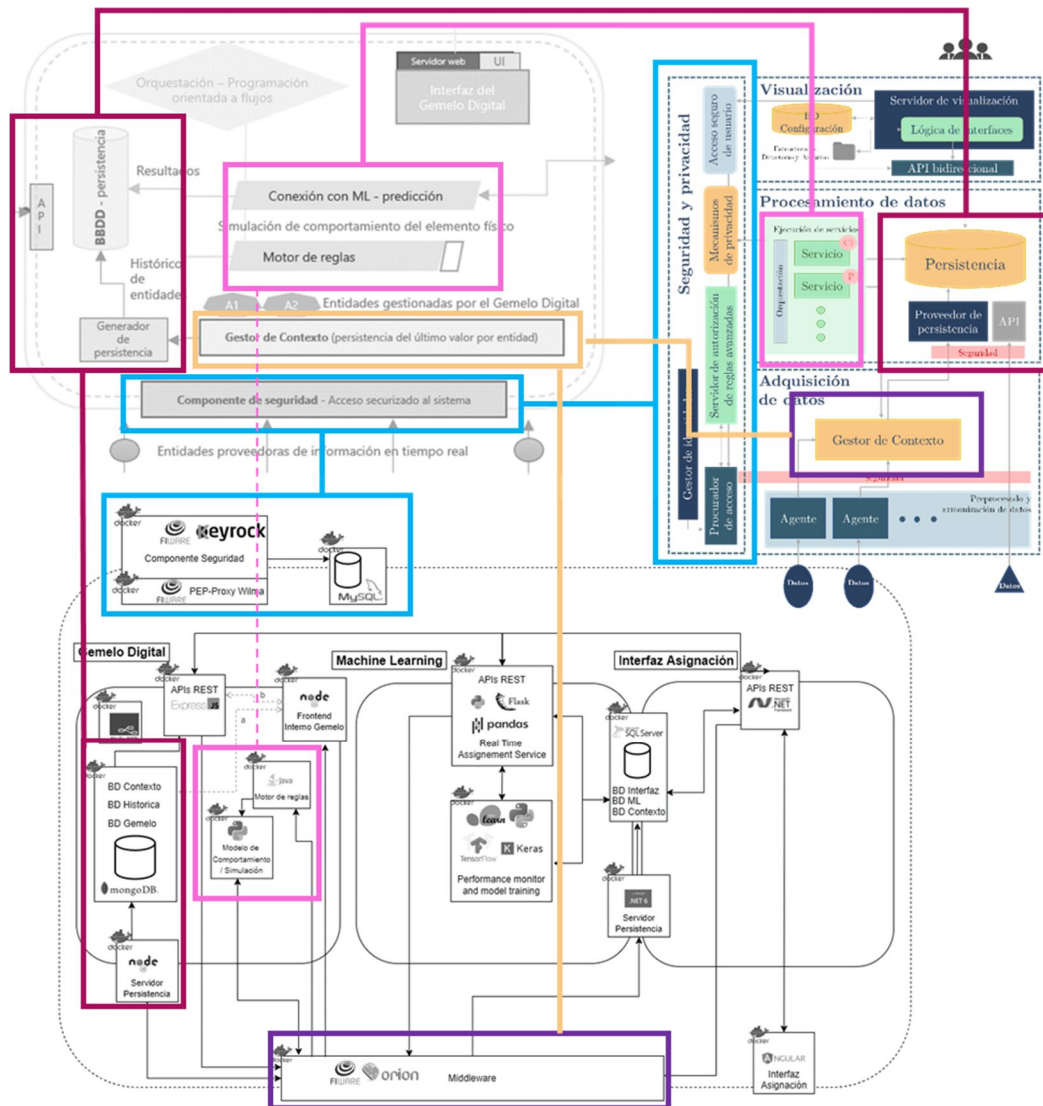
### Fase II – Arquitectura y conectores

En este caso, al contrario que en otros escenarios de validación, la arquitectura no fue discutida tras la definición del indicador. Como se ha comentado, el proyecto donde se enmarca ya poseía como objetivo la implementación de un Gemelo Digital. Según el doctorando, la solución IoT propuesta en esta tesis componía una base ideal para la construcción de este DT. Por tanto, en lugar de un método deductivo (índice-necesidades-arquitectura), utilizó una aproximación inductiva (solución IoT-Gemelo Digital-servicio de CI) y decidió transponer la arquitectura IoT planteada (véase sección 3) para dar cobertura a los requerimientos del DT, facilitando asimismo la utilización de índices compuestos sobre la misma.

En este sentido, la imagen de la página siguiente ilustra cómo se emplearon los conceptos y los módulos en el diseño de (esquina superior izquierda) el bloque del Gemelo Digital y (imagen inferior) del esquema global de la plataforma IoT para sostener el funcionamiento de todo el sistema objetivo del proyecto.

Tras la figura se realiza una pequeña reflexión sobre las necesidades de dichos componentes, relacionándose casi de manera unívoca y directa con los módulos de la arquitectura planteada en esta tesis.





**Figura 5.82:** Relación de diseño del Gemelo Digital con la solución propuesta

Al tratarse de un sistema en tiempo real que se nutrirá de fuentes de información, se requiere la presencia de un elemento concentrador, que deberá ser utilizable y accesible desde dichas fuentes de información (gestor de contexto – sección 3.3.2). Se requieren elementos para enviar esta información de forma estructurada acorde a unos modelos de datos (agentes – sección 3.3.1). La persistencia de datos es necesaria en este sistema (véase sección 3.4.1) ya que, desde la perspectiva de sistema completo del proyecto, existe el módulo de Aprendizaje Automático que requerirá la alimentación de datos históricos para su entrenamiento, del mismo modo que el Gemelo Digital necesitará la consulta de información previa al momento actual para aplicar sus reglas de negocio. Como se ha comentado, una de las funciones principales del Gemelo Digital es la “simulación” o “emulación” del comportamiento del sistema a replicar (ejecución

de servicios – sección 3.4.2). Estos componentes internos están íntimamente relacionados entre sí, por tanto se requirió la incorporación de una herramienta de conexión visual (basada en flujo) entre los diferentes componentes, donde fue seleccionada Node-Red (esta parte está fuera del alcance de la solución de esta tesis. Por último, se ha incluido una capa de seguridad, necesaria para cualquier sistema de uso en producción (módulo planteado también en esta tesis)

### *Fase III – Configuración y resultados*

La fase de configuración del índice compuesto y de obtención de resultados aún no ha sido llevada a cabo en este escenario de validación. En adelante a esta actividad, se ha realizado (en colaboración con otros miembros del equipo del proyecto) un refinamiento de la interfaz visual, que se ha orientado a los objetivos de visualización del Gemelo Digital en lugar de estar centrada en la representación del índice compuesto, aportando un mapa de visualización de la flota e información gestionada por el DT. La recomendación generada por el servicio CIC se incrustaría según un código de colores en la tabla de asignaciones.

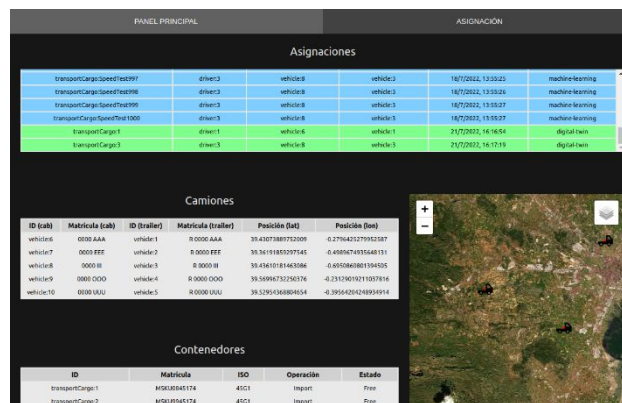


Figura 5.83: Interfaz relacionada con el escenario OAI (1)

En particular (y esto es relevante para los planteamientos de la tesis), la interfaz permitirá la configuración y selección de reglas de forma dinámica para servir como paso previo a la ejecución del CI (aplicaría el segundo caso de motor de reglas mencionado en la Figura 4.46: Servicio de recomendaciones basada en reglas sobre indicadores (II)).

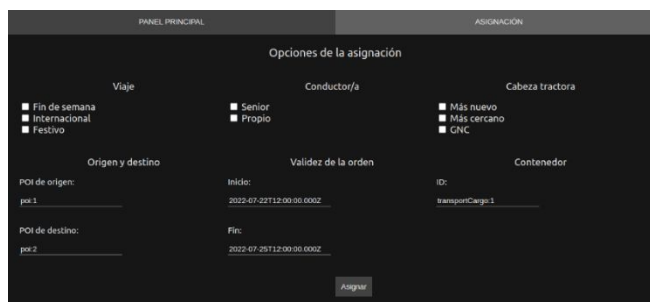


Figura 5.84: Interfaz relacionada con el escenario OAI (2)

# Capítulo 6

## Consideraciones hacia la transferencia tecnológica

Según la definición de la CE (por parte del JRC), el proceso de transferencia tecnológica se refiere específicamente a la transmisión de los resultados de la investigación científica y tecnológica **financiada con fondos públicos** al mercado y a la sociedad en general, junto con los conocimientos y procedimientos asociados, y es, como tal, parte intrínseca del proceso de innovación tecnológica [6].

Gran parte de los desarrollos expuestos en esta tesis doctoral provienen directamente de acciones con financiación pública, pudiendo considerarse el resultado un elemento tecnológico potencialmente transferible para proveer de un servicio a la sociedad. En particular, la solución ha sido diseñada como un elemento de soporte a la toma de decisiones, permitiendo abstraer -en un valor sencillamente

interpretable- un fenómeno complejo de la realidad, a partir de la adquisición de información de diferentes fuentes (IoT) en tiempo real y apoyado en procesos formales de establecimiento y cálculo de indicadores compuestos.

Esta sección realiza un repaso a dicho proceso de transferencia tecnológica, esbozando un camino (para este resultado) hacia un perfil de explotación en entornos reales operativos que recorre una primera definición de un posible “producto”, junto con una valoración inicial que permita el establecimiento de un plan de negocio futuro a partir del mismo.

## 6.1.Perspectiva de financiación

Para que una tecnología proveniente de la investigación pueda ser correctamente transferida a su uso en el mercado, ésta debe tener un nivel de madurez suficiente. Hoy en día, la escala más aceptada para medir el grado de madurez de una tecnología es la denominada TRL [518]. Esta escala fue introducida por la NASA en la década de 1970 y, desde 2014 (tras unas pequeñas modificaciones y refinamientos) fue adoptada por la Comisión Europea, en el marco del programa H2020, como una herramienta para establecer los puntos de alcance de las convocatorias de financiación.



**Figura 6.1:** Niveles de madurez tecnológica - TRL

El principal grueso de lo que tradicionalmente se conoce como “investigación” se encuentra entre los TRLS 1-3. En estos estadios de madurez, las universidades y centros tecnológicos juegan un papel crucial, ya que son la entidades con suficiente capacidad de infraestructura y personal para abordar tareas de investigación de bajo nivel complejas y sin garantía de éxito. No obstante, estas mismas entidades son las que más necesitan del apoyo público para la consecución de estos objetivos, principalmente en Europa, donde el modelo de investigación público prevalece sobre el privado.

En este sentido, estas primeras fases están bien cubiertas por ciertos instrumentos de financiación y por parte de iniciativas dentro del ámbito

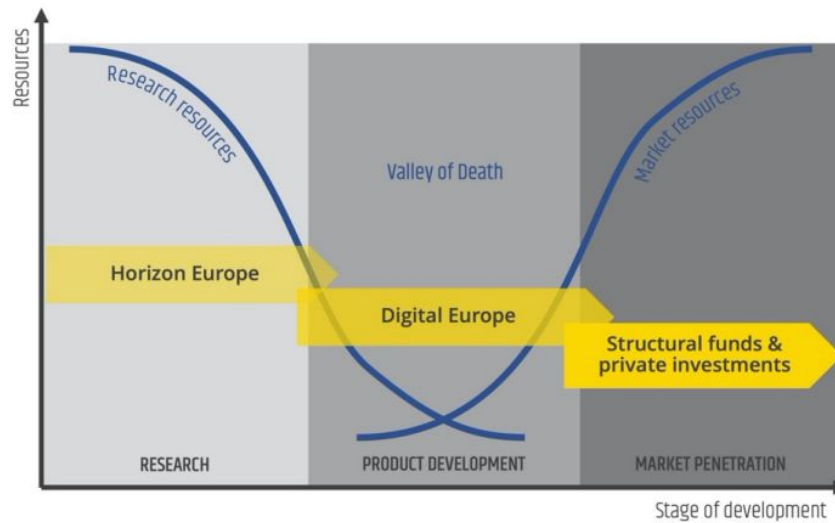
académico: líneas de investigación, propuestas de laboratorio, etc. así como (más recientemente) por parte de la comunidad *open source*. A nivel europeo, destacan los programas ERC [519] y Marie Skłodowska-Curie [520], centrados en investigaciones de bajo nivel, y el programa Horizon Europe (HE) [521], heredero del exitoso programa marco Horizonte 2020 (H2020), finalizado recientemente. Las dimensiones (a nivel volumen de financiación) de estos programas son inmensas, habiendo reservado, por ejemplo HE, más de 95 mil millones de euros para su ejecución en el periodo 2021-2027, de entre los cuales más de 5 mil millones provienen de fondos NextGenerationEU [522]. Este aspecto es de especial relevancia para las perspectivas de transferencia del resultado de esta tesis, ya que el principal receptor de estas ayudas será España [523].

Pensando en clave nacional, en España también existe una gran cantidad de líneas de financiación para proyectos de investigación [524]: Retos colaboración, proyectos I+D del CDTI [525], Programa Torres Quevedo o Tecnologías Habilitadoras Digitales (THD). Además, en España existen incentivos alternativos para el fomento de la I+D+i como las deducciones fiscales para empresas. Sin embargo, como se anunciaba al inicio de la sección, la práctica totalidad de las líneas e instrumentos anteriores tienen un aspecto en común: se orientan a financiar desarrollos e investigaciones hasta TRL4, dejando el desarrollo de tecnología y demostración de sistema, normalmente, al margen.

No obstante, cabe decir que alguna de las líneas mencionadas (principalmente *Horizon Europe*) incluye el alcance hasta las pruebas de concepto. Integrar las innovaciones de las fases TRL4 y TRL5 en entornos reales es todo un desafío ya que, en muchos casos, las compañías no desean tomar riesgos ni caer en ineficiencias de recursos y prefieren optar por tecnologías consolidadas. Además, la bifurcación de actividades sin garantía de éxito es un freno en este aspecto. Es, por tanto, conveniente destacar la labor de estos programas para alcanzar grados de madurez TRL6, donde, gracias a los incentivos y ayudas públicas, entidades competitivas prueban tecnologías de reciente creación en los denominados “pilotos” o “escenarios de validación” (véase sección 5).

No obstante, existe un claro punto donde esta transición se hace complicada. Es ahí donde termina la competencia del ámbito de la pura investigación, y se requiere la entrada de prototipos en entornos (semi) operativos, la apuesta en firme por avanzar en la certificación y encapsulado de producto, la de formalización de registros de propiedad intelectual, etc. Es justamente esta etapa, la comprendida entre los TRL6 y TRL9, que se conoce como “**el valle de la muerte**” [6]. Esta se concibe como la frase clave para la transferencia tecnológica real, favoreciendo que los resultados de investigación puedan aportar un valor a la sociedad y a las empresas. Y es en ese espacio donde la financiación pública o los instrumentos existentes no están logrando dar el impulso para conseguir dicha transferencia. De hecho, existe el consenso actualmente que la excelencia de investigación en la Unión Europea no traslada (transfiere) a un alto nivel de adopción de la innovación y de

impacto en el mercado [6], llegando a haberse acuñado como término “la paradoja europea”.



**Figura 6.2:** El valle de la muerte de la financiación tecnológica

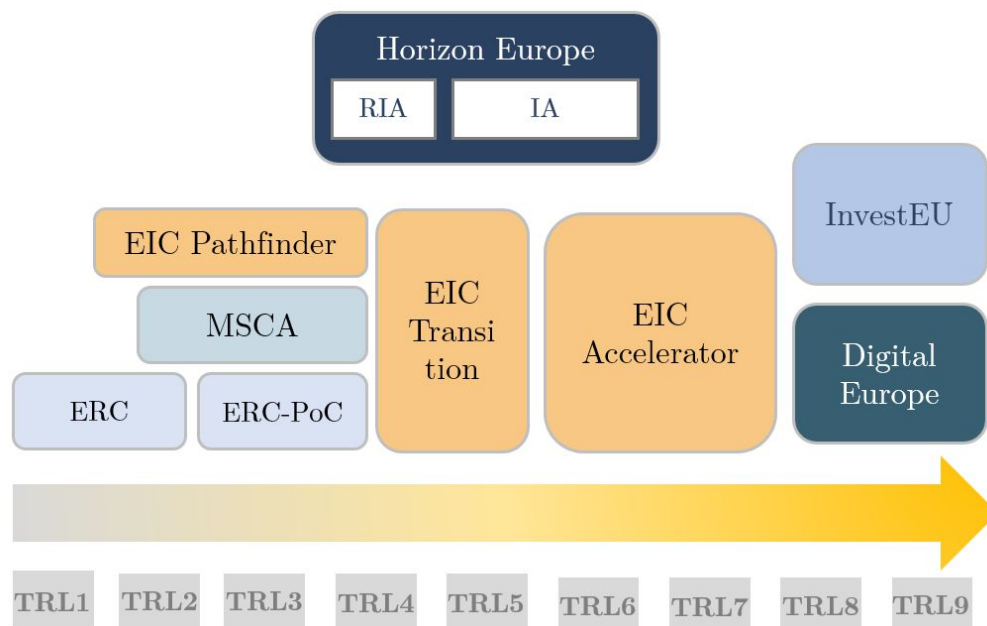
Los motivos de esta paradoja, aunque no formalmente definidos, pueden asociarse a diferentes situaciones. En primer lugar, no existen documentos de referencia consolidados ni grandes guías de éxito para realizar este proceso de transferencia, ya que tradicionalmente se ha tenido la concepción de la separación de las esferas académica y comercial [526]. En segundo lugar, las iniciativas de I+D tienden a abordar la cuantificación del impacto social y comercial como un ejercicio académico separado de aspectos empresariales (tales como encaje, desarrollo de cliente o flujos de ingresos). En tercer lugar, las *start-ups* o empresas industriales suelen incurrir en el error contrario, centrándose en ventas, valor y crecimiento sin tener en cuenta las barreras o riesgos tecnológicos de según qué líneas. Por último, según fuentes consultadas, puede concluirse que desde el punto de vista de la financiación privada, las actividades de transferencia tecnológica tienen asociados (todavía) factores de riesgo elevados, alejando a potenciales inversores que prefieren apostar por objetivos más seguros [6].

Es también en este punto en el que, de cara a una transferencia tecnológica adecuada, debe empezar a plantearse la creación de una empresa para gestionar la propiedad intelectual y explotar los resultados de dicha investigación. Sin embargo, poner en marcha un negocio desde cero es tremendamente complejo y difícil. Por cada historia de éxito (por ejemplo, Google, Facebook, Instagram, Twitter, etc.) hay incontables PYMES que se quedan sin recursos antes de alcanzar el encaje producto/mercado. El emprendimiento de este tipo de iniciativas, fuertemente basadas en proyectos de I+D, suele caer en el error de lanzarse a la comercialización de un producto sin (1) involucrar a suficientes beneficiarios, usuarios o clientes potenciales y (2) hacer hincapié en sus hipótesis iniciales para reforzar/descartar asunciones intermedias (de TRL5-6), refinar el problema o reconcebir el alcance de los MVPs (Producto Mínimo Viable), ya que desde el momento del fin de la

investigación se encuentran ante un gran abismo que separa el mundo académico del empresarial. La falta de apoyo en esta fase, unida a estos aspectos mencionados, hace que este paso sea realmente difícil [527].

Sin embargo, durante los últimos años se está observando una tendencia a atacar este punto en la escala de madurez tecnológica de cara a favorecer esta transferencia. Tanto en la escala nacional como europea, existe un incremento en el interés de apostar por la financiación en estos estadios, permitiendo una mejor transición y una mayor facilidad de explotación de productos provenientes de la investigación.

Como puede observarse en la siguiente figura, la nueva estrategia de financiación publicada por la Comisión Europea [528] incluye nuevos programas que encajan en este punto.



**Figura 6.3:** Instrumentos recientes de financiación de la Unión Europea

- *Digital Europe* [529] es un programa orientado a mejorar las capacidades digitales de la UE, centrado en áreas clave como la IA, la ciberseguridad, computación avanzada e infraestructura de datos en sectores críticos como el medioambiente, la industria manufacturera, la agricultura y la salud. Conviene destacar aquí que este programa ha reservado más de 400 millones de euros para acciones relacionadas con datos y el continuo de computación IoT-edge-cloud [518]. Esto es de crucial importancia para la perspectiva de transferencia de los resultados de esta tesis doctoral.
- *EIC Accelerator* [530]: El antiguo *SME Instrument* ha sido ampliado y mejorado de cara a favorecer esta transferencia tecnológica en dos de los grados más importantes de la misma: TRL6 y TRL7. Se trata de un instrumento ideado casi explícitamente para compañías tecnológicas



(o con clara orientación a la formación de empresas de base tecnológica), con la intención de significar un punto de inflexión en el (hasta ahora limitado) panorama de financiación en este estadio.

- *InvestEU* [531], programa que fomenta la inversión en proyectos y empresas de reciente creación por parte de fondos privados, parcialmente apoyados por la Unión Europea para reducir el riesgo y favorecer la apuesta por productos tecnológicos innovadores provenientes de la investigación.

Por último, cabe destacar que es una buena oportunidad ya que el campo donde se enmarca esta tesis doctoral está entre las principales prioridades para la Unión Europea, nacional (por ejemplo, en las líneas del programa NEOTEC [532]) y a nivel regional (procesos avanzados y productos innovadores del eje de desarrollo TIC en los objetivos RIS3CV de la Comunitat Valenciana [533] , siendo la más importante la establecida por el programa Digital Europe que explicita la voluntad de la UE de alcanzar el 80% del procesamiento de datos en Europa en entornos *IoT-Edge* para el año 2030.

Tras fortalecer la posición para librar el “valle de la muerte”, se llega al objetivo culminante de transferencia tecnológica: la adopción por parte de una empresa consolidada/establecida o creación de *start-ups* o *spin-offs* para la comercialización y mejora del resultado obtenido tras las fases iniciales de investigación y las intermedias de maduración del producto explotable.

## 6.2. Análisis del resultado explotable

Como se ha expuesto en la sección anterior, son cada vez más las oportunidades de financiación y crecimiento de cara a una transferencia tecnológica de un resultado de investigación. El objetivo efectivo de las mismas (en mayor o menor medida) es la creación de un producto explotable a partir de dichos resultados. Es por ello que uno de los aspectos más relevantes en las mismas (por ejemplo, en *EIC Accelerator*, en *InvestEU* y en NEOTEC) sea el establecimiento de estrategias de negocio basadas en el desarrollo de la tecnología proveniente de las líneas de I+D+i (siendo éste el factor diferencial de la empresa de base tecnológica que explotará el producto).

Es por ello que, con vistas a la preparación de estas posibles solicitudes y acciones de emprendimiento, se debe avanzar hacia la definición de dicho producto y de una estrategia de negocio.

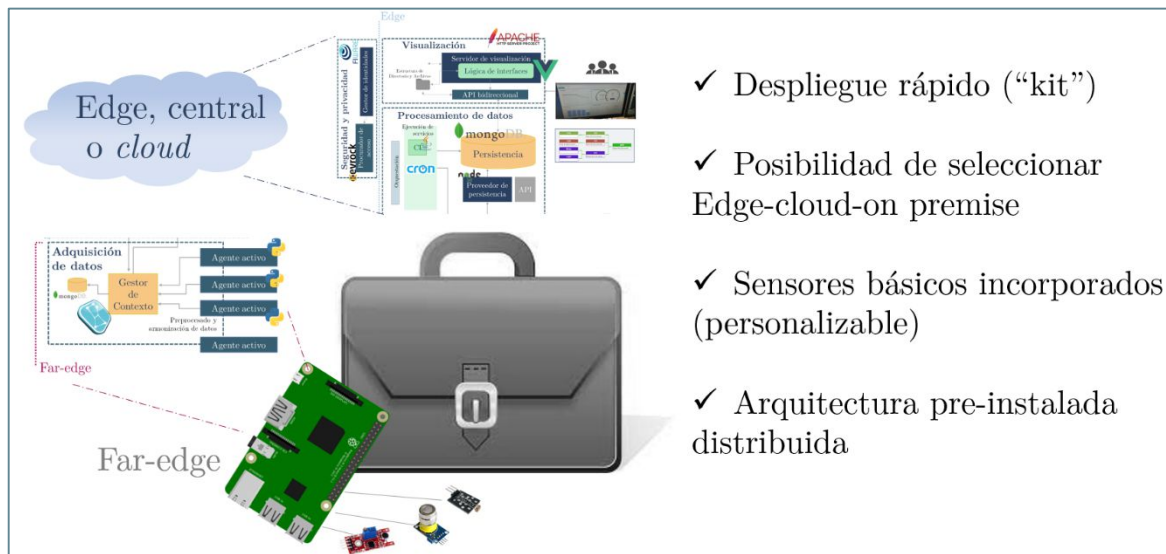
El objetivo de esta subsección es la concepción de una idea de producto a partir de los resultados expuestos en esta tesis doctoral, mientras que la siguiente (sección 6.3) se centra en esbozar dicha estrategia. Ambas secciones se apoyan en herramientas consolidadas para la creación de este tipo de material tales como análisis DAFO [534], [535], PESTEL [536], BMC [537] o 5-fuerzas-de-Porter [538].



*Potencial producto explotable*

A la hora de establecer un esbozo de un producto explotable, se prestó atención, principalmente, a uno de los escenarios de validación expuestos en la sección 5. En particular, el despliegue simplificado  $\mu$ PEI fue considerado una buena opción ya que se trata de una implementación de la arquitectura completa que cuenta con todas las bondades perseguidas en la solución: (i) captura datos en tiempo real de sensores asociados al índice a calcular, (ii) distribuye los módulos de la arquitectura en dispositivos *far-edge* y *edge*, existiendo la posibilidad de ejecutar algún módulo en un entorno *cloud*, (iii) incluye los servicios de cálculo y de predicción de índice compuesto, (iv) es ampliable con nuevo equipamiento y altamente customizable. Además, en el escenario de validación 5.5 se ha probado con hardware de bajo coste, obteniendo buenos resultados.

Por tanto, se esbozó un concepto de explotación de la solución denominado “*CIC kit*” que podría ofrecer claras perspectivas innovadoras, facilitando un despliegue rápido de una solución IoT para la medición y predicción de indicadores compuestos. La idea (como puede verse en la siguiente imagen) sería crear un despliegue “portable” (todo cabe dentro de un maletín) en el que, con la provisión de sensores básicos y tras la definición del índice compuesto necesario, podría probarse en un espacio corto de tiempo y proveer las características de la arquitectura diseñada.



**Figura 6.4:** Esbozo de producto explotable “CIC kit”

Inicialmente, podría equipar un índice compuesto por defecto (por ejemplo, de impacto medioambiental) o podría darse la opción a diseñar uno a medida (posible gracias a la flexibilidad del servicio – sección 4), pero en este caso deberían desarrollarse nuevos agentes, lo que ralentizaría el despliegue. La solución estaría basada en tecnologías libres (*open source*) y se seguirían las metodologías de despliegue y validación planteadas en esta tesis doctoral (sección 5.2).

### Definición de valor y características

Una vez definido el producto sobre el que se centraría la transferencia tecnológica, resultó interesante plantear una serie de cuestiones sobre cuál es su valor y lo que podría diferenciarlo del resto de soluciones disponibles. Pudo considerarse esta actividad como un paso previo de cara a plantear una estrategia de negocio (sección 6.3)

#### **Problema / Ganancia y Valor**

Este producto atacaría el problema de cuantificar de una manera fiable, precisa y rápida un fenómeno real (por ejemplo, impacto medioambiental) de tal manera que pudiera hacerse un seguimiento en el tiempo en escalas fácilmente interpretables. La solución dotaría de un único valor entre 0 y 1, de interpretación directa y con baja necesidad de inversión inicial, calculado en tiempo real y siguiendo unas formalidades (de índice compuesto) validadas por la Unión Europea y replicables en otras sedes, localizaciones, compañías o para otros fenómenos (a medida).

#### **Funcionalidad, tecnología y validación**

El “*CIC kit*” se basa en la adquisición de datos en tiempo real, por tanto requiere una metodología sólida de adquisición e introducción de información en el sistema. Además, son necesarias estrategias para disponer de esa información al instante, a parte de almacenarla en un repositorio histórico. Igualmente, se requiere no solo la obtención de datos sino su utilización para popular el índice compuesto. Todo lo anterior está incluido de manera nativa en la instalación de la arquitectura propuesta, en la que acuerdos y convenciones durante la instalación (hecha previo despliegue) consigue una integración de todos los componentes. Una vez en el lugar de implantación, únicamente unas pocas configuraciones relacionadas con la conectividad y los parámetros de red, así como comprobaciones de funcionamiento serían necesarias. Un usuario podría acceder a través del portal web expuesto por la solución (siempre que estuviera dentro de la misma red local) para observar los resultados del CI. Este despliegue específico ha sido validado durante la tesis, al igual que otros cuatro similares.

#### **Diferenciación**

No existe ningún sistema de IoT específicamente diseñado para el cálculo y predicción de índices compuestos ni orientado a despliegues rápidos y de bajo coste. Soluciones o plataformas de IoT como Google Cloud IoT Core, Azure IoT, AWS IoT, IBM Watson u otras proporcionan soporte de sistemas IoT en el que, tras varias instalaciones, desarrollos a medida, adquisición de sensores, conexión de los mismos, y manipulación de las interfaces de visualización, podría conseguirse la observación de valores de una manera similar a lo propuesto con este producto. Sin embargo, para ello se requiere un gran conocimiento de dichas plataformas (o la contratación de empresas con dicho conocimiento), así como estar supeditado a sus entornos *cloud*, con baja personalización. Este producto ofrece un despliegue ágil y adaptado a las necesidades, con una serie de sensores e índices precargados,

ofreciendo asistencia (consultoría, mantenimiento, nuevos sensores, nuevos índices) de una manera customizada y con la garantía de basarse en desarrollos *open source* provenientes de resultados de proyectos I+D europeos y nacionales. Además, se trata del único sistema IoT que incorpora de forma nativa un servicio de indicadores compuestos alineado con las disposiciones formales para la formulación de los mismos según organismos oficiales de la Unión Europea.

### **Mensaje del producto**

Entidades de cualquier tipo, tamaño y disponibilidad de recursos pueden, con la solución “*CIC kit*”, monitorizar fenómenos complejos en tiempo real a través de una interfaz de interpretación y servicio sencillos y en base a la adquisición de información en tiempo real. Todo ello en poco tiempo y a bajo coste. Léase los requerimientos de la solución y los procesos de instalación y validación para comprobar el ajuste a las características de la entidad y definir de un índice que sea útil para sus necesidades.

### **Otras consideraciones**

El producto “*CIC kit*”, además, permitirá las siguientes funcionalidades que pueden considerarse parte de su valor añadido:

- Capacidad para insertar desarrollos propios de la entidad (por ejemplo, un modelo de IA, un nuevo sistema de almacenamiento, un servidor web) y poder distribuirlo en diferentes puntos del continuo de computación (*cloud, far-edge, edge*), siempre que se sigan las indicaciones de este documento (sección 3.2.1).
- Permitir que entidades/personas externas a la entidad adoptante puedan hacer uso de los servicios/resultados y también acceder a los datos persistentes en el sistema (gracias a las APIs del producto) si así se desea.
- Potenciar la interoperabilidad con las plataformas que operan a varios niveles (dispositivo, red, middleware, aplicación) basándose en las APIs REST y la documentación aportada en este manuscrito.
- Se trata de un ecosistema flexible y escalable que permite la incorporación de nuevas tecnologías siempre que los principios de encapsulación y las capacidades de los equipos lo permitan.
- Se trata de una excelente solución (en abierto, basado en tecnologías consolidadas y en otras personalizadas expuestas en esta tesis) para crear nuevas plataformas de cara a nuevas líneas de investigación dentro de las entidades o a nuevos estudiantes e investigadores.

### **Análisis DAFO**

DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades) es una herramienta que permite analizar la realidad del producto sobre el que orbitar una empresa (en este caso, EBT – Empresa de Base Tecnológica) para analizar si es

viable y reflexionar sobre su idoneidad. Analiza aspectos tanto negativos como positivos y tanto externos como internos, tratando de establecer las bases para una estrategia de negocio posterior que potencie las fortalezas y controle los riesgos y amenazas.

La información contenida en las siguientes tablas ha sido extraída tras la utilización de las herramientas provistas por el portal de instrumentos PYME de la Secretaría General de Industria y de la PYME del Ministerio de Industria, Comercio y Turismo [534]

**Tabla 6.1:** Análisis DAFO del producto explotable

|                   | Aspectos positivos  | Aspectos negativos  |
|-------------------|---|---|
| Factores internos | FORTALEZAS  | DEBILIDADES   |
|                   | <ul style="list-style-type: none"> <li>- Despliegue rápido, ágil y de bajo coste.</li> <li>- Atención personalizada y de proximidad, con soporte durante todas las fases.</li> <li>- Basado en tecnologías <i>open source</i> y en desarrollos a medida documentados.</li> <li>- Los servicios incluidos están fuertemente basados en metodologías validadas científicamente.</li> <li>- Cuantificación e interpretación sencilla basada en datos en tiempo real.</li> <li>- Soporte de origen I+D europeo.</li> </ul>  | <ul style="list-style-type: none"> <li>- Puede requerir cierta formación por parte de los miembros de la empresa adoptante para añadir nuevas fuentes, configurar el servicio, predicción, etc.</li> <li>- No se centra en la monitorización de los datos de los sensores sino en los resultados del servicio de CI.</li> <li>- El producto no es completamente de propósito general de forma nativa. Requiere configuraciones y adaptaciones para establecer los CIs e incorporar nuevos servicios.</li> <li>- Se trata de una tecnología validada únicamente hasta TRL6.</li> </ul> |
| Factores externos | OPORTUNIDADES   | AMENAZAS  |
|                   | <ul style="list-style-type: none"> <li>- Posibilidad de añadir nuevos servicios (predicción, bases de datos replicadas, compartición, comparación, etc.) y tecnologías (gemelo digital, Tactile Internet, redes neuronales, TSN...) del NGIoT por el diseño de la arquitectura.</li> <li>- Los últimos años están significando una explosión en la disponibilidad de sensores y en la miniaturización de componentes de computación, útiles para el producto.</li> <li>- En línea con las prioridades de inversión y de financiación de <i>edge computing</i> y continuo de computación.</li> </ul> | <ul style="list-style-type: none"> <li>- Existe una amplia variedad de plataformas IoT.</li> <li>- Las soluciones de CI digitalizadas están avanzando mucho durante los últimos años y es probable que surjan soluciones similares a la propuesta.</li> <li>- La adopción de IoT es creciente, por tanto, muchas entidades pueden ya disponer de un sistema propio o adquirido y ser reacias a incorporar un nuevo despliegue IoT en la empresa.</li> <li>- Otros productos disponibles disponen de experiencia, cartera de clientes y planes de negocio consolidados.</li> </ul>     |

**Análisis PESTEL**

El análisis PESTEL (Político, Económico, Social, Tecnológico, Ecológico y Legal) -en inglés, PESTLE-, es un método internacionalmente reconocido para evaluar el contexto macroambiental al que se enfrentará un posible producto/empresa [536]. Aunque el resultado de transferencia planteado es de carácter eminentemente tecnológico, es conveniente analizar cómo el resto de factores PESTEL pueden impactar en la consecución de objetivos de dicha transferencia.

**Tabla 6.2:** Análisis PESTEL del producto explotable

|   |  |  |
|---|--|--|
| <p><b>P Políticos</b></p> <ul style="list-style-type: none"> <li>- Grupo K4P JRC apuesta por los CIs durante los últimos años</li> <li>- <i>IoT-edge-cloud</i> es una prioridad temática en investigación e inversión</li> <li>- Iniciativas españolas como el Kit Digital podrían afectar</li> </ul>   | <p><b>E Económicos</b></p> <ul style="list-style-type: none"> <li>- La disponibilidad de ayudas de transferencia tecnológica parece estable.</li> <li>- Se trata de un producto de bajo coste (época de recesión).</li> <li>- La apuesta por PYMES parece clara en Europa y España.</li> </ul>   | <p><b>S Socio-culturales</b></p> <ul style="list-style-type: none"> <li>- Apoyo y apuesta por tecnologías <i>open source</i></li> <li>- Reactancia a la adopción de soluciones que impliquen recogida de datos y/o compartición de ellos.</li> <li>- Mejora del nivel de auditoría por parte de entidades públicas.</li> </ul> |
| <p><b>T Tecnológicos</b></p> <ul style="list-style-type: none"> <li>- El desabastecimiento global de chips y sensores puede afectar.</li> <li>- El advenimiento de nuevas tendencias, tecnologías o soluciones IoT puede condicionar la aparición de productos alternativos.</li> <li>- Librerías/herramientas de terceros usadas podrían quedar obsoletas</li> </ul> | <p><b>E Ecológicos</b></p> <ul style="list-style-type: none"> <li>- El uso de baterías en los dispositivos <i>far-edge</i> y los sensores puede ser un factor en este aspecto.</li> <li>- El producto y el servicio tienen un claro encaje para la monitorización del impacto ambiental (probado en diversos escenarios).</li> <li>- El uso de <i>edge</i> es más favorable que <i>cloud</i>.</li> </ul> | <p><b>L Legales</b></p> <ul style="list-style-type: none"> <li>- La gestión de propiedad intelectual del resultado debe aún debatirse, siempre sujeta a la legalidad vigente.</li> <li>- La solución contempla las disposiciones de la ley GDPR de protección de datos personales.</li> </ul>                                  |

Por supuesto, el contenido incluido en las dos tablas anteriores dista mucho de un ejercicio exhaustivo de elaboración de análisis DAFO y PESTEL. Este tipo de métodos son aplicados por expertos en mercadotecnia y en entornos empresariales. Únicamente se ha propuesto esbozar unas primeras líneas de cara a la presentación de solicitudes de financiación y a la preparación de las siguientes fases de transferencia tecnológica del producto “*CIC kit*”.

## 6.3. Esbozo de un modelo de negocio

Según algunas metodologías de emprendimiento y desarrollo empresarial (por ejemplo, *LEAN methodology*), el resumen del planteamiento de negocio comienza con cubrir los puntos: (1) encaje problema/solución, (2) encaje producto/mercado y (3) encaje de modelo de negocio apropiado[539].

En este sentido, el primer punto ha sido cubierto en la sección anterior, destacando los problemas de representación de fenómenos reales a través de índices compuestos y cómo el producto “*CIC kit*” ofrece una solución a los mismos. Se trata de un resultado de bajo coste de despliegue rápido y aplicación inmediata, basándose en componentes (en su mayor parte) estandarizados *de facto*, lo que garantizaría una correcta tasa de reposición y flexibilidad (Docker, FIWARE, Elasticsearch...). Además, requiere de una baja inversión inicial, por lo que aparte de responder al primer punto, promete ser apropiado para el segundo.

Con respecto a dicho segundo aspecto (el enfoque producto/mercado), este análisis sería objeto de estudio en actividades y ejercicios más exhaustivos, que se encuentran fuera del alcance de esta tesis (lista completa de competidores, beneficios, cuotas de mercado, etc.). Únicamente se ha procedido en este trabajo a la realización de un análisis de las 5 fuerzas de Porter [538]. Se marca como una tarea futura en caso de continuar con el proceso de transferencia tecnológica expuesto.

Por último, el establecimiento de un modelo de negocio apropiado es clave en este aspecto de la transferencia. De hecho, según la literatura de mercadotecnia, se estima que un 90% de las *start-ups* (pueden considerarse como EBTs) no alcanzan los dos años de vida debido a una mala construcción de los modelos de negocio iniciales [3]. Es objeto de esta subsección un primer análisis hacia el esbozo de un modelo de negocio para el producto “*CIC kit*”. Para ello, se plantea la identificación de varios perfiles adoptantes, el establecimiento de un BMC (*Business Model Canvas*), el análisis de fuerzas mencionado (Porter) y, por último, se propone un potencial esquema de comercialización por niveles (*tiers*).

### Establecimiento de perfiles adoptantes

Basado en la experiencia del candidato en los tres proyectos de investigación mencionados (sección 1.4.2), en el potencial de utilización del producto y siguiendo algunas disposiciones expuestas por la metodología *Silicon Valley Product Group* [13], se han establecido dos perfiles iniciales como potenciales adoptantes de la solución:

#### **Empresas privadas**

Este tipo de despliegues tendrían el objetivo de dotar a la empresa de una herramienta para monitorizar (en el tiempo) su impacto sobre un elemento real o la evolución de un fenómeno complejo que pudiera ser capturado con fuentes asimilables al IoT. Este tipo de adopciones podrían estar también motivadas por la

constatación del cumplimiento de ciertas normativas (por ejemplo, relacionadas con el impuesto sobre el carbono Fit-for-55 [540]) que pudieran ser resumidas a un índice compuesto. Por otro lado, esta solución serviría a empresas privadas para observar el efecto de ciertas medidas aplicadas (por ejemplo, analizando el valor del CI antes y después de implantar medidas de teletrabajo). Por otro lado, también podrían ayudar a aportar una imagen de mercado a la empresa; el “*CIC kit*” se basa en metodologías formales de adquisición de datos y de establecimiento y cálculo de índices compuestos, por tanto, si una empresa decidiera publicar sus datos de (por ejemplo) economía circular o responsabilidad social corporativa, avalados por la utilización de este producto, podría utilizarlo como un reclamo de clientes y un sello de calidad. En este punto, es también interesante recordar que el servicio incluido en el resultado propone un índice complementario para informar sobre la fiabilidad tecnológica (RR – sección 4.2.4), lo cual aumentaría igualmente la transparencia de esta comunicación por parte de la empresa adoptante. En este caso, los despliegues serían individuales y, muy probablemente, con altas necesidades “a medida”, descartando las opciones de compartición de datos, normalización o resultados.

#### **Organismos públicos:**

Esta solución podría ser interesante para diferentes organismos públicos por su capacidad para establecer mecanismos de medición, agregación y composición de un índice sobre el que comparar el rendimiento/impacto de varias entidades sobre un fenómeno particular de una forma directa. Esta solución podría ser utilizada como una herramienta comparativa informativa (por ejemplo, Producto Interior Bruto de dos países), como una herramienta de auditoría para el cumplimiento de normativa (por ejemplo, en igualdad de género o en calidad de educación y gobierno [541]). Además, esta solución podría ser utilizada como baremo o prerrequisito para la concesión de licitaciones (por ejemplo, solo pueden financiarse aquellas propuestas cuya entidad supere 0,8 en el índice de sostenibilidad). En estos casos, la solución “*CIC kit*” podría ser instanciada por cada entidad comparable para demostrar el valor resultante de su índice o podría ser adquirida por la entidad pública que sería la encargada de tomar las mediciones, instalar el producto y realizar las inspecciones. Las posibilidades son amplias y variadas. En estos casos, la comparación, la normalización compartida y otros servicios auxiliares como el ranking o el motor de recomendaciones serían de gran utilidad (véase sección 4.2.4).

#### ***BMC Business Model Canvas – Modelo de negocio***

El *Business Model Canvas* (BMC) es una herramienta útil utilizada en el sector empresarial para representar los aspectos más importantes de cara a la explotación de un producto: segmento de mercado, clientes, propuesta de valor, canales, flujos de ingresos, estructura de costes, recursos clave, problema a resolver y socios para conseguirlo [9]. En la siguiente tabla se muestra una primera aproximación a un BMC para el producto “*CIC kit*”. Igual que en la reflexión anterior, esta actividad merecerá trabajo especializado futuro.

**Tabla 6.3:** BMC – Esbozo de modelo de negocio

| Problema y socios clave  | Actividades clave/Solución  | Proposición de valor  | Relaciones con cliente  | Segmentos de clientes   |
|--|---|---|---|---|
| <p><u>Problema</u><br/>Cuantificar en un índice de 0 a 1 impactos o fenómenos reales para facilitar una toma de decisiones a partir de métodos y tecnologías fiables.</p> <p><u>Socios clave:</u><br/>Proveedores de sensores y equipamiento, entidades públicas promotoras y organismos públicos financiadores.</p> | <p>Captura de datos con sensores y otros elementos IoT.</p> <p>Cálculo y predicción basado en procedimientos formales de la UE.</p> <p>Uso de tecnologías open source.</p> <p>Bajo coste.</p> | <p>CIC kit es el único sistema de IoT específicamente diseñado para el cálculo y predicción de índices compuestos ni orientado a despliegues ágiles y de bajo coste.</p> <ul style="list-style-type: none"> <li>- Flexibilidad</li> <li>- Escalabilidad</li> <li>- En tiempo real</li> <li>- Demostración rápida in-situ disponible</li> <li>- Basado en código abierto</li> <li>- Incorporación de nuevas tecnologías</li> </ul> | <p>Asistencia técnica</p> <p>Producto en varios niveles</p> <p>SaaS</p> <p>Cercanía y periodo de prueba</p> <p>Modelo de venta por <i>tiers</i></p> | <p><b>Empresas privadas</b></p> <p>Sector logística, medioambiente, urbano, energía, agricultura...</p> <p>Con el objetivo de monitorizar y simular escenarios complejos a partir de mediciones IoT.</p> <p><b>Organismos Públicos</b></p> <p>Países, aytos, gobiernos regionales, autoridades de regulación medioambiental...con la intención de comparar y establecer rankings.</p> |
| <p><b>Recursos clave</b></p> <p>Personal de Desarrollo y mantenimiento del producto (UPV SATRD)</p> <p>Equipamiento de computación y sensores.</p> <p>Movilidad y conexión de red.</p>   |   | <p><b>Canales</b></p> <p>Página web</p> <p>Ferias y eventos de tecnología</p> <p>Redes sociales, <i>Whitepapers</i>, Publicaciones científicas GitHub</p>   |   |   |
| <p><b>Estructura de costes</b></p> <p>Equipamiento de base (sensores, <i>far-edge</i>, <i>edge</i> y central – <i>on-premise</i>).</p> <p>Espacio <i>cloud</i> si fuera necesario.</p> <p>Mantenimiento del software y corrección de errores, soporte, marketing y ventas</p>  |   | <p><b>Canales de ingresos</b></p> <p>Instalación de la solución</p> <p>Desarrollo de servicios sobre la arquitectura</p> <p>Mantenimiento del producto</p> <p>Inclusión de nuevas tecnologías</p> <p>Adición de nuevas fuentes de datos</p> <p>Establecimiento y puesta en marcha de nuevos índices compuestos sobre el producto</p>  |   |   |



### *Análisis del modelo de 5 fuerzas de Porter*

El análisis de competidores y mercado es un trabajo gigantesco que no se aborda aquí, pero se ha decidió incluir el análisis de 5 fuerzas según el modelo de Porter [542]. Los siguientes párrafos están basados en la consulta de fuentes [538], [543] y en la participación del candidato en los proyectos de investigación PIXEL y ASSIST-IoT, donde ha tenido la oportunidad de participar en actividades relacionadas con el programa *Horizon Results Booster* de la Comisión Europea [544]. Como su nombre indica, esta actividad consiste en analizar cinco fuerzas relacionadas con la competencia en un mercado, la sostenibilidad y el rendimiento, especialmente útil en mercados cambiantes y dinámicos como el de NGIoT (y las nuevas tecnologías en general).

A continuación se expone un breve análisis de estas cinco fuerzas en relación al producto “*CIC kit*”. De nuevo, esta actividad es solo el comienzo que podría aprovecharse en caso de continuar con esta línea de transferencia tecnológica.

#### **1. Rivalidad entre competidores existentes**

Trata de expresar el número y la competitividad entre las empresas que actualmente ofrecen productos similares al planteado. En este sentido, un producto entrante debe buscar una diferenciación en servicio o en precio.

- El mercado IoT está altamente copado por empresas que facturan miles de millones de euros (Amazon, Microsoft -Azure, Google, IBM o SAP). No obstante estas empresas no basan su oferta en servicios especializados en el dominio de los CIs y raras veces se ciñen a formalizaciones o recomendaciones de la Unión Europea en términos de servicio.
- La transformación digital está abriendo las puertas del IoT a pequeñas y medianas empresas, que pueden diferenciarse de los grandes competidores en cercanía, rapidez de servicio, mantenimiento y personalización, pudiendo plantear estructuras de costes razonables.

#### **2. Amenaza de entrada de nuevos competidores**

Esta fuerza modela la capacidad de nuevas empresas de entrar en un mercado; es decir, la facilidad para superar las barreras de entrada al mismo. Cuanto más altas sean, más fácil será mantener una posición consolidada, no obstante también será difícil acceder si se trata de una nueva empresa. Desde el punto de vista del “*CIC kit*”, el mercado sería el de soluciones IoT especializadas para aportar valor añadido en la toma de decisiones.

- No existe una gran lealtad a las soluciones ya existentes.
- El coste de inversión en infraestructura suele ser elevado.
- El grado de especialización del personal técnico es muy elevado.
- Analizando los segmentos de clientes, el canal de acceso a los mismos parece relativamente complejo si no se parte de una base inicial.

### 3. Poder de negociación de los proveedores

La siguiente fuerza de Porter versa sobre el nivel de dependencia de proveedores. Cuanto más alto sea, más difícil será mantener una buena posición en dicho mercado. Con respecto al “*CIC kit*”:

- No existe un monopolio de proveedores de sensores IoT.
- Existe una gran variedad de proveedores de equipamiento de computación (*edge* y *central on-premise*).
- La utilización de licencias *open source* en el software de terceros reduce la dependencia de proveedores de herramientas necesarias para el funcionamiento de la solución.
- No parece que los proveedores de sensores ni equipamiento deseen copar el mercado de los servicios especializados, sino más bien servir como habilitadores de los mismos.
- Los precios están experimentando gracias subidas debido a la inflación global y al desabastecimiento de productos electrónicos.
- La dependencia con proveedores *cloud* puede ser reducida abogando por despliegues locales (factible y recomendado por la arquitectura del producto).

### 4. Poder de negociación de los clientes

Se refiere a la habilidad de los clientes para conseguir una merma en el precio y un mayor poder de negociación sobre el producto. Este hecho es frecuente en mercados donde existe una gran variedad de oferta relativamente similar. En este caso, el “*CIC kit*” será un producto diferenciado, pero esta fuerza debe ser analizada junto a los siguientes factores:

- Las grandes empresas y organismos públicos tendrán capacidad de negociar ya que podrán optar a una mayor elección debido a los recursos disponibles.
- Las soluciones similares a “*CIC kit*” son, normalmente, específicas por sector, por tanto deberá observarse con atención si existen productos siendo usados en un nicho en concreto antes de acceder.
- Las características de la arquitectura propuesta son, en general, diferentes a las encontradas en soluciones IoT en las empresas.

### 5. Amenaza de productos y servicios sustitutivos

La última fuerza modela aquellas situaciones en que un producto puede ser sustituido por otro y, de ese modo, impedir su crecimiento o establecimiento en el mercado. En el caso, del “*CIC kit*”, debería analizarse si existen formas alternativas de cuantificar un impacto o fenómeno real a través de un índice compuesto. El objetivo de este servicio es la toma de decisiones de una forma rápidamente interpretable.

- No existen compañías especializadas en una solución como esta, basada en adquisición de datos IoT en tiempo real.
- Existen soluciones que permiten capturar datos y obtener *dashboards* de la información en tiempo real de forma ágil. No incluyen el servicio de CIs, pero podrían ser un sustituto bajo ciertas condiciones.
- Si el servicio de soporte, instalación, mantenimiento y continuación es satisfactorio, el riesgo de los productos sustitutivos se reduce.
- Las iniciativas de gemelos digitales y modelado de fenómenos reales en el *edge* podrían suponer un riesgo de sustitución del producto.

### Esquema de comercialización

Se plantea un esquema de comercialización en el que se ofrecería una solución que pudiera adquirirse por capas (*tiers*). Tal y como se observa en las tendencias actuales de (principalmente) las plataformas *cloud* (por ejemplo, Atlassian, Amazon u Owncloud, por mencionar algunas) [545], las ofertas por niveles establecen modalidades *premium* o avanzadas en función de: a) mayor número de usuarios, volumen de computación o características técnicas o b) de funcionalidades o servicios disponibles.

En relación al producto “*CIC kit*”, las opciones son realmente variadas (al tratarse de una solución muy flexible), por tanto, se requirió un ejercicio de análisis inicial listando los aspectos a tener en cuenta para establecer dichos niveles:

- **El servicio de CI:** Puede incluirse por defecto una configuración para algún índice compuesto prediseñado (por ejemplo, alguno de los desarrollados en los escenarios de validación como el PEI). La herramienta permite la creación de nuevos índices (selección de métodos matemáticos, establecimiento de estructura del árbol, etc.). Estas configuraciones, si se desea establecer algún otro CI por defecto, podrían incluirse, así como los componentes necesarios para ello.
- **Sensores:** Inicialmente, puede incluirse un juego de sensores como el expuesto en 5.5., donde datos como la temperatura, el sonido, la iluminación y la contaminación estarían disponibles inicialmente para alimentar el índice compuesto. No obstante, podrían incorporarse nuevos sensores si se quisiera, con los correspondientes cambios y adiciones software y hardware.
- **Agentes:** Inicialmente, el equipamiento *far-edge* incluido en el maletín (típicamente, una Raspberry Pi) incorporaría los conectores de datos asociados a los sensores indicados en el punto anterior. Éstos podrían aumentarse conforme aumentarían dichos dispositivos.
- **Servicios auxiliares adicionales:** Servicio de predicción, de índice de fiabilidad asociado, recomendaciones...

- **Topología de computación:** (i) Podría incluirse únicamente un elemento de computación en el maletín (*far-edge*) y delegar el resto de la ejecución de los módulos de la arquitectura a un entorno *cloud* o a un entorno central *on-premise* o (ii) podría incluirse equipamiento *edge* adicional en el mismo producto, de tal manera que se lograra un despliegue local completo.

|                                     | Basic Cloud  | Cloud  | Premium Cloud  | On-premise   | Premium Local  |
|-------------------------------------|--|--|--|--|--|
| Precio                              | €  | € €  | € € €  | € €  | € € € €  |
| Servicio de índices compuestos      | PEI por defecto<br>Flexibilidad para crear nuevos                | PEI por defecto<br>Flexibilidad para crear nuevos                        | PEI y OAI por defecto<br>Flexibilidad para crear nuevos  | PEI por defecto<br>Flexibilidad para crear nuevos  | PEI y TCI por defecto<br>Flexibilidad para crear nuevos  |
| Sensores y agentes                  | CO2, iluminación, ruido y temperatura.                           | CO2, iluminación, ruido y temperatura<br>2 sensores adicionales a elegir | CO2, iluminación, ruido y temperatura<br>5 sensores adicionales a elegir                                     | CO2, iluminación, ruido y temperatura<br>2 sensores adicionales a elegir                             | CO2, iluminación, ruido y temperatura<br>6 sensores adicionales a elegir   |
| Servicios adicionales               | Índice de fiabilidad   | Índice de fiabilidad<br>Predicción de índices                            | Índice de fiabilidad<br>Predicción de índices<br>Motor de reglas<br>Recomendaciones<br>Normalización externa | Índice de fiabilidad<br>Predicción de índices<br>Motor de reglas                                     | Índice de fiabilidad<br>Predicción de índices<br>Motor de reglas<br>Recomendaciones<br>Normalización externa<br>Aprendizaje federado |
| Detalles de topología computacional | <b>En maletín:</b><br>Equipo far-edge<br><b>Cloud:</b><br>Básica | <b>En maletín:</b><br>Equipo far-edge<br><b>Cloud:</b><br>Intermedia     | <b>En maletín:</b><br>Equipo far-edge<br><b>Cloud:</b><br>Avanzada   | <b>En maletín:</b><br>Equipo far-edge<br>Equipo edge<br>Soporte en servidor on premise de la entidad | <b>En maletín:</b><br>Equipo far-edge<br>Equipo edge<br>Soporte en servidor on premise propio  |

Figura 6.5: Esquema tentativo de comercialización por *tiers*

A partir del esquema expuesto, una estrategia de comercialización se completaría con las siguientes adiciones:

- Instalación gratuita durante una semana de prueba. En este punto se generaría confianza y necesidad por parte del adoptante.
- Fuera de la oferta inicial, pero dentro de la estrategia de fuentes de ingresos, se podrían realizar desarrollos de servicios a medida (arquitectura flexible) e incorporación de nuevas tecnologías.
- Además, la entidad (EBT) que llevara a cabo esta estrategia de comercialización podría proponer acciones de consultoría para la mejora de la solución, mantenimiento, gestiones de espacio *cloud*, etc.
- Por último, estos paquetes podrían incluir un mayor o menor nivel de formación al personal de la entidad para la utilización y aprovechamiento (desarrollo) del producto.

# Capítulo 7

## Conclusiones

Esta tesis ha explorado las arquitecturas de referencia que existen a la hora de diseñar un sistema aplicado construido sobre los conceptos técnicos del IoT. Además, ha indagado sobre el campo de índices compuestos, definiendo una aproximación práctica para el establecimiento de indicadores alimentados de datos en tiempo real. En este sentido, la conjunción de ambas líneas resume la innovación planteada. Las actividades de investigación han redundado en el diseño y desarrollo de una arquitectura IoT especialmente preparada para ejecutar servicios de CI y asociados. Además, se ha acompañado de una metodología que constituye una suerte de guía para aplicar los conocimientos y tecnologías adquiridas. Se ha aprovechado lo anterior para validar todo el sistema en cinco casos de uso (en diferentes grados de madurez) y se ha reflexionado sobre su consideración como un posible resultado explotable más allá del ámbito académico.

Durante los últimos tres años se han perseguido los objetivos establecidos en el plan de investigación de la tesis doctoral. Esta sección repasa el grado de cumplimiento de dichos objetivos, así como expone las conclusiones obtenidas en cada uno de los capítulos. Busca, además, condensar aquellos hallazgos de interés sobre la materia y proponer posibles trabajos y líneas de investigación futuras.

## 7.1. Grado de cumplimiento de objetivos

En las siguientes tablas se hace un repaso del cumplimiento de los objetivos establecidos en el plan de investigación (2019). Se incluye la cobertura por capítulo de los mismos, así como una descripción sucinta del alcance de cada uno.

**Tabla 7.1:** Grado de cumplimiento del objetivo 1

|  |   |
|--|---|
| Objetivo 1                             | Analizar las diferentes arquitecturas IoT preexistentes, sus diferentes componentes, opciones de integración y tendencias tecnológicas para los próximos años.  |
| Capítulo                               | Alcance obtenido  |
| 2.2. Arquitecturas NGIoT de referencia | Se ha realizado una diferenciación entre plataformas IoT y arquitecturas para el desarrollo de soluciones IoT. Dentro de este segundo, se han estudiado las principales referencias: RAMI, IIRA, IoT-A, AIOTI HLA, OpenFog, ASSIST-IoT, LSP IoT 3D, RAMEC y FAR-EDGE, así como... |
| 3.2.1. Principios de diseño            | ... los diferentes patrones y tipos de arquitectura, y su uso actual y futuro: monolíticas, SOA, microservicios, comparando entre ellas y escogiendo la más oportuna dados los requerimientos de esta tesis.  |

**Tabla 7.2:** Grado de cumplimiento del objetivo 2

|  |  |
|--|--|
| Objetivo 2   | Conocer el estado del arte respecto a la adopción de índices compuestos en aplicaciones multi-dominio, especialmente centradas en la interacción puerto-ciudad   |
| Capítulo   | Alcance obtenido   |
| 4. Cálculo y predicción de índices compuestos<br>4.1. Estado de la técnica | Se ha realizado un exhaustivo análisis de los diferentes índices utilizados en diferentes áreas (SPI, PEI, ESI, TAI...) y cómo están siendo fomentados por entidades oficiales (K4P-CE), exponiendo sus finalidades, ventajas y limitaciones. Además, se ha documentado un claro análisis sobre las herramientas disponibles (Compind, CIAO, CIM, CIF, COIN) y sus carencias, principalmente centradas en la adquisición de datos y en su utilización en el contexto mencionado. Este análisis sirvió como punto de partida para el desarrollo global del sistema. |

**Tabla 7.3:** Grado de cumplimiento del objetivo 3

|                            |  |
|----------------------------|--|
| Objetivo 3                 | Estudiar modelos de caracterización de nivel de “precisión tecnológica” de un índice compuesto y descubrir el más apropiado para el uso en el sistema a diseñar. |
| Capítulo                   | Alcance obtenido   |
| 4. Cálculo y predicción de | Se ha realizado un repaso a los tipos de fuentes que   |

|  |   |
|--|---|
| índices compuestos<br>4.2.4. Servicios auxiliares    | pueden existir en dichos entornos, proponiendo un indicador compuesto secundario (de “fiabilidad”) basado en las opiniones informadas de expertos en la materia a través del método de asignación de presupuesto (BAM). |
| Apéndice B – Características de las fuentes de datos |   |

**Tabla 7.4:** Grado de cumplimiento del objetivo 4

|                                  |   |
|----------------------------------|---|
| Objetivo 4                       | Obtener un conocimiento profundo sobre las posibilidades de computación ubicua en un entorno aplicado de Internet de las Cosas.   |
| Capítulo                         | Alcance obtenido  |
| 2.3.Edge computing en NGIoT      | Se ha realizado un amplio estudio sobre las posibilidades del <i>edge computing</i> . Se han analizado los tipos de EC (MEC, cloudlets, <i>fog computing</i> ), siendo FC el más apropiado para el ámbito de la tesis. Se han analizado las arquitecturas de referencia (destacando OpenFog) y posibilidades de instanciación de nodos <i>fog</i> (Fledge, INTER-IoT, fog05, FogAtlas, FORA...), materialización esperada en los despliegues. |
| 5. Aplicación en entornos reales | El conocimiento adquirido ha permitido diseñar una arquitectura flexible que ha adoptado varias aproximaciones y topologías de EC en los escenarios de validación.  |

**Tabla 7.5:** Grado de cumplimiento del objetivo 5

|   |  |
|---|--|
| Objetivo 5  | Aplicar las más avanzadas técnicas de ingeniería del software para el diseño de sistemas complejos distribuidos.   |
| Capítulo  | Alcance obtenido   |
| 2.5.3. DevSecOps                                  | Se ha realizado una investigación sobre las técnicas actuales de CI/CD, especialmente observando la integración y despliegue DevOps, y su principal innovación durante los últimos años (explorada por el candidato): DevSecOps.         |
| 3.1.2. Metodología<br>3.2.1. Principios de diseño | Se han aplicado los principios y técnicas de DevSecOps al desarrollo del software, así como se han incluido los conceptos de encapsulación en microservicios (usando Docker) y el despliegue de MVPs para un testeo ágil de la solución. |

**Tabla 7.6:** Grado de cumplimiento del objetivo 6

|            |   |
|------------|---|
| Objetivo 6 | Diseñar un sistema software modular que cubra el objetivo general del proyecto acorde a unos requisitos definidos por el doctorando según experiencias en proyectos de investigación. |
| Capítulo   | Alcance obtenido  |

|   |   |
|---|---|
| 3. Definición e implementación de la arquitectura | Se han definido requisitos técnicos, funcionales y de servicio mediante la metodología Volere. Se ha especificado la metodología de especificación siguiendo las indicaciones DevSecOps y los conceptos más modernos de diseño de sistemas software   |
| 3.7. Solución integrada                           | Se ha creado una arquitectura compuesta de varios módulos (adquisición de datos, procesamiento, ejecución de servicios, visualización y seguridad y privacidad) con funcionalidades claramente diferenciadas. Dentro de los mismos, algunos componentes se consideran esenciales y otros complementarios. |

**Tabla 7.7:** Grado de cumplimiento del objetivo 7

|  |   |
|--|---|
| <b>Objetivo 7</b>  | Desarrollar una herramienta útil y usable de cálculo automático de índices compuestos   |
| <b>Capítulo</b>  | <b>Alcance obtenido</b>   |
| 4. Cálculo y predicción de índices compuestos<br>4.2.2. Cálculo de indicadores compuestos<br>4.2.3. Predicción de índices<br>4.2.4. Servicios auxiliares | Se ha desarrollado una herramienta ligera (lenguaje Java) encapsulada como un servicio (Docker) desplegable en la arquitectura propuesta. Dicho servicio posee flexibilidad a la hora de integrarse en diferentes entornos, y contiene lógica suficiente para realizar un cálculo robusto de índices compuestos siguiendo las especificaciones formales de K4P-CE. Se basa en el cálculo iterativo de una estructura jerárquica en árbol sobre la que pueden añadirse nodos, establecer relaciones entre los indicadores, indicar los métodos matemáticos (de normalización, ponderación y agregación) deseados, observar los resultados y obtener informes automatizados todo ello a través de una interfaz intuitiva. |

**Tabla 7.8:** Grado de cumplimiento del objetivo 8

|  |   |
|--|---|
| <b>Objetivo 8</b>  | Desarrollar una metodología de diseño, implementación, configuración y despliegue de dicho sistema para su uso en entornos reales.  |
| <b>Capítulo</b>  | <b>Alcance obtenido</b>   |
| 5.2. Metodología de aplicación y despliegue de la solución | En el estado del arte no existe una guía combinada para el diseño e implementación de índices compuestos de forma automatizada. Existen recomendaciones y manuales para la definición de CIs, y para la selección de métodos, y por otro lado existen multitud de plataformas y sistemas IoT. En esta tesis se propone una unión de ambos, involucrando activamente actores relacionados con la interpretación de los CIs para la toma de decisiones y a miembros técnicos conocedores del sistema IoT. |



**Tabla 7.9:** Grado de cumplimiento del objetivo 9

| Objetivo 9                     | Implementar todos los componentes necesarios para la implementación de dicho sistema.   |
|--------------------------------|---|
| Capítulo                       | Alcance obtenido  |
| 3.3.<br>Adquisición de datos   | Se han analizado las opciones de adquisición de datos y se ha propuesto el concepto de agentes (activos o pasivos) para introducir datos <i>batch</i> y en tiempo real (recomendando <i>pyngsi</i> ); estableciendo a su vez la necesidad de un concentrador de datos (gestor de contexto), donde la oferta es variada (Kafka, RabbitMQ) y se ha escogido ORION de FIWARE. Se ha especificado también el modelo de datos a utilizar, principal elemento de interoperabilidad introducido en la tesis. |
| 3.4.<br>Procesamiento de datos | Se han analizado las posibilidades de persistencia a largo plazo en el sistema considerando el resto de decisiones de arquitectura. Tras decidir optar por bases no relacionales, muchas opciones estaban disponibles (InfluxDB, Cassandra, pila ELK...). Decidió crearse dos alternativas: Elasticsearch en caso general y MongoDB en caso en que desee realizarse persistencia en entorno <i>edge computing</i> del continuo.   |
| 4.2.2. Cálculo de CIs          | Se desarrolló un software de cálculo y predicción del índice compuesto acorde al objetivo 7.  |
| 3.5.<br>Visualización          | Se realizó un extenso análisis sobre las opciones de visualización (Dashboard) para despliegues IoT, decidiendo optar por un diseño a medida siguiendo <i>frameworks</i> facilitadores como Vue.js o Element.js.  |

**Tabla 7.10:** Grado de cumplimiento del objetivo 10

| Objetivo 10     | ... para evaluar la solución implementada tras su despliegue en, al menos, tres casos de uso, principalmente relacionados con la interfaz ciudad puerto.     |
|-----------------|--|
| Capítulo        | Alcance obtenido   |
| 5.3 – TCI       | Monitorización y predicción de la ocupación de las puertas de acceso al puerto de Tesalónica, en un despliegue mayoritariamente centralizado.                |
| 5.4 - PEI       | Evaluación retrospectiva del fenómeno complejo de impacto medioambiental de actividades portuarias, en un despliegue mayoritariamente centralizado.          |
| 5.5 - $\mu$ PEI | Simplificación del anterior, en un despliegue <i>edge computing</i> ligero y altamente sensorizado. Aproximación al resultado explotable “ <i>CIC kit</i> ”. |
| 5.6 – CEDI      | Predicción de situaciones de incumplimiento de distancia social COVID-19 en terminal portuaria. Despliegue mayoritariamente centralizado.                    |
| 5.7 - OAI       | Recomendación de asignación dupla conductor/camión en transporte por carretera en un despliegue híbrido.   |

## 7.2. Conclusiones globales

En las siguientes páginas se exponen las principales conclusiones obtenidas en cada capítulo elaborado en esta tesis doctoral. No se trata de un resumen de ejecución de las acciones acometidas, sino de un repaso de los hallazgos (y propuesta) de interés que han ido siendo descubiertos por el candidato. El objetivo es generar una lista de aspectos relevantes para una consulta rápida por parte de otros investigadores o cualquier persona interesada en la materia.

### *Estado del arte*

- Existe una gran cantidad de plataformas IoT, cuyo objetivo (que no debe confundirse con el de una arquitectura IoT) es el de controlar un despliegue sensorizado, incluyendo funcionalidades como el registro y descubrimiento de dispositivos, monitorización del estado del despliegue, etc.
- De entre las arquitecturas de referencia para la materialización de despliegues industriales IoT, las más destacadas son RAMI, AIOTI HLA y LSP IoT 3D.
- Existen varios patrones de diseño de arquitecturas IoT, habiendo predominado hasta ahora las aproximaciones monolíticas. Éstas poseen desventajas como una baja escalabilidad o flexibilidad, que no se adaptan a las necesidades del NGIoT, siendo preferibles hacia el futuro enfoques de microservicios y orientados a servicios (SOA).
- Situar la capacidad de computación más cerca de la fuente de datos (a lo largo del denominado “continuo de computación”) aporta grandes ventajas como una menor latencia, optimización del ancho de banda o seguridad entre otras. Existe una clara tendencia a la adopción de principios de CC (*cloud-native*) en estos entornos, pero se trata de una línea todavía abierta.
- Existen tres tipos principales de *edge computing* desde su advenimiento en 2012 (MEC, cloudlets y *fog computing*). Con respecto a este último, la arquitectura de referencia es OpenFog, con nuevas instanciaciones y propuestas siendo registradas cada año, en un campo extremadamente activo y cambiante.
- Los datos de series temporales a menudo tienen un volumen muy elevado, especialmente en escenarios de IoT. Almacenar, indexar, consultar, analizar y visualizar datos de series temporales puede suponer un desafío. Además, a nivel práctico, cada caso puede variar y encontrar la combinación correcta entre capacidades técnicas y ajuste a un escenario, al tiempo que se minimiza el tiempo de despliegue e integración con una arquitectura IoT, es un aspecto que no se encuentra cerrado a día de hoy en la literatura.

- Existen muchos métodos para la predicción de series temporales (ARIMA, LSTM-RNN, SARIMAX, SRM, TBATS...)-algunos de ellos aplicados con éxito en casos similares al entorno de esta tesis- pero cada uno tiene sus debilidades y fortalezas, normalmente relacionadas con la correlación de los datos, ventanas de predicción, entrenamiento requerido, precisión obtenida o consideración de estacionalidades.
- Según AIOTI y otras entidades implicadas en el NGIoT, el aprendizaje federado (*Federated Learning*) está llamado a ejercer un rol clave en las arquitecturas de Internet de las Cosas de nueva generación. Este concepto permite que, en entornos altamente distribuidos (*edge computing*), los datos permanezcan cerca de su origen mientras que pueda compartirse un modelo de aprendizaje que sirva para “aprovechar” la inteligencia adquirida por un nodo en otro, dentro el mismo despliegue. No obstante, la gestión de este aprendizaje (compartición, acuerdos, comunicación, orquestación) en entornos *fog computing* es aún una línea abierta de investigación.
- La virtualización se antoja un elemento clave para la nueva generación de despliegues de IoT al pensar en la consecución de escalabilidad en los mismos. En concreto la tendencia es plantear despliegues modulares basados en contenedores. En un ejercicio adicional de escalabilidad, estos contenedores pueden orquestarse, gestionando su ciclo de vida o replicándolos en agrupaciones de nodos, siendo Kubernetes la tecnología principal para conseguirlo. Sus variantes más ligeras para entornos *edge computing* (k3s, Akri, microk8s...) aún son inestables y componen un campo abierto de investigación.
- Todos los despliegues de IoT que deseen ser instanciados en entornos (pre- y) operativos deben incluir mecanismos de seguridad. Tras un análisis del estado del arte, se concluye que existen soluciones estandarizadas que pueden aplicarse en casi cualquier entorno.
- La metodología DevOps establece un *pipeline* de acciones centradas en la producción, empaquetado y entrega de código que facilita la automatización de las tareas, el trabajo en equipo (por ejemplo, utilizando Git) y, en general, el proceso de producción de software. DevSecOps es una evolución natural que se centra en incluir mecanismos de seguridad desde fases tempranas de dicha producción.
- El Tactile Internet es una de las tecnologías de más reciente aparición que deberá ser considerada por los despliegues de NGIoT futuros. Requiriendo unas estrictas condiciones de infraestructura (red, equipos, códecs) y de interacción con el usuario (objetivos de latencia háptica de 1ms y caída máxima de 3,17 segundos/año), aún son pocas las propuestas realistas para la implementación de este paradigma, clave para la investigación durante los próximos años.

### *Definición e implementación de la arquitectura*

- La licitación de requerimientos y la selección de herramientas y metodologías para generarlos/monitorizarlos son clave para el éxito de cualquier diseño de sistema software, así como la documentación y el versionado.
- En aras de la flexibilidad y escalabilidad en despliegues IoT, es preferible escoger arquitecturas basadas en una estructura modular, así como buscar una independencia a las tecnologías y lenguajes de desarrollo utilizados en cada uno de dichos módulos. Además, el estudio de las características de los paradigmas CC y EC puede hacer que este diseño deba decantarse por uno, por otro o por una combinación de ambos (como en el caso de esta tesis).
- Existe otra decisión a la hora de diseñar una arquitectura que es igualmente relevante: multi-tenancia o multi-instancia. Mientras que la primera sirve a varios clientes desde un mismo entorno de computación (y, sobre todo, de gestión de datos), la segunda separa los dominios de una manera diferenciada. Ambas tienen ventajas e inconvenientes que deben analizarse con cautela.
- La conectividad de datos hacia una plataforma (o sistema, en este caso) IoT es un terreno ampliamente explorado. Existen multitud de soluciones como IBM Connector, Azure, Apache Edgent o IoT-Agent de FIWARE que tienen como finalidad recoger datos e integrarlos en un entorno de Internet de las Cosas. En esta tesis se concluye que suelen ser, sin embargo, componentes altamente especializados y muy orientados a la plataforma que luego va a procesar dicho dato, existiendo relativamente baja oferta de conectores genéricos consolidados. En este sentido, se ha propuesto la distinción entre “agentes” pasivos (implementan un *broker* que recibe actualizaciones de fuentes de datos emisoras) o activos (van a buscar el dato mediante peticiones HTTP, a bases de datos, a documentos, etc.).
- No existe un modelo de datos establecido para el intercambio de información en un sistema como el propuesto. Si bien es cierto que las tendencias a la creación de ontologías en entornos de Smart City y transporte son crecientes (eFTI, SAREF, Lov4IoT), se puede encontrar cabida en las especificaciones flexibles NGSIv2 para cualquier aplicación.
- La inserción de un gestor de mensajería (Kafka, RabbitMQ, ActiveMQ) se antoja clave para aquellos sistemas que trabajen con datos IoT. Dentro de estos, existe una subespecialización (gestores de contexto) que mejora algunas carencias de los *brokers* de mensajes de propósito general (prestan atención a las informaciones contextuales de los datos que envían: qué entidades, de qué tipo, los cambios ocurridos, etc.).

- Desde el punto de vista de persistencia en los despliegues NGIoT, donde los datos cada vez serán más ricos y cambiantes, a la vez que incluirán relaciones complejas entre diferentes entidades y, sobretodo, aumentarán su tamaño requiriendo escalabilidad horizontal y vertical, las bases de datos preferibles son las no relacionales. Dentro de éstas, planteando un sistema (como el estudiado) donde los datos consistirán en series temporales numéricas o en archivos expresados en JSON (o elementos binarios), esta tesis recomienda una valoración pormenorizada entre las opciones InfluxDB, Cassandra, MongoDB, Elasticsearch y Prometheus.
- A la hora de ejecutar procesamiento (aportando valor a través de un servicio) sobre los datos almacenados, existen varias opciones. Para un caso general, esta tesis concluye que existen herramientas de software abierto que gestionan la creación, destrucción y ciclo de vida de aplicaciones encapsuladas (contenedores Docker) para controlar este proceso (Kubernetes, Docker Swarm o Apache Mesos). Si el sistema se plantea como una interconexión entre servicios activos y desea controlarse el acceso, observar soluciones de API Broker (Kong, KrakenD) son también opciones válidas.
- En lo que concierne a la visualización de datos, se trata de uno de los campos más amplios y con más opciones, ya que muchas tecnologías *cloud* o de propósito general pueden aplicarse en despliegues IoT. Esta tesis doctoral concluye que de entre las comerciales (IBM Watson, IBM Sphere, AWS IoT...) y las de software abierto (ThingsBoard, ThingWorx), las segundas son preferibles. No obstante, este tipo de soluciones van de la mano de una plataforma subyacente, implicando problemas de interoperabilidad y esfuerzos extra de adaptación. En solución a esto, optar por *dashboards* genéricos que representen la información contenida en una BBDD o recibida en tiempo real, parecen ser la respuesta. Grafana, Freeboard o Kibana son las herramientas recomendadas por el candidato, a pesar de que, como se ha evidenciado en esta tesis, aproximaciones “a medida” suelen ser una respuesta factible en la mayoría de los casos si desea obtenerse un sistema orientado a alguna aplicación en concreto.
- El diseño de un sistema de gestión basado en políticas parece ser una respuesta eficaz a los requerimientos de seguridad y privacidad de los despliegues IoT cuyo principal foco no sea la protección de los datos o la ciberseguridad.

### ***Cálculo y predicción de índices compuestos***

- Los indicadores compuestos tienen como objetivo proporcionar una visión clara y fácil de interpretar (y lo más ajustada posible) sobre un aspecto de la realidad.

- Este tipo de herramientas son consideradas útiles para ayudar en la elaboración de políticas o en la toma de decisiones, aportando un valor único sobre el que se puede actuar, relajando la cantidad de recursos y tiempo que requeriría analizar grandes cantidades de datos estadísticos. No obstante, también presentan inconvenientes como ofrecer conclusiones simplistas si no se construye adecuadamente, si no se alimenta con suficientes datos, puede conllevar pérdida de información o sesgos producidos por una mala aproximación estadística o por un conocimiento superficial del fenómeno a representar.
- El principal grueso de referencias y trabajos relacionados con índices compuestos ha orbitado alrededor del grupo de la Comisión Europea denominado *K4P-Knowledge4Policy*, perteneciente a la rama JRC (*Joint Research Centre*). Este hecho, juntamente con todo el análisis realizado en la tesis, hace concluir al candidato que los indicadores compuestos son una apuesta concreta por parte de la CE para tener relevancia en el futuro.
- Hoy en día no existe una metodología estándar ni única para la construcción de estos índices, aunque se recomienda (como se ha hecho en esta tesis), seguir la línea marcada por dicho grupo de la CE. Dentro de la misma se especifica que la selección del CI y los indicadores que lo componen es la actividad más importante para garantizar el éxito del índice, y sugiere que debe ocupar el 70% del tiempo y los esfuerzos de todo el proceso de construcción y cálculo.
- En cuanto a las primeras operaciones sobre los datos que alimentan los indicadores, debe realizarse un cribado para descartar valores atípicos y seleccionar una gestión de los datos ausentes. En la tesis se concluye que una aproximación  $1.5 \times \text{IQR}$  e imputaciones *hot-deck*, *cold-deck* o media incondicional son las más apropiadas para un entorno IoT flexible y de baja carga de computación. Otras opciones existen (que han sido estudiadas durante la tesis).
- Con respecto a las principales operaciones a efectuar para obtener un CI, destacan la normalización (situar los indicadores en una misma escala para poder ser agregados), en la que existe una gran variedad de métodos como *z-scores*, min-max, distancia a referencia... La ponderación es tal vez la fase más crítica ya que puede introducir serios errores de interpretación. La búsqueda de dichos factores puede acometerse desde varios puntos de vista, siendo (i) el equitativo (todos los indicadores de un grupo son considerados de igual importancia) el más utilizado; (ii) el basado en métodos estadísticos el que requiere más conocimiento y disposición de los datos y el más costoso computacionalmente (incluso irrealizable, en ocasiones); y (iii) el basado en la consulta expertos externos o público general un buen método en caso de querer plantear un sistema para toma de decisiones

públicas o para comparaciones. Los diferentes métodos (y una suerte de criterios para su elección) son proporcionados en esta tesis.

- Para un análisis exhaustivo de la validación de un CI, una serie de métodos estadísticos complejos como análisis de sensibilidad o análisis de incertidumbre deberían realizarse, lo que se ha concluido en esta tesis estar fuera del alcance de las competencias de un sistema IoT orientado al cálculo de dichos índices.
- Existen (principalmente con soluciones nuevas aparecidas muy recientemente, como CIF o COIN-R) algunas opciones de automatización de estos cálculos, no obstante, ninguna de ellas se basa en información medida directamente de las actividades/impactos de una entidad sobre un fenómeno particular. A diferencia de otros enfoques, el CIC (servicio desarrollado en esta tesis) se nutre de información recogida de fuentes de datos heterogéneas IoT.

### *Aplicación en entornos reales*

- La participación en proyectos de investigación ha permitido al candidato observar la adecuación de la solución propuesta para aportar valor en diferentes escenarios y casos de uso.
- Las casuísticas abordadas han resultado un complemento excepcional para validar los módulos de la arquitectura y los servicios desarrollados, existiendo despliegues con gran variedad de fuentes de datos y tecnologías (HTTP, MQTT, GPS, FTP, RFID, documentación estática, introducción manual...), topologías y requerimientos de despliegue (centralizado *on-premise*, centralizado *cloud*, centralizado híbrido, descentralizado parcialmente, descentralización total), capacidades del servicio (monitorización retrospectiva, análisis prescriptivo, predicción de índices, índice de fiabilidad, motor de reglas y generador de recomendaciones...) y métodos asociados del CI (normalización de reescalado, por distancia de referencia, ponderación equitativa, por método de asignación de presupuesto...). Esta variabilidad, y la oportunidad de poder validar la solución trabajando con diferentes actores de interés involucrados ha creado el caldo de cultivo ideal para considerarlo como una herramienta útil y con potencial.
- Como conclusión sobre la ejecución operativa, al tratarse de trabajos en pilotos de proyecto de investigación, su traslado a una operativa real en una empresa no ha podido ser constatada en una total expresión. Además, como nota negativa, se debe indicar que, en muchas ocasiones, la falta de datos, de comunicación interdepartamental, o falta de alineamiento con la estrategia de las corporaciones son barreras importantes para la adopción de este tipo de soluciones.

### *Consideraciones hacia un modelo de negocio*

- Existe una clara brecha entre los esfuerzos de investigación (en los sectores de IoT y de índices compuestos) y la adopción real de la tecnología por parte del mercado.
- Existe una tendencia por parte de los países y de la CE a enmendar esta situación a través de la creciente financiación a PYMES y start-ups de base tecnológica, situando IoT y *edge computing* como algunas de las prioridades en dichos programas.
- La solución propuesta en esta tesis (y, más en particular, la instanciación en uno de los escenarios de validación documentados) puede ser entendida como un resultado de investigación susceptible a ser transferido a un entorno de explotación.
- Las debilidades, amenazas, fortalezas y oportunidades expuestas, junto con el primer borrador de posible plan de negocio sobre dicho resultado explotable, hacen al candidato concluir que podría existir recorrido para pensar en próximas acciones hacia una comercialización del mismo.

## **7.3. Reflexiones sobre la innovación**

La primera reflexión realizada es que, a pesar de haber sido iniciada en el año 2019, considerando la rápida evolución de las tecnologías involucradas en la tesis, las hipótesis formuladas y su aplicación permanecen todavía vigentes a fecha de finalización de la misma (septiembre de 2022).

La automatización de la construcción de índices compuestos se encuentra en un momento álgido, con soluciones apareciendo cada año desde diferentes entidades e iniciativas. En lo concerniente al IoT, mientras que hace unos pocos años resultaba un campo de investigación por explorar, hoy en día se considera una tecnología razonablemente consolidada, con un amplio repertorio de ejemplos prácticos y arquitecturas de referencia, cuyos principales desafíos residen ahora en su aplicación exitosa e innovadora en diferentes sectores. En este sentido, los resultados de la tesis resultan de actualidad, ya que han propuesto la validación de un sistema IoT orientado a un fin específico, evitando caer en generalidades y con una aproximación eminentemente práctica.

Llama la atención, especialmente, la alineación con los objetivos de la Comisión Europea en cuanto a la futura dirección de la innovación en el campo del IoT. Según una publicación de marzo de 2022 [518] en la que se presentaba el *roadmap* de las prioridades de investigación, innovación y despliegue IoT en la Unión Europea, se destacó el papel del *edge computing* como catalizador principal de esta innovación, incluyendo otros como ciberseguridad, computación distribuida, Inteligencia Artificial (IA), realidad aumentada o *Tactile Internet*. Además, estos



conceptos están siendo reflejados en las convocatorias de financiación a través del programa *Horizon Europe*, con duración prevista hasta 2027.

Ayuda también a llegar a esta conclusión (es una propuesta innovadora) los artículos presentados y publicados por el candidato en revistas y congresos internacionales (ordenados por capítulo), que en cierto modo validan como novedosas las asunciones e implementaciones hechas en las diferentes secciones:

**Tabla 7.11:** Publicación de artículos por secciones – validación de la innovación

| Publicación  | Capítulo   |
|--|--|
| <i>Efficient Deployment of Predictive Analytics in Edge Gateways: Fall Detection Scenario</i>  | 2.3-Edge computing en NGIoT                      |
| <i>DevSecOps Methodology for Ng-IoT Ecosystem Development Lifecycle – ASSIST-IoT Perspective</i>   | 2.5.3-DevSecOps                                  |
| <i>Tactile Internet in Internet of Things Ecosystems</i>   | 2.5.4-Tactile Internet                           |
| <i>Reviewing SDN adoption strategies for Next Generation Internet of Things networks</i> (Febrero 2021)<br><i>A Novel Approach for Calculating Real-Time Composite Indicators Relying on Internet of Things and Industrial Data Spaces</i> (Mayo 2022) | 3-Definición e implementación de la arquitectura |
| <i>ASSIST-IoT: A Reference Architecture for Next Generation Internet of Things</i> (Septiembre 2022)   | 3.2.1-Principios de diseño                       |
| <i>A novel approach for assessing the port's environmental impacts in real time - the IoT based Port Environmental Index</i>   | 4-Cálculo y predicción de índices compuestos     |
| <i>Framework and Methodology for Establishing Port-City Policies Based on Real-Time Composite Indicators and IoT: A Practical Use-Case</i>   | 5.3-TCI  |
| <i>Leveraging IoT and prediction techniques to monitor COVID-19 restrictions in port terminals</i> (Nov.2019)<br><i>Advantage Of a Green Port Of The Future</i> (Sept.2021)  | 5.4-PEI  |
| <i>Functioning prototype of IoT and composite indicators for smart port environmental monitoring</i>   | 5.5- $\mu$ PEI                                   |
| <i>Small and medium ports' activities modelling: Introduction to the PIXEL approach.</i> (Oct.2019)<br><i>Leveraging IoT and prediction techniques to monitor COVID-19 restrictions in port terminals</i> (Nov.2021)                                   | 5.6-CEDI   |

Este hecho se refrenda al realizar algunas búsquedas sobre los temas investigados en esta tesis doctoral. Al buscar en Google Scholar los términos

“*composite indicator IoT*” (se realizó la búsqueda en inglés para abarcar el espectro internacional), los primeros cuatro resultados son artículos escritos por el doctorando (de entre la tabla anterior). Por otro lado al efectuar la misma búsqueda en la barra de navegación de Google, también los 4 primeros resultados corresponden a artículos de los señalados anteriormente, apareciendo 5 entre los primeros 8 (primera página de búsqueda). Los datos indicados se refieren a consultas realizadas a mediados de agosto 2022, utilizando navegadores sin caché.

La siguiente observación es que la propuesta tecnológica de esta tesis ha resultado de aplicación para resolver problemas reales asociados a la toma de decisiones (evidenciado en escenarios de ciudad-puerto). En este sentido, se destaca la utilidad de la solución, pero también el punto diferenciador que ha supuesto la participación del candidato en proyectos I+D financiados por la Comisión Europea y por el plan de ayudas públicas regional de la Agencia Valenciana de Innovación. De no haber sido por esta financiación, muchos desarrollos y validaciones expuestos en este documento difícilmente hubieran visto la luz.

Es por ello que, como última reflexión (hilando con el argumento esgrimido en la sección 6), el doctorando se atreve a concluir que cuando se refuerza el vínculo (a través de programas públicos y de iniciativas privadas) entre el ámbito universitario/académico y una puesta en marcha en entornos (ceranos a) reales, ambas partes se ven beneficiadas. Por la parte investigadora, trabajos como este pueden llevarse a cabo, y por la parte industrial, se favorece una transición de tecnologías en bajos niveles de madurez a una aplicación real, aportando, en muchos casos, un valor que no puede conseguirse a partir de las herramientas comerciales disponibles.

## 7.4. Trabajos futuros

Como se ha evidenciado en la sección 7.1, la tesis doctoral ha cumplido sus objetivos planificados al comienzo. En esta subsección se plantean cuatro grupos de trabajos futuros que podrían servir para complementar la investigación realizada y para abrir nuevas líneas de investigación sobre los campos del Internet de las Cosas y los índices compuestos.

### *Mejoras en la tecnología producida*

A continuación se lista una serie de actividades que han sido identificadas para la mejora tecnológica de la solución propuesta. No han sido abordadas durante este trabajo al no ser un aspecto requerido por la arquitectura ni por los escenarios de validación o al encontrarse fuera el alcance planteado:

- Creación de una doble aproximación a la persistencia, permitiendo convivir bases relacionales y no relacionales y gestionando su acceso desde un único punto a través de un API inteligente que pudiera manejar el sistema de persistencia.

- Incrementar el nivel de automatización en: (1) la ejecución de servicios. Una mejora interesante consistiría en pasar del modelo semiautomatizado actual (acuerdos de índices y modelos de datos) a un enfoque completamente automatizado. Esto requeriría desarrollos propios (posiblemente sobre soluciones de orquestación de contenedores) que merecería un trabajo aparte. Y en (2) la instalación de agentes conectores de datos (a través de una plataforma como la propuesta por el proyecto DataPorts, en lugar de instalar las imágenes desde línea de comandos).
- Convertir el sistema IoT en una plataforma IoT. Como se ha comentado, el objetivo era diseñar y desarrollar una arquitectura IoT con un objetivo específico (servicio CI), por tanto se han excluido características propias de plataformas IoT.
- En línea con lo anterior, sería interesante desarrollar un módulo más completo de visualización de datos en tiempo real. Se ha delegado en las capacidades de Elasticsearch y otros *frameworks* utilizados la monitorización de las fuentes, enfocando la representación a los resultados de ejecución de los servicios. Esto podría ser claramente una mejora de la solución.
- Validación de la solución en escenarios centrados en otros sectores. Los cinco casos planteados en la tesis se enmarcan en entornos ciudad-puerto. Sería interesante aplicar la solución a otras casuísticas, que podrían presentar desafíos y consideraciones que hubieran podido ser pasadas por alto en este trabajo.
- Inclusión del resto de herramientas estadísticas en el servicio de cálculo del CI. En la sección 4 se ha descrito una serie de procesos estadísticos que, dentro del espectro de construcción de un CI, se realizan *offline* por parte de personal especializado. Una buena línea de trabajo sería mejorar el servicio (y la arquitectura, si fuera necesario) para incluir dichos métodos estadísticos dentro de la solución de una manera automatizada. En especial, sería interesante incluir un módulo EDA (para realizar un análisis exploratorio de los datos) y servicios de análisis de sensibilidad (*Sensitivity Analysis* – SA) e incertidumbre (*Uncertainty Analysis* – UA).

### ***Posible incorporación de otras tecnologías relacionadas***

En este grupo se listan varias propuestas de trabajo sobre la solución mediante la incorporación de conceptos/técnicas/tecnologías directamente relacionadas y con las que el candidato (y el grupo de investigación) se encuentra familiarizado. Estas tecnologías añadirían más valor a la solución, aun habiendo estado fuera del alcance de este trabajo:

- La privacidad ha sido incorporada a la solución como una serie de recomendaciones y técnicas individuales, sin hacer uso de una tecnología específica. Como línea interesante de investigación, se plantea incorporar la tecnología IOTA para generar un grafo acíclico dirigido para el registro de datos e interacciones.
- La interoperabilidad se ha tratado en esta tesis de una forma muy tangencial, basando ésta en el establecimiento de modelos de datos conocidos por todos los módulos y servicios, sin haber aplicado ninguna tecnología adicional. Si se plantea que la solución interactúe con plataformas IoT (por ejemplo, ya existentes en la compañía), se podría trabajar en el futuro en varias líneas: (i) interoperabilidad a nivel de dispositivo, (ii) interoperabilidad a nivel de *middleware*, (iii) interoperabilidad a nivel semántico y (iv) interoperabilidad a nivel de aplicación.
- Podría trabajarse en la incorporación de conectores IDS (International Data Spaces). Este aspecto es especialmente relevante para los planteamientos de esta tesis ya que permitiría un intercambio de datos basado en estándares industriales que además pudiera utilizarse para el establecimiento de un “mercado de datos” (por ejemplo, el resultado del CI de todas las empresas marítimo-portuarias). Además, estaría mejorándose las provisiones de seguridad y privacidad de la solución.
- Podría extenderse la capacidad de los agentes planteados para aceptar la inclusión de nuevos sensores o fuentes de datos que trabajaran con tecnologías más recientes de comunicación como TSN, OPC-UA, UltraWideBand (UWB) o 5G.
- La solución podría incorporar tecnologías Big Data que no han sido consideradas tales como Data Mining, Hadoop, Redes Neuronales o Business Intelligence, que podrían ser combinadas en cierto modo con la solución.

### ***Trabajo hacia la creación de un producto***

Siguiendo la línea expuesta en la sección 6, el resultado explotable seleccionado en la tesis debería experimentar una serie de mejoras para poder convertirse en un producto (comercializable en un futuro). Además, otras actividades relacionadas deberían llevarse a cabo:

- Establecimiento de un soporte hardware para el procesamiento en *edge* de la arquitectura. En los escenarios de validación planteados, se ha hecho uso de equipamiento *edge* estándar (Raspberry Pi, *clusters*...), sobre el que se instalaba algún módulo de la plataforma. De cara a una formalización del producto en un “kit”, convendría disponer de hardware *edge* preparado con la arquitectura preinstalada.
- En el caso de proceder con el punto anterior, nuevas líneas podrían abrirse como (i) la encapsulación del “kit” prototipo siguiendo estándares industriales como IEC:60509, adquiriendo clasificaciones

como IP65, IP66 o IP67 o (ii) el análisis de duración de baterías de dichos equipos para asegurar un servicio durante un tiempo determinado (en función de la movilidad de cada instalación).

- Análisis serio de competidores en el sector donde se eligiera centrar la solución en primer término (por ejemplo, para entidades públicas de cara a certificaciones/auditorías).
- Refinamiento del DAFO y el Plan de Negocio esbozados en este trabajo.
- Búsqueda de financiación para las primeras fases de creación de empresa de base tecnológica.

### *Tecnologías emergentes como potenciales líneas futuras de investigación*

Acorde a las nuevas tendencias observadas en diferentes fuentes [518] [546], [547][548] para el IoT en los próximos años, las siguientes líneas de investigación podrían resultar de interés:

- Aplicación de principios *cloud-native* en entornos *edge computing*, incluyendo la orquestación de servicios y recursos virtualizados a lo largo del continuo de computación.
- Aplicación de Inteligencia Artificial fiable (*Trustworthy AI*) y explicable (*Explainable AI – XAI*).
- Adición de inteligencia a los procesos de DevSecOps anunciados (*CognitiveOps*).
- Espacios de datos (*data spaces*) y mercados de datos (*data markets*).
- Gemelo Digital
- IoT aumentado: el Internet de las Cosas explorado y desplegado en conjunción con tecnologías de realidad aumentada.
- 5G, IoT y *edge computing* parecen ir de la mano en las tendencias para los próximos meses/años.
- De Aprendizaje Federado (analizado en esta tesis) a aprendizaje de enjambre (*swarm learning*) y MLOps (*pipelines* de AI/ML descentralizadas).
- Virtualización en entornos de red (SDN y NFV).
- Inteligencia Artificial en el extremo de la red (*AI on far edge*).



# Referencias

- [1] “IDC forecasts connected IoT devices to generate 79.4ZB of data in 2025.” , Online: <https://futureiot.tech/idc-forecasts-connected-iot-devices-to-generate-79-4zb-of-data-in-2025/>.
- [2] J. ho Park, M. M. Salim, J. H. Jo, J. C. S. Sicato, S. Rathore, and J. H. Park, “CIoT-Net: a scalable cognitive IoT based smart city network architecture,” *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1–20, Dec. 2019.
- [3] M. Saisana and S. Tarantola, *State-of-the-art report on current methodologies and practices for composite indicator development*, vol. 214. Citeseer, 2002.
- [4] P. Frederiksen *et al.*, “Policy influence of indicators–POINT.” 2020.
- [5] “Economic Commission for Europe In-depth review of leading, composite and sentiment indicators,” 2014.
- [6] M. Dinnetz, “Technology Transfer From Research to Impact.”
- [7] P. Annicchino, A. Bréchine, F. M. Facca, A. Heijnen, and F. M. Castro, “NG-IoT - D3.1 - IoT research, innovation and deployment priorities in the EU,” 2018.
- [8] “IoT Platform Companies Landscape 2021/2022.” , Online: <https://iot-analytics.com/iot-platform-companies-landscape/>.
- [9] “Definition of IoT Platforms - Gartner Information Technology Glossary.” , Online: <https://www.gartner.com/en/information-technology/glossary/iot-platforms>.
- [10] “IoT-EPI.” , Online: <https://iot-epi.eu/>.
- [11] G. Ilieva and T. Yankova, “IoT System Selection as a Fuzzy Multi-Criteria Problem,” *Sensors*, vol. 22, p. 4110, Aug. 2022.
- [12] L. Babun, K. Denney, Z. B. Celik, P. McDaniel, and S. Uluagac, “A survey on IoT platforms: Communication, security, and privacy perspectives,” *Computer Networks*, vol. 192, p. 108040, Aug. 2021.
- [13] “INTER-IoT Project H2020 official website,” <https://inter-iot.eu/>, 31-Dec-2020. .
- [14] “ACTIVAGE Project : Internet of Things (IoT) for ageing well.” , Online: <http://www.activageproject.eu/>.
- [15] G. Fortino, A. Guerrieri, P. Pace, C. Savaglio, and G. Spezzano, “IoT Platforms and Security: An Analysis of the Leading Industrial/Commercial Solutions,” *Sensors 2022, Vol. 22, Page 2196*, vol. 22, no. 6, p. 2196, Mar. 2022.
- [16] M. Weyrich and C. Ebert, “Reference architectures for the internet of things,” *IEEE Softw*, vol. 33, no. 1, pp. 112–116, Jan. 2016.



- 
- [17] “Introduction to the Architectural Reference Model.”
- [18] L. Perry, *Internet of things for architects. IoT Architecture and Core IoT Modules*. Birmingham: Packt Publishing Ltd, 2018.
- [19] C. E. Palau *et al.*, “D6.1 PIXEL Information System architecture and design v1,” Sep. 2021.
- [20] “Internet of Things - Architecture — IOT-A: Internet of Things Architecture.” , Online: <https://www.iot-a.eu/>.
- [21] “Internet of Things Architecture | IoT-A Project | Fact Sheet | FP7 | CORDIS | European Commission.” , Online: <https://cordis.europa.eu/project/id/257521>.
- [22] “A Reference Architecture for the Internet of Things.” , Online: <https://wso2.com/whitepapers/a-reference-architecture-for-the-internet-of-things/>.
- [23] “Plattform Industrie 4.0 - Homepage.” , Online: <https://www.plattform-i40.de/IP/Navigation/EN/Home/home.html>.
- [24] K. Schweichhart, “Reference Architectural Model Industrie 4.0 (RAMI 4.0) - An Introduction.”
- [25] “IEC PAS 63088. (2017). Smart Manufacturing – Reference Architecture Model Industry 4.0 (RAMI4.0). V1.0 | StandICT.eu 2023.” , Online: <https://www.standict.eu/standards-repository/iec-pas-63088-2017-smart-manufacturing-reference-architecture-model-industry>.
- [26] “The Industrial Internet of Things Volume G1: Reference Architecture,” 2019.
- [27] A. Torkaman and M. A. Seyyedi, “Analyzing IoT Reference Architecture Models,” *International Journal of Computer Science and Software Engineering (IJCSSE)*, vol. 5, no. 8, 2016.
- [28] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, “A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective,” *Internet of Things Journal, IEEE*, vol. 1, pp. 349–359, Aug. 2014.
- [29] A. el Hakim, “Internet of Things (IoT) System Architecture and Technologies, White Paper.” Aug-2018.
- [30] “Y.2060 : Overview of the Internet of things.” , Online: <https://www.itu.int/rec/T-REC-Y.2060-201206-I/en>.
- [31] ISO/IEC/IEEE 42010, “ISO/IEC/IEEE 42010 - Systems and software engineering - Architecture description,” 2011.
- [32] A.-I. Consortium, “D3.1 – State-of-the-Art and Market Analysis Report,” Jan. 2021.
- [33] NGI Project, “A Vision of the Future Internet,” 2020.

- [34] A. Sharma, M. Kumar, and S. Agarwal, “A Complete Survey on Software Architectural Styles and Patterns,” in *Procedia Computer Science*, 2015, vol. 70, pp. 16–28.
- [35] A. Willner and V. Gowtham, “Toward a Reference Architecture Model for Industrial Edge Computing,” *IEEE Communications Standards Magazine*, vol. 4, pp. 42–48, Aug. 2020.
- [36] OpenfogConsortium, “OpenFog Reference Architecture for Fog Computing Produced,” *Reference Architecture*, no. February, pp. 1–162, 2017.
- [37] ECC and AII, “Edge Computing Reference Architecture 2.0,” 2017.
- [38] CREATE-IoT Project, “D6.3. Assessment of convergence and interoperability in LSP platforms,” 2020.
- [39] FAR-EDGE Project, “D2.4. FAR-EDGE Architecture and Components Specification,” 2017.
- [40] AIOTI WG Standardisation, “High Level Architecture (HLA) Release 5.0,” 2020.
- [41] A. Sharma, M. Kumar, and S. Agarwal, “A Complete Survey on Software Architectural Styles and Patterns,” in *Procedia Computer Science*, 2015, vol. 70, pp. 16–28.
- [42] M. Richards, *Software Architecture Patterns*. O’Reilly Media, 2015.
- [43] R. A. P. Rajan, “Serverless Architecture - A Revolution in Cloud Computing,” in *2018 10th International Conference on Advanced Computing, ICoAC 2018*, 2018, pp. 88–93.
- [44] C. M. Mackenzie, F. McCabe, P. F. Brown, P. Net, R. Metz, and A. Hamilton, “Reference Model for Service Oriented Architecture 1.0,” 2006.
- [45] S. T. Alanazi, N. Abdullah, M. Anbar, and O. A. Al-Wesabi, “Evaluation Approaches of Service Oriented Architecture (SOA) - A Survey,” in *2nd International Conference on Computer Applications and Information Security, ICCAIS 2019*, 2019.
- [46] J. Prathap Irudayaraj and Mp. Research Scholar, “Comparative study on Cloud Software Architecture: Monolithic, SOA,” *Journal of Applied Science and Computations*, vol. 6, pp. 2257–2261, May 2019.
- [47] C. M. Mackenzie, F. McCabe, P. F. Brown, P. Net, R. Metz, and A. Hamilton, “Reference Model for Service Oriented Architecture 1.0,” 2006.
- [48] T. Cerny, M. J. Donahoo, and J. Pechanec, “Disambiguation and comparison of SOA, microservices and self-contained systems,” in *Proceedings of the 2017 Research in Adaptive and Convergent Systems, RACS 2017*, 2017, vol. 2017-Janua, pp. 228–235.
- [49] Z. Wu, S. Deng, and J. Wu, “Service-Oriented Architecture and Web

- Services,” in *Service Computing*, Elsevier, 2015, pp. 17–42.
- [50] E. Wolff, *Microservices: Flexible Software Architecture*. Addison-Wesley Professional, 2016.
- [51] M. Richards, *Microservices vs. Service-Oriented Architecture*. O’Reilly Media, 2016.
- [52] P. Di Francesco, I. Malavolta, and P. Lago, “Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption,” in *Proceedings - 2017 IEEE International Conference on Software Architecture, ICOSA 2017*, 2017, pp. 21–30.
- [53] E. Berrio-Charry, J. Vergara-Vargas, and H. Umana-Acosta, “A Component-Based Evolution Model for Service-Based Software Architectures,” in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2020, vol. 2020-Octob, pp. 111–115.
- [54] J. Ghofrani and D. Lübke, “Challenges of Microservices Architecture: A Survey on the State of the Practice,” in *10th Central European Workshop on Services and their Composition*, 2018.
- [55] “Diferencias entre IaaS, PaaS y SaaS.” , Online: <https://www.redhat.com/es/topics/cloud-computing/iaas-vs-paas-vs-saas>.
- [56] Kiryong Ha *et al.*, “The Impact of Mobile Multimedia Applications on Data Center Consolidation,” in *2013 IEEE International Conference on Cloud Engineering (IC2E)*, 2013, pp. 166–176.
- [57] “Webpages Are Getting Larger Every Year, and Here’s Why it Matters.” , Online: <https://www.pingdom.com/blog/webpages-are-getting-larger-every-year-and-heres-why-it-matters/>.
- [58] “10 Disadvantages & Risks of Cloud Computing | by Sarah Lavinski | FAUN Publication.” , Online: <https://faun.pub/10-disadvantages-risks-of-cloud-computing-35111de75611>.
- [59] “5 Trends Drive the Gartner Hype Cycle for Emerging Technologies, 2020.” , Online: <https://www.gartner.com/smarterwithgartner/5-trends-drive-the-gartner-hype-cycle-for-emerging-technologies-2020>.
- [60] “Fog computing and its role in the Internet of Things,” <https://readwrite.com/2020/10/30/fog-computing-and-its-role-in-the-internet-of-things/>, 30-Oct-2020. .
- [61] “IDC forecasts connected IoT devices to generate 79.4ZB of data in 2025 - FutureIoT.” , Online: <https://futureiot.tech/idc-forecasts-connected-iot-devices-to-generate-79-4zb-of-data-in-2025/>.
- [62] S. A. Mostafavi, A. Dawlatnazar, and F. Paydar, “Edge Computing for IoT: Challenges and Solutions,” *Journal of Communications Technology, Electronics and Computer Science, ISSN 2457-905X*, no. 26, 2019.

- [63] “LightKone Project - Deliverable D3.1 - Initial Runtime Edge Computing,” [https://wordix.inesctec.pt/wp-lightkone/wp-content/uploads/2019/04/D3.1\\_InitialRuntimeEdgeComputingSystem.pdf](https://wordix.inesctec.pt/wp-lightkone/wp-content/uploads/2019/04/D3.1_InitialRuntimeEdgeComputingSystem.pdf), 22-Dec-2020. .
- [64] “AWS vs Cisco Fog Computing and Cloud,” <https://etherealmind.com/aws-iot-vs-cisco-fog-computing-cloud-vs-network-iot/>, 29-Dec-2020. .
- [65] “Next steps in the architecture and design of Edge Computing,” <https://www.openstack.org/use-cases/edge-computing/edge-computing-next-steps-in-architecture-design-and-testing/>, 29-Dec-2020. .
- [66] “Open Edge Computing (OEC) association website,” <https://www.openedgecomputing.org/>, 26-Dec-2020. .
- [67] ETSI, “Multi-Access Edge Computing (MEC) association website,” <https://www.etsi.org/technologies/multi-access-edge-computing>, 29-Dec-2020. .
- [68] K. Dolui and S. K. Datta, “Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing,” in *2017 Global Internet of Things Summit (GIoTS)*, 2017.
- [69] C. C. Byers, “Architectural Imperatives for Fog Computing: Use Cases, Requirements, and Architectural Techniques for Fog-Enabled IoT Networks,” *IEEE Communications Magazine*, vol. 55, no. 8, Aug. 2017.
- [70] R. K. Barik *et al.*, “Mist Data: Leveraging Mist Computing for Secure and Scalable Architecture for Smart and Connected Health,” *Procedia Comput Sci*, vol. 125, 2018.
- [71] M. Gushev, “Dew Computing Architecture for Cyber-Physical Systems and IoT,” *Internet of Things*, vol. 11, Sep. 2020.
- [72] K. Habak, M. Ammar, K. A. Harras, and E. Zegura, “Femto Clouds: Leveraging Mobile Devices to Provide Cloud Service at the Edge,” in *2015 IEEE 8th International Conference on Cloud Computing*, 2015.
- [73] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, “Cloudlets: at the Leading Edge of Mobile-Cloud Convergence,” in *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*, 2014.
- [74] H. Elazhary, “Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions,” *Journal of Network and Computer Applications*, vol. 128, Feb. 2019.
- [75] Z. Wang, F. Gao, and X. Jin, “Optimal deployment of cloudlets based on cost and latency in Internet of Things networks,” *Wireless Networks*, vol. 26, no. 8, Nov. 2020.

- [76] C. Avasalcai, I. Murturi, and S. Dustdar, “Edge and Fog: A Survey, Use Cases, and Future Challenges,” in *Fog Computing*, Wiley, 2020.
- [77] A. B. Manju and S. Sumathy, “Efficient Load Balancing Algorithm for Task Preprocessing in Fog Computing Environment,” 2019.
- [78] C.-C. Lin and J.-W. Yang, “Cost-Efficient Deployment of Fog Computing Systems at Logistics Centers in Industry 4.0,” *IEEE Trans Industr Inform*, vol. 14, no. 10, Oct. 2018.
- [79] E. Marín-Tordera, X. Masip-Bruin, J. García-Almiñana, A. Jukan, G.-J. Ren, and J. Zhu, “Do we all really know what a fog node is? Current trends towards an open definition,” *Comput Commun*, vol. 109, Sep. 2017.
- [80] P. Habibi, M. Farhoudi, S. Kazemian, S. Khorsandi, and A. Leon-Garcia, “Fog Computing: A Comprehensive Architectural Survey,” *IEEE Access*, vol. 8, 2020.
- [81] OpenFog Consortium, “OpenFog Reference Architecture for Fog Computing,” [http://site.ieee.org/denver-com/files/2017/06/OpenFog\\_Reference\\_Architecture\\_2\\_09\\_17-FINAL-1.pdf](http://site.ieee.org/denver-com/files/2017/06/OpenFog_Reference_Architecture_2_09_17-FINAL-1.pdf), 31-Dec-2020. .
- [82] “Cisco IOX Datasheet,” <https://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/iox/datasheet-c78-736767.html>, 31-Dec-2020. .
- [83] LFEEdge, “LFEEdge initiative,” <https://www.lfedge.org/projects/edgexfoundry/>, 31-Dec-2020. .
- [84] LFEEdge, “EdgeXFoundry official website,” <https://www.edgexfoundry.org/>, 31-Dec-2020. .
- [85] “FogLamp-Fledge code reference,” <https://github.com/foglamp/FogLAMP>, 31-Dec-2020. .
- [86] FBK, “Fondazione Bruno Kessler official website,” <https://www.fbk.eu/en/>, 31-Dec-2020. .
- [87] Eclipse, “fog05 code reference,” <https://fog05.io/>, 31-Dec-2020. .
- [88] P. Pop *et al.*, “The FORA Fog Computing Platform for Industrial IoT,” *Inf Syst*, vol. 98, May 2021.
- [89] “Custom Resources extension - Kubernetes,” <https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/custom-resources/>, 31-Dec-2020. .
- [90] A. L. Samuel, “Some Studies in Machine Learning Using the Game of Checkers.”
- [91] R. Mena-Yedra, R. Gavaldà, and J. Casas, “Adarules: Learning rules for real-time road-traffic prediction,” *Transportation Research Procedia*, vol. 27, pp.

- 11–18, 2017.
- [92] R. P. Masini, M. C. Medeiros, and E. F. Mendes, “Machine Learning Advances for Time Series Forecasting,” *J Econ Surv*, Dec. 2020.
- [93] F. Firouzi, B. Farahani, F. Ye, M. Barzegari, and A. J. D’angelo, “Machine Learning for IoT,” *Intelligent Internet of Things*, pp. 243–313, 2020.
- [94] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, “An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction,” *Procedia Soc Behav Sci*, vol. 96, pp. 653–662, Nov. 2013.
- [95] K. Sharma and R. Nandal, “A literature study on machine learning fusion with IoT,” *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*, vol. 2019-April, pp. 1440–1445, Apr. 2019.
- [96] A. Khattab and N. Youssry, “Machine learning for IoT systems,” *Internet of Things (IoT): Concepts and Applications*, pp. 105–127, Jan. 2020.
- [97] D. Mukherjee and S. Datta, “Incremental time series algorithms for IoT analytics: An example from autoregression,” *ACM International Conference Proceeding Series*, vol. 04-07-January-2016, Jan. 2016.
- [98] A. A. Cook, G. Misirli, and Z. Fan, “Anomaly Detection for IoT Time-Series Data: A Survey,” *IEEE Internet Things J*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.
- [99] J.-R. Jiang, J.-B. Kao, Y.-L. Li, C. : Jiang, J.-R. ; Kao, and J.-B. ; Li, “Semi-Supervised Time Series Anomaly Detection Based on Statistics and Deep Learning,” *Applied Sciences 2021, Vol. 11, Page 6698*, vol. 11, no. 15, p. 6698, Jul. 2021.
- [100] P. Ni, C. Zhang, and Y. Ji, “A hybrid method for short-term sensor data forecasting in Internet of Things,” *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2014*, pp. 369–373, Dec. 2014.
- [101] S. R. Beeram and S. Kuchibhotla, “Time Series Analysis on Univariate and Multivariate Variables: A Comprehensive Survey,” *Lecture Notes in Networks and Systems*, vol. 134, pp. 119–126, 2021.
- [102] M. Fahim and A. Sillitti, “Anomaly Detection, Analysis and Prediction Techniques in IoT Environment: A Systematic Literature Review,” *IEEE Access*, vol. 7, pp. 81664–81681, 2019.
- [103] S. J. Russell and P. Norvig, “Artificial intelligence: a modern approach. Malaysia,” p. 1145, 2016.
- [104] “Build Foundation for Time Series Forecasting | by Ajay Tiwari | Towards Data Science.” , Online: <https://towardsdatascience.com/time-series-forecasting-968192b3781a>.

- 
- [105] “ARIMA vs. Prophet: Forecasting Air Passenger Numbers | by Michael Grogan | Towards Data Science.” , Online: <https://towardsdatascience.com/arima-vs-prophet-forecasting-air-passenger-numbers-4e01b2d93608>.
- [106] “Taxonomy of Time Series Forecasting Problems.” , Online: <https://machinelearningmastery.com/taxonomy-of-time-series-forecasting-problems/>.
- [107] J. du Preez and S. F. Witt, “Univariate versus multivariate time series forecasting: an application to international tourism demand,” *Int J Forecast*, vol. 19, no. 3, pp. 435–451, Jul. 2003.
- [108] T. Handhika, Murni, D. P. Lestari, and I. Sari, “Multivariate time series classification analysis: State-of-the-art and future challenges,” *IOP Conf Ser Mater Sci Eng*, vol. 536, no. 1, p. 012003, Jun. 2019.
- [109] “The Advantages of Using an Independent Group T-Test.” , Online: <https://sciencing.com/advantages-using-independent-group-ttest-8647277.html>.
- [110] G. Chavan and B. Momin, “An integrated approach for weather forecasting over Internet of Things: A brief review,” *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, pp. 83–88, Oct. 2017.
- [111] M. Awad and R. Khanna, “Support Vector Regression,” *Efficient Learning Machines*, pp. 67–80, 2015.
- [112] F. Hamami and I. A. Dahlan, “Univariate Time Series Data Forecasting of Air Pollution using LSTM Neural Network,” *2020 International Conference on Advancement in Data Science, E-Learning and Information Systems, ICADEIS 2020*, Oct. 2020.
- [113] I. Lacalle *et al.*, “D4.4 - Predictive Algorithms v2,” Apr. 2020.
- [114] “Time Series Forecast in Python using SARIMAX and PROPHET.” , Online: <https://towardsdatascience.com/time-series-forecast-in-python-using-sarimax-and-prophet-c970e6056b5b>.
- [115] A. Mindang and P. Siripongwutikorn, “Solar Power Prediction in IoT Devices using Environmental and Location Factors,” *ACM International Conference Proceeding Series*, pp. 119–123, Jun. 2020.
- [116] “ARIMA & SARIMA: Real-World Time Series Forecasting - neptune.ai.” , Online: <https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide>.
- [117] G. A. N. Pongdatu and Y. H. Putra, “Seasonal Time Series Forecasting using SARIMA and Holt Winter’s Exponential Smoothing,” *IOP Conf Ser Mater Sci Eng*, vol. 407, p. 12153, Sep. 2018.

- [118] “Prophet | Forecasting at scale.” , Online: <https://facebook.github.io/prophet/>.
- [119] “Time Series Forecasting — ARIMA vs Prophet | by Krish Hariharan | Analytics Vidhya | Medium.” , Online: <https://medium.com/analytics-vidhya/time-series-forecasting-arima-vs-prophet-5015928e402a>.
- [120] “Deep Learning for Time Series Forecasting.” , Online: <https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/>.
- [121] S. Lorenz, “Predicting Time Series for the IoT.”
- [122] “Gentle Introduction to the Adam Optimization Algorithm for Deep Learning.” , Online: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>.
- [123] B. Abdel Ouahab Ikram, B. Anouar Abdelhakim, A. Abdelali, B. Zafar, and B. Mohammed, “Deep Learning architecture for temperature forecasting in an IoT LoRa based system,” *Proceedings of the 2nd International Conference on Networking, Information Systems & Security - NISS19*, 2019.
- [124] Z. Yu, X. Zheng, F. Huang, W. Guo, L. Sun, and Z. Yu, “A framework based on sparse representation model for time series prediction in smart city,” *Front Comput Sci*, vol. 15, no. 1, p. 151305, Feb. 2021.
- [125] M. P. Raju and A. J. Laxmi, “IOT based Online Load Forecasting using Machine Learning Algorithms,” *Procedia Comput Sci*, vol. 171, pp. 551–560, Jan. 2020.
- [126] Y. Cheng, X. Zhou, S. Wan, and K. K. R. Choo, “Deep belief network for meteorological time series prediction in the internet of things,” *IEEE Internet Things J*, vol. 6, no. 3, pp. 4369–4376, Jun. 2019.
- [127] C. Yu *et al.*, “Time Series Analysis and Forecasting of the Hand-Foot-Mouth Disease Morbidity in China Using An Advanced Exponential Smoothing State Space TBATS Model,” *Infect Drug Resist*, vol. 14, p. 2809, 2021.
- [128] M. Karimpour, A. Karimpour, K. Kompany, and A. Karimpour, “Online Traffic Prediction Using Time Series: A Case study,” in *Integral Methods in Science and Engineering, Volume 2: Practical Applications*, C. Constanda, M. Dalla Riva, P. D. Lamberti, and P. Musolino, Eds. Cham: Springer International Publishing, 2017, pp. 147–156.
- [129] “What is edge computing and why does it matter? | Network World.” , Online: <https://www.networkworld.com/article/3224893/what-is-edge-computing-and-how-it-s-changing-the-network.html>.
- [130] “Google AI Blog: Federated Learning: Collaborative Machine Learning without Centralized Training Data.” , Online: <https://ai.googleblog.com/2017/04/federated-learning-collaborative.html>.



- 
- [131] K. Bonawitz *et al.*, “TOWARDS FEDERATED LEARNING AT SCALE: SYSTEM DESIGN,” 2019.
- [132] O. Elloumi, M. Carugi, J. P. Desbenoit, G. Karagiannis, and P. Murdock, “AIOTI-WG3-High Level Architecture (HLA) Release 5.0,” 2020, Online: [https://aioti.eu/wp-content/uploads/2020/12/AIOTI\\_HLA\\_R5\\_201221\\_Published.pdf](https://aioti.eu/wp-content/uploads/2020/12/AIOTI_HLA_R5_201221_Published.pdf).
- [133] B. Yuan, S. Ge, and W. Xing, “A Federated Learning Framework for Healthcare IoT devices,” May 2020.
- [134] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” Feb. 2016.
- [135] “Architecture for Scalable, Self-human-centric, Intelligent, Secure, and Tactile next generation IoT D3.5 ASSIST-IoT Architecture Definition-Initial,” 2020.
- [136] P. Kairouz *et al.*, “Advances and Open Problems in Federated Learning,” *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, Jun. 2021.
- [137] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated Learning: Strategies for Improving Communication Efficiency,” Oct. 2016.
- [138] J. Konečný and P. Richtárik, “Randomized Distributed Mean Estimation: Accuracy vs. Communication,” *Front Appl Math Stat*, vol. 4, p. 62, Dec. 2018.
- [139] Z. Wang and Q. Hu, “Blockchain-based Federated Learning: A Comprehensive Survey,” Oct. 2021.
- [140] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, “Revisiting Distributed Synchronous SGD,” Apr. 2016.
- [141] J. Qian, L. K. Hansen, X. Fafoutis, P. Tiwari, and H. M. Pandey, “Robustness analytics to data heterogeneity in edge computing,” *Comput Commun*, vol. 164, pp. 229–239, Dec. 2020.
- [142] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, “FedPD: A Federated Learning Framework with Optimal Rates and Adaptivity to Non-IID Data.” Aug-2020.
- [143] Q. Li *et al.*, “A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection,” Jul. 2019.
- [144] Y. Liu *et al.*, “FedVision: An Online Visual Object Detection Platform Powered by Federated Learning,” *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pp. 13172–13179, Jan. 2020.

- [145] “Tencent’s WeBank applying ‘federated learning’ in A.I. - Digital Finance.” , Online: <https://www.digfingroup.com/webank-clustar/>.
- [146] “Major pharma companies, including Novartis and Merck, build federated learning platform for drug discovery | VentureBeat.” , Online: <https://venturebeat.com/ai/major-pharma-companies-including-novartis-and-merck-build-federated-learning-platform-for-drug-discovery/>.
- [147] I. Ceballos *et al.*, “SplitNN-driven Vertical Partitioning,” Aug. 2020.
- [148] “High Level Architecture (HLA) Release 5.0 AIOTI WG Standardisation,” 2020.
- [149] D. Roca, J. v. Quiroga, M. Valero, and M. Nemirovsky, “Fog function virtualization: A flexible solution for IoT applications,” *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC): 8-11 May, 2017 Valencia, Spain*, pp. 74–80, Jun. 2017.
- [150] “ETSI - Standards for NFV - Network Functions Virtualisation | NFV Solutions.” , Online: <https://www.etsi.org/technologies/nfv>.
- [151] “Kubernetes.” , Online: <https://kubernetes.io/>.
- [152] “Swarm mode overview | Docker Documentation.” , Online: <https://docs.docker.com/engine/swarm/>.
- [153] “Apache Mesos.” , Online: <https://mesos.apache.org/>.
- [154] M. Alam, J. Rufino, J. Ferreira, S. Ahmed, N. Shah, and Y. Chen, “Orchestration of Microservices for IoT Using Docker and Edge Computing,” *IEEE Communications Magazine*, vol. 56, pp. 118–123, Aug. 2018.
- [155] “K3s: Lightweight Kubernetes.” , Online: <https://k3s.io/>.
- [156] “MicroK8s - Zero-ops Kubernetes for developers, edge and IoT.” , Online: <https://microk8s.io/>.
- [157] “GitHub - project-akri/akri: A Kubernetes Resource Interface for the Edge.” , Online: <https://github.com/project-akri/akri>.
- [158] R. bin Sulaiman, “Future Threats to Internet of Things (IoT) Security & Privacy: A Survey,” *SSRN Electronic Journal*, Aug. 2019.
- [159] E. Shaikh, I. Mohiuddin, and A. Manzoor, “Internet of Things (IoT): Security and Privacy Threats,” 2019, pp. 1–6.
- [160] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, “A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures,” *IEEE Access*, vol. 7, pp. 82721–82743, 2019.
- [161] L. Babun, K. Denney, Z. Berkay Celik, P. Mcdaniel, and A. S. Uluagac, “A Survey on IoT Platforms: Communication, Security, and Privacy Perspectives,” 2021.

- 
- [162] “Hacking the IoT with MQTT | by Victor Pasknel | Morplus Labs.” , Online: <https://morpluslabs.com/hacking-the-iot-with-mqtt-8edaf0d07b9b>.
- [163] J. Y. Lee and J. Lee, “Current Research Trends in IoT Security: A Systematic Mapping Study,” *Mobile Information Systems*, vol. 2021, pp. 1–25, Mar. 2021.
- [164] “2021 IoT spending to grow 24%, led by investments in IoT software.” , Online: <https://iot-analytics.com/2021-global-iot-spending-grow-24-percent/>.
- [165] A. Fornés-Leal *et al.*, “D3.5 – ASSIST-IoT Architecture Definition – Initial,” Apr. 2021.
- [166] A. Chaudhari and R. Bansode, “Survey on Securing Iot Data using Homomorphic Encryption Scheme,” 2021.
- [167] E. L. C. Macedo *et al.*, “On the Security Aspects of Internet of Things: A Systematic Literature Review,” *JOURNAL OF COMMUNICATIONS AND NETWORKS*, vol. 21, no. 5, 2019.
- [168] O. L. ´ Opez *et al.*, “DEVSECOPS METHODOLOGY FOR NG-IOT ECOSYSTEM DEVELOPMENT LIFECYCLE – ASSIST-IOT PERSPECTIVE,” *Journal of Computer Science and Cybernetics*, vol. 37, no. 3, pp. 321–337, Sep. 2021.
- [169] “AIOTI Strategic Foresight Through Digital Leadership: IoT and Edge Computing Convergence – AIOTI.” , Online: <https://aioti.eu/aioti-strategic-foresight-through-digital-leadership-iot-and-edge-computing-convergence/>.
- [170] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge Intelligence: Paving the Last Mile of Artificial Intelligence with Edge Computing.”
- [171] “A maturing DevSecOps landscape.”
- [172] M. Paprzycki *et al.*, “DEVSECOPS METHODOLOGY FOR NG-IOT ECOSYSTEM DEVELOPMENT LIFECYCLE – ASSIST-IOT PERSPECTIVE,” *Journal of Computer Science and Cybernetics*, vol. 37, no. 3, pp. 321–337, Sep. 2021.
- [173] K. L. Beck *et al.*, “Manifesto for Agile Software Development,” 2013.
- [174] L. Leite, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles, “A Survey of DevOps Concepts and Challenges,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, Nov. 2019.
- [175] “Six Reasons It Matters and How to Get There DevOps in the Internet of Things WHEN IT MATTERS, IT RUNS ON WIND RIVER.”
- [176] “How to implement security by design for IoT — ENISA.” , Online: <https://www.enisa.europa.eu/news/enisa-news/how-to-implement-security-by-design-for-iot>.

- [177] R. Kumar and R. Goyal, “Modeling Continuous Security: A Conceptual Model for Automated DevSecOps using Open-source software over Cloud (ADOC),” *Comput Secur*, vol. 97, p. 101967, Aug. 2020.
- [178] “Secure IoT – IoT Security Foundation.” , Online: <https://www.iotsecurityfoundation.org/best-practice-guidelines/>.
- [179] “Red Team VS Blue Team: What’s The Difference? | PurpleSec.” , Online: <https://purplesec.us/red-team-vs-blue-team-cyber-security/>.
- [180] I. Policy and T. Watch Division, “The Tactile Internet,” 2014.
- [181] G. Fettweis, “The Tactile Internet: Applications and Challenges,” *Vehicular Technology Magazine, IEEE*, vol. 9, pp. 64–70, Aug. 2014.
- [182] “Standardisation – AIOTI.” , Online: <https://aioti.eu/about-us/our-groups/standardisation/>.
- [183] S. Haddadin, L. Johannsmeier, and F. D. Ledezma, “Tactile Robots as a Central Embodiment of the Tactile Internet,” *Proceedings of the IEEE*, vol. 107, pp. 471–487, 2019.
- [184] A. Aijaz and M. Sooriyabandara, “The tactile internet for industries: A review,” *Proceedings of the IEEE*, vol. 107, no. 2, pp. 414–435, Feb. 2019.
- [185] D. van den Berg *et al.*, “Challenges in Haptic Communications Over the Tactile Internet.”
- [186] N. Promwongsa *et al.*, “A Comprehensive Survey of the Tactile Internet: State of the art and Research Directions,” Sep. 2020.
- [187] X. Wei, Q. Duan, and L. Zhou, “A QoE-Driven Tactile Internet Architecture for Smart City,” *IEEE Netw*, vol. 34, no. 1, pp. 130–136, Jan. 2020.
- [188] B. Neha, S. K. Panda, P. K. Sahu, K. S. Sahoo, and A. H. Gandomi, “A Systematic Review on Osmotic Computing,” *ACM Transactions on Internet of Things*, vol. 3, no. 2, Feb. 2022.
- [189] I. Lacalle, M. Á. Llorente, and C. E. Palau, “Towards Environmental Impact Reduction Leveraging IoT Infrastructures: The PIXEL Approach,” in *Internet and Distributed Computing Systems*, 2019, pp. 33–45.
- [190] A. Bennaceur, T. T. Tun, Y. Yu, and B. Nuseibeh, “Requirements Engineering,” in *Handbook of Software Engineering*, Cham: Springer International Publishing, 2019, pp. 51–92.
- [191] S. Huber, C. Mühlroth, C. Zagel, S. Schwarz, and F. Bodendorf, “Agile Innovation Management,” 2016.
- [192] D. Leffingwell and D. Widrig, *Managing Software Requirements: A Use Case Approach*. Addison Wesley Professional, 2003.
- [193] “Volere Requirements.” , Online: <https://www.volere.org/>.

- 
- [194] C. E. Palau *et al.*, “Introduction to Interoperability for Heterogeneous IoT Platforms,” *Internet of Things*, pp. 1–26, 2021.
- [195] “AIOTI – The Alliance for the Internet of Things Innovation.” , Online: <https://aioti.eu/>.
- [196] “Next Generation IoT.” , Online: <https://www.ngiot.eu/>.
- [197] “Free: Jira Software | Atlassian.” , Online: <https://www.atlassian.com/es/software/jira/free>.
- [198] “What is CI/CD? | GitLab.” , Online: <https://about.gitlab.com/topics/ci-cd/>.
- [199] K. Petersen, C. Wohlin, and D. Baca, “The waterfall model in large-scale development,” *Lecture Notes in Business Information Processing*, vol. 32 LNBIP, pp. 386–400, 2009.
- [200] “OWASP Top Ten | OWASP Foundation.” , Online: <https://owasp.org/www-project-top-ten/>.
- [201] A. Sharma, M. Kumar, and S. Agarwal, “A Complete Survey on Software Architectural Styles and Patterns,” in *Procedia Computer Science*, 2015, vol. 70, pp. 16–28.
- [202] M. Richards, “Software Architecture Patterns,” 2015.
- [203] M. Gusev and S. Dustdar, “DEPARTMENT: Internet of Things, People, and Processes BACKGROUND AND BASIC DEFINITIONS.”
- [204] “IoT in Telecommunications: Challenges, Opportunities, Benefits & The Future - DataScienceCentral.com.” , Online: <https://www.datasciencecentral.com/iot-in-telecommunications-challenges-opportunities-benefits-amp/>.
- [205] A. Kamilaris and N. Botteghi, “The penetration of Internet of Things in robotics: Towards a web of robotic things,” *J Ambient Intell Smart Environ*, vol. 12, no. 6, pp. 491–512, Jan. 2020.
- [206] “Containers are Shaping IoT Development | Enterprise Networking Planet.” , Online: <https://www.enterprisenetworkingplanet.com/data-center/containers-are-shaping-iot-development/>.
- [207] “LXC vs Docker: Why Docker is Better in 2022 | UpGuard.” , Online: <https://www.upguard.com/blog/docker-vs-lxc>.
- [208] “On-Premises vs Cloud vs Colocation: Is Hybrid the Best | Telehouse.” , Online: <https://www.telehouse.net/blog/on-premises-vs-cloud-vs-colocation/>.
- [209] “Cloud vs on-premises vs hybrid - which is best? – ServiceMuse.” , Online: <https://servicemuse.com/cloud-vs-on-premises-vs-hybrid/>.
- [210] C. I. Valero, A. Belsa, A. Fornes-Leal, F. Boronat, M. A. Llorente, and M.

- Montesinos, “INTER-Framework: An Interoperability Framework to Support IoT Platform Interoperability,” *Internet of Things*, pp. 167–193, 2021.
- [211] F. Galán Márquez, “NGSIV2 Overview for Developers That Already Know NGSIV1.”
- [212] “GitHub - smart-data-models/data-models: A joint collaboration program to support the adoption of a reference architecture and compatible common data models underpinning a digital market of interoperable and replicable smart solutions.” , Online: <https://github.com/smart-data-models/data-models>.
- [213] “Qué es multitenant y por qué es importante - Platzi.” , Online: <https://platzi.com/blog/multi-tenant-que-es-y-por-que-es-importante/>.
- [214] ISO/IEC/IEEE 42010, “ISO/IEC/IEEE 42010 - Systems and software engineering - Architecture description,” 2011.
- [215] N. Rozanski and E. Woods, *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. Addison Wesley, 2011.
- [216] P. Sethi and S. R. Sarangi, “Internet of Things: Architectures, Protocols, and Applications,” 2017.
- [217] “¿Qué es el servicio FHIR? | Microsoft Docs.” , Online: <https://docs.microsoft.com/es-es/azure/healthcare-apis/fhir/overview>.
- [218] “Inside Geo IoT: Intelligent geolocation of assets and people - Geoawesomeness.” , Online: <https://geoawesomeness.com/inside-geo-iot-intelligent-geolocation-assets-people/>.
- [219] “Overview of the IoT Connector - IBM Documentation.” , Online: <https://www.ibm.com/docs/en/mam/7.6.1.2?topic=platforms-overview>.
- [220] “Edgent Incubation Status - Apache Incubator.” , Online: <https://incubator.apache.org/projects/edgent.html>.
- [221] “Introducción a Azure IoT Edge | Microsoft Docs.” , Online: <https://docs.microsoft.com/es-es/azure/iot-edge/about-iot-edge?view=iotedge-2020-11>.
- [222] “MQTT IoT Connector for Industrial Internet of Things - IoT applications.” , Online: <https://openautomationsoftware.com/products/communications/mqtt-connector/>.
- [223] “ODINS IoT Connector.” , Online: <https://www.odins.es/project/iot-connector/>.
- [224] “Paving new ways for next-gen IoT - Neways.” , Online: <https://newayselectronics.com/paving-new-ways-for-next-gen-iot/>.

- 
- [225] “Universal IoT Connector | UIC.” , Online: <https://uic.sget.org/>.
- [226] “D3.5 Data Processing Services - DataPorts.”
- [227] S. J. D. Cox, “Ontology for observations and sampling features, with alignments to existing models,” *Semant Web*, vol. 8, no. 3, pp. 453–470, 2017.
- [228] “The SOSA/SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Actuation, and Sampling | www.semantic-web-journal.net.” , Online: <https://www.semantic-web-journal.net/content/sosassn-ontology-joint-w3c-and-ogc-standard-specifying-semantics-sensors-observations>.
- [229] “SAREF Portal.” , Online: <https://saref.etsi.org/>.
- [230] “EU Regulation on electronic freight transport information (eFTI).”
- [231] “SAREF4EHAW: an extension of SAREF for eHealth Ageing Well domain.” , Online: <https://saref.etsi.org/saref4ehaw/v1.1.1/>.
- [232] “SAREF4AGRI: an extension of SAREF for the agriculture and food domain.” , Online: <https://saref.etsi.org/saref4agri/v1.1.2/>.
- [233] “Linked Open Vocabularies for Internet of Things (LOV4IoT).” , Online: <http://lov4iot.appspot.com/?p=ontologies>.
- [234] “FIWARE NGSI version 2 Release Candidate – FIWARE.” , Online: <https://www.fiware.org/2016/06/08/fiware-ngsi-version-2-release-candidate/>.
- [235] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szmeja, and K. Wasielewska, “Towards semantic interoperability between internet of things platforms,” *Internet of Things*, vol. 0, no. 9783319612997, pp. 103–127, 2018.
- [236] C. I. Valero *et al.*, “AIoTES: Setting the principles for semantic interoperable and modern IoT-enabled reference architecture for Active and Healthy Ageing ecosystems,” *Comput Commun*, vol. 177, pp. 96–111, Sep. 2021.
- [237] “What is a Data Model? | Center for Data, Analytics and Reporting.” , Online: <https://cedar.princeton.edu/understanding-data/what-data-model>.
- [238] “Smart Data Models – A global program led by.” , Online: <https://smartdatamodels.org/>.
- [239] “pyngsi 2.1.10 on PyPI - Libraries.io.” , Online: <https://libraries.io/pypi/pyngsi>.
- [240] “Node-RED.” , Online: <https://nodered.org/>.
- [241] “Apache NiFi.” , Online: <https://nifi.apache.org/>.
- [242] “Tools & Techniques for scalable data processing in Industrial IoT.” , Online: <https://www.umh.app/post/tools-techniques-for-scalable-data-processing-in-industrial-iot>.

- [243] L. T. T. Nguyen *et al.*, “BMDD: A novel approach for IoT platform (brokerless and microservice architecture, decentralized identity, and dynamic transmission messages),” *PeerJ Comput Sci*, vol. 8, p. e950, Apr. 2022.
- [244] “Everything You Need To Know About Publish/Subscribe | Ably Realtime.” , Online: <https://ably.com/topic/pub-sub>.
- [245] “MQTT.” , Online: <https://www.ibm.com/support/pages/mqtt>.
- [246] “Eclipse Mosquitto.” , Online: <https://mosquitto.org/>.
- [247] “GitHub - moscajs/aedes: Barebone MQTT broker that can run on any stream server, the node way.” , Online: <https://github.com/moscajs/aedes>.
- [248] “Moquette Broker.” , Online: <https://moquette-io.github.io/moquette/>.
- [249] “VerneMQ - A MQTT broker that is scalable, enterprise ready, and open source.” , Online: <https://vernemq.com/>.
- [250] “STOMP.” , Online: <https://stomp.github.io/>.
- [251] “The History of IoT: a Comprehensive Timeline of Major Events, Infographic.” , Online: <https://hqsoftwarelab.com/blog/the-history-of-iot-a-comprehensive-timeline-of-major-events-infographic/>.
- [252] “Apache Kafka.” , Online: <https://kafka.apache.org/>.
- [253] “Messaging that just works — RabbitMQ.” , Online: <https://www.rabbitmq.com/>.
- [254] “ActiveMQ.” , Online: <https://activemq.apache.org/>.
- [255] “HiveMQ - Enterprise ready MQTT to move your IoT data.” , Online: <https://www.hivemq.com/>.
- [256] “ZeroMQ.” , Online: <https://zeromq.org/>.
- [257] “Apache ZooKeeper.” , Online: <https://zookeeper.apache.org/>.
- [258] “What Is an X.509 Certificate? - SSL.com.” , Online: <https://www.ssl.com/faqs/what-is-an-x-509-certificate/>.
- [259] “What is Data Brokering? - Definition from Techopedia.” , Online: <https://www.techopedia.com/definition/34111/data-brokering>.
- [260] “What’s new - Cumulocity IoT Guides.” , Online: <https://cumulocity.com/guides/release-10-10-0/whatsnew-10-10-0/>.
- [261] “System overview - Yggio Docs.” , Online: <https://docs.ygg.io/system-overview/>.
- [262] “Homepage - LoRa Alliance®.” , Online: <https://lora-alliance.org/>.
- [263] “About Z-Wave Technology - Z-Wave Alliance.” , Online: [https://z-wavealliance.org/about\\_z-wave\\_technology/](https://z-wavealliance.org/about_z-wave_technology/).
- [264] “Ratatosk context broker - Sensative.” , Online:



- <https://sensative.com/yggio/ratatosk-context-broker/>.
- [265] “FIWARE Orion Context Broker.” , Online: <https://fiware-orion.readthedocs.io/en/master/>.
- [266] “FIWARE - Open APIs for Open Minds.” , Online: <https://www.fiware.org/>.
- [267] “DT, Orange, Vodafone top European telcos based on 2021 revenues | Fierce Wireless.” , Online: <https://www.fiercewireless.com/wireless/dt-orange-vodafone-top-european-telcos-based-2021-revenues>.
- [268] “Developers Catalogue – FIWARE.” , Online: <https://www.fiware.org/catalogue/>.
- [269] “Plataforma VLCi - València Ciudad Inteligente.” , Online: <https://smartcity.valencia.es/vlci/plataforma-vlci/>.
- [270] “FIWARE – Smart Cities.” , Online: <https://www.fiware.org/about-us/smart-cities/>.
- [271] “fiware-orion/raspberry\_pi.md at master · telefonicaid/fiware-orion · GitHub.” , Online: [https://github.com/telefonicaid/fiware-orion/blob/master/docker/raspberry\\_pi.md](https://github.com/telefonicaid/fiware-orion/blob/master/docker/raspberry_pi.md).
- [272] “Entity service paths - Fiware-Orion.” , Online: [https://fiware-orion.readthedocs.io/en/1.4.0/user/service\\_path/index.html](https://fiware-orion.readthedocs.io/en/1.4.0/user/service_path/index.html).
- [273] “Top 10 Databases to Use in 2021. MySQL, Oracle, PostgreSQL, Microsoft... | by Md Kamaruzzaman | Towards Data Science.” , Online: <https://towardsdatascience.com/top-10-databases-to-use-in-2021-d7e6a85402ba>.
- [274] “Types of databases.” , Online: <https://www.tutorialspoint.com/Types-of-databases>.
- [275] “¿Qué es el modelo ACID en bases de datos?” , Online: <https://keepcoding.io/blog/que-es-acid-bases-datos/>.
- [276] “A Review of Different Database Types: Relational versus Non-Relational - DATAVERSITY.” , Online: <https://www.dataversity.net/review-pros-cons-different-databases-relational-versus-non-relational/>.
- [277] “MySQL.” , Online: <https://www.mysql.com/>.
- [278] “MariaDB Server.” , Online: <https://mariadb.org/>.
- [279] “Database Services | Oracle España.” , Online: <https://www.oracle.com/es/database/>.
- [280] “Documentación de Microsoft SQL - SQL Server | Microsoft Docs.” , Online: <https://docs.microsoft.com/es-es/sql/?view=sql-server-ver16>.
- [281] “PostgreSQL: The world’s most advanced open source database.” , Online: <https://www.postgresql.org/>.

- [282] “Redis.” , Online: <https://redis.io/>.
- [283] “What is DynamoDB? | DynamoDB, explained.” , Online: <https://www.dynamodbguide.com/what-is-dynamo-db/>.
- [284] “Apache Cassandra | Apache Cassandra Documentation.” , Online: [https://cassandra.apache.org/\\_/index.html](https://cassandra.apache.org/_/index.html).
- [285] “Apache HBase – Apache HBase™ Home.” , Online: <https://hbase.apache.org/>.
- [286] “DataStax | NoSQL Database Built on Apache Cassandra.” , Online: <https://www.datastax.com/>.
- [287] “MongoDB.” , Online: <https://www.mongodb.com/>.
- [288] “Couchbase: The Modern Database for Enterprise Applications.” , Online: <https://www.couchbase.com/>.
- [289] “Apache CouchDB.” , Online: <https://couchdb.apache.org/>.
- [290] “Top 5 Databases to store data of IoT applications - IoTEDU.” , Online: <https://iot4beginners.com/top-5-databases-to-store-iot-data/>.
- [291] “Learn IoT from scratch #7 - Best databases for IoT - DEV Community.” , Online: <https://dev.to/josethz00/learn-iot-from-scratch-7-best-databases-for-iot-5511>.
- [292] “Choosing the right IoT database.” , Online: <https://www.influxdata.com/iot-database/>.
- [293] “Elasticsearch.” , Online: <https://www.elastic.co/elasticsearch/>.
- [294] “Splunk DB Connect | Splunkbase.” , Online: <https://splunkbase.splunk.com/app/2686/>.
- [295] “sphinx - Buscar con Google.” , Online: <https://www.google.com/search?q=sphinx&oq=sphinx&aqs=chrome..69i57j46i512j0i512j46i10i131i433j0i512j46i512j0i512l2j46i512j0i512.887j0j4&sourceid=chrome&ie=UTF-8>.
- [296] “GraphQL | A query language for your API.” , Online: <https://graphql.org/>.
- [297] “ArangoDB, the multi-model database for graph and beyond.” , Online: <https://www.arangodb.com/>.
- [298] “Graph Data Platform | Graph Database Management System | Neo4j.” , Online: <https://neo4j.com/>.
- [299] “Home | OrientDB Community Edition.” , Online: <https://orientdb.org/>.
- [300] “Amazon Timestream – Base de datos de series temporales – Amazon Web Services.” , Online: <https://aws.amazon.com/es/timestream/>.
- [301] “Prometheus - Monitoring system & time series database.” , Online:

- <https://prometheus.io/>.
- [302] “InfluxDB: Open Source Time Series Database | InfluxData.” , Online: <https://www.influxdata.com/>.
- [303] “El ELK Stack: de los creadores de Elasticsearch. | Elastic.” , Online: <https://www.elastic.co/es/what-is/elk-stack>.
- [304] “Kibana: Explora, visualiza y descubre datos | Elastic.” , Online: <https://www.elastic.co/es/kibana/>.
- [305] “Hardware prerequisites | Elastic Cloud Enterprise Reference [3.3] | Elastic.” , Online: <https://www.elastic.co/guide/en/cloud-enterprise/current/ece-hardware-prereq.html>.
- [306] “Production Notes — MongoDB Manual.” , Online: <https://www.mongodb.com/docs/manual/administration/production-notes/#hardware-considerations>.
- [307] “Install & Configure MongoDB on the Raspberry Pi | MongoDB.” , Online: <https://www.mongodb.com/developer/products/mongodb/mongodb-on-raspberry-pi/>.
- [308] “API Walkthrough - Fiware-Orion.” , Online: [https://fiware-orion.readthedocs.io/en/2.1.0/user/walkthrough\\_apiv2/index.html](https://fiware-orion.readthedocs.io/en/2.1.0/user/walkthrough_apiv2/index.html).
- [309] “Binary field type | Elasticsearch Guide [8.3] | Elastic.” , Online: <https://www.elastic.co/guide/en/elasticsearch/reference/current/binary.html>.
- [310] “Appendix C. Binary data and GridFS · MongoDB in Action.” , Online: <https://livebook.manning.com/book/mongodb-in-action/appendix-c/>.
- [311] “Fiware-STH-Comet.” , Online: <https://fiware-sth-comet.readthedocs.io/en/latest/>.
- [312] “fiware-cygnus.” , Online: <https://fiware-cygnus.readthedocs.io/en/latest/>.
- [313] “Welcome to Apache Flume — Apache Flume.” , Online: <https://flume.apache.org/>.
- [314] “16 Best Container Orchestration Tools And Services In 2022.” , Online: <https://devopscube.com/docker-container-clustering-tools/>.
- [315] “Enterprise Kubernetes Management | Rancher.” , Online: <https://rancher.com/>.
- [316] “Red Hat OpenShift facilita la organización de los contenedores.” , Online: <https://www.redhat.com/es/technologies/cloud-computing/openshift>.
- [317] “API Gateway and Service Connectivity Platform | Kong Inc.” , Online: <https://konghq.com/>.
- [318] “KrakenD - Open source API Gateway.” , Online: <https://www.krakend.io/>.

- [319] PIXEL Project, “D6.3 PIXEL data acquisition, information hub and data representation v1,” 2019. , Online: [https://pixel-ports.eu/wp-content/uploads/2020/05/D6.3\\_PIXEL-data-acquisition-information-hub-and-data-representation-v1.pdf](https://pixel-ports.eu/wp-content/uploads/2020/05/D6.3_PIXEL-data-acquisition-information-hub-and-data-representation-v1.pdf).
- [320] “Docker Hub Container Image Library | App Containerization.” , Online: <https://hub.docker.com/>.
- [321] “Crontab en Linux: con ejemplos y herramientas en tiempo real.” , Online: <https://geekflare.com/es/crontab-linux-with-real-time-examples-and-tools/>.
- [322] A. Belsa *et al.*, “INTER-Layer: A Layered Approach for IoT Platform Interoperability,” *Internet of Things*, pp. 95–132, 2021.
- [323] “IoT Platform Companies Landscape 2021/2022.” , Online: <https://iot-analytics.com/iot-platform-companies-landscape/>.
- [324] “ThingsBoard - Open-source IoT Platform.” , Online: <https://thingsboard.io/>.
- [325] “ThingWorx: IIoT Platform Built to Drive Industrial Innovation | PTC.” , Online: <https://www.ptc.com/es/products/thingworx>.
- [326] “IoT Analytics - ThingSpeak Internet of Things.” , Online: <https://thingspeak.com/>.
- [327] “Industry-Trusted Digitalisation Partner | WolkAbout.” , Online: <https://wolkabout.com/>.
- [328] “GitHub - itead/IoTgo: Open source IoT cloud service.” , Online: <https://github.com/itead/IoTgo>.
- [329] “► Enterprise IoT Platform with Free Plan | Kaa.” , Online: <https://www.kaaiot.com/>.
- [330] “Cloud Native Computing Foundation.” , Online: <https://www.cncf.io/>.
- [331] “Cloud vs. On-Premise Comparison: Key Differences and Similarities | Spiceworks Tech.” , Online: <https://www.spiceworks.com/tech/cloud/articles/cloud-vs-on-premise-comparison-key-differences-and-similarities/>.
- [332] “Azure IoT: plataforma de Internet de las cosas | Microsoft Azure.” , Online: <https://azure.microsoft.com/es-es/solutions/iot/>.
- [333] “IBM Watson IoT Platform.” , Online: <https://internetofthings.ibmcloud.com/>.
- [334] “Internet de las cosas | Plataforma como servicio | AWS IoT.” , Online: <https://aws.amazon.com/es/iot/>.
- [335] “Introduction - WireCloud.” , Online: <https://wirecloud.readthedocs.io/en/stable/>.

- [336] “Grafana: The open observability platform | Grafana Labs.” , Online: <https://grafana.com/>.
- [337] “Grafana vs. Kibana: The Key Differences to Know | Logz.io.” , Online: <https://logz.io/blog/grafana-vs-kibana/>.
- [338] “freeboard - Dashboards For the Internet Of Things.” , Online: <https://freeboard.io/>.
- [339] “¿Qué es Tableau?” , Online: <https://www.tableau.com/es-es/why-tableau/what-is-tableau>.
- [340] “Qué es Power BI | Microsoft Power BI.” , Online: [https://powerbi.microsoft.com/es-mx/what-is-power-bi/?&ef\\_id=Cj0KCQjw852XBhC6ARIsAJsFPN2X8sxIJDv4FqHH4iDOcFyxX5Ml8VIetG8f3BfFqh\\_EDjrT6h0-iXMaAsx8EALw\\_wcB:G:s&OCID=AIDcmm2x16xx83\\_SEM\\_Cj0KCQjw852XBhC6ARIsAJsFPN2X8sxIJDv4FqHH4iDOcFyxX5Ml8VIetG8f3BfFqh\\_EDjrT6h0-iXMaAsx8EALw\\_wcB:G:s&gclid=Cj0KCQjw852XBhC6ARIsAJsFPN2X8sxIJDv4FqHH4iDOcFyxX5Ml8VIetG8f3BfFqh\\_EDjrT6h0-iXMaAsx8EALw\\_wcB](https://powerbi.microsoft.com/es-mx/what-is-power-bi/?&ef_id=Cj0KCQjw852XBhC6ARIsAJsFPN2X8sxIJDv4FqHH4iDOcFyxX5Ml8VIetG8f3BfFqh_EDjrT6h0-iXMaAsx8EALw_wcB:G:s&OCID=AIDcmm2x16xx83_SEM_Cj0KCQjw852XBhC6ARIsAJsFPN2X8sxIJDv4FqHH4iDOcFyxX5Ml8VIetG8f3BfFqh_EDjrT6h0-iXMaAsx8EALw_wcB:G:s&gclid=Cj0KCQjw852XBhC6ARIsAJsFPN2X8sxIJDv4FqHH4iDOcFyxX5Ml8VIetG8f3BfFqh_EDjrT6h0-iXMaAsx8EALw_wcB).
- [341] “Graphite.” , Online: <https://graphiteapp.org/>.
- [342] “¿Qué es Kibana? | Elastic.” , Online: <https://www.elastic.co/es/what-is/kibana>.
- [343] “MongoDB plugin for Grafana | Grafana Labs.” , Online: <https://grafana.com/grafana/plugins/grafana-mongodb-datasource/>.
- [344] “MongoDB Atlas Database | Multi-Cloud Database Service | MongoDB.” , Online: <https://www.mongodb.com/es/atlas/database>.
- [345] “Welcome! - The Apache HTTP Server Project.” , Online: <https://httpd.apache.org/>.
- [346] “Nginx vs. Apache usage statistics, August 2022.” , Online: <https://w3techs.com/technologies/comparison/ws-apache,ws-nginx>.
- [347] “10 Web Development Frameworks You Must Know About in 2022.” , Online: <https://citrusbug.com/blog/best-web-development-framework>.
- [348] “Vue.js.” , Online: <https://vuejs.org/>.
- [349] “MVC - Glosario | MDN.” , Online: <https://developer.mozilla.org/es/docs/Glossary/MVC>.
- [350] “Control de acceso HTTP (CORS) - HTTP | MDN.” , Online: <https://developer.mozilla.org/es/docs/Web/HTTP/CORS>.
- [351] “Welcome to Flask — Flask Documentation (2.1.x).” , Online: <https://flask.palletsprojects.com/en/2.1.x/>.

- [352] “Element - A Desktop UI Toolkit for Web.” , Online: <https://element.eleme.io/#/en-US>.
- [353] “Leaflet - a JavaScript library for interactive maps.” , Online: <https://leafletjs.com/>.
- [354] “JavaScript Charts & Maps - amCharts.” , Online: <https://www.amcharts.com/>.
- [355] “Kibana: Explore, Visualize, Discover Data.” , Online: <https://www.elastic.co/kibana>.
- [356] “Grafana: The open observability platforms.” , Online: <https://grafana.com/>.
- [357] “Let’s Encrypt - Certificados SSL/TLS Gratuitos.” , Online: <https://letsencrypt.org/es/>.
- [358] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, “Scalable and Secure Architecture for Distributed IoT Systems,” *2020 IEEE Technology and Engineering Management Conference, TEMSCON 2020*, Jun. 2020.
- [359] R. Boutaba and I. Aib, “Policy-based Management,” *Journal of Network and Systems Management*, vol. 15, no. 4, pp. 447–480, Dec. 2007.
- [360] “OAuth 2.0 — OAuth.” , Online: <https://oauth.net/2/>.
- [361] “What is OAuth 2.0 and what does it do for you? - Auth0.” , Online: <https://auth0.com/intro-to-iam/what-is-oauth-2/>.
- [362] E. Rissanen, “eXtensible Access Control Markup Language (XACML) Version 3.0,” Jan. 2013.
- [363] “eXtensible Access Control Markup Language (XACML) Version 3.0.” , Online: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>.
- [364] “Policy Based Management System - LDAP Wiki,” <https://ldapwiki.com/wiki/Policy%20Based%20Management%20System>. .
- [365] “API Manager - On-Premise and in the Cloud.” , Online: <https://wso2.com/api-manager/>.
- [366] “Pros and Cons of using an Enterprise Service Bus in Digital Transformation.” , Online: <https://www.cetrixcloudservices.com/blog/enterprise-integration-pros-and-cons-of-enterprise-service-bus>.
- [367] “Konga - More than just another GUI to Kong Admin API.” , Online: <https://pantsel.github.io/konga/>.
- [368] “Gestión de APIs con Apigee | Google Cloud.” , Online: <https://cloud.google.com/apigee?hl=es>.
- [369] “APIInf - API management — APIInf API Management 0.0.1 documentation.”

- , Online: <https://apinf-fiware.readthedocs.io/en/latest/>.
- [370] “Identity Manager - Keyrock - Fiware-IdM.” , Online: <https://fiware-idm.readthedocs.io/en/7.4.0/>.
- [371] “¿Qué es el Single Sign on (SSO)? Características y ventajas.” , Online: <https://www.chakray.com/es/que-es-el-single-sign-on-sso-definicion-caracteristicas-y-ventajas/>.
- [372] “GitHub - authzforce/fiware: FIWARE Generic Enabler for PDP/PAP support functions using an access control schema based on the XACML standard.” , Online: <https://github.com/authzforce/fiware>.
- [373] “What is CRUD? | CRUD Definition | API Glossary.” , Online: <https://rapidapi.com/blog/api-glossary/crud/>.
- [374] “Web Service Definition Language (WSDL).” , Online: <https://www.w3.org/TR/wsdl.html>.
- [375] “Linux.org.” , Online: <https://www.linux.org/>.
- [376] “x64 Architecture - Windows drivers | Microsoft Docs.” , Online: <https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/x64-architecture>.
- [377] “Arm Architecture – Arm®.” , Online: <https://www.arm.com/architecture>.
- [378] “Communication from the Commission Structural indicators - 2002.” .
- [379] M. Nelson Edberg, “Composite Indicators as a Tool for Better Understanding of Socio-Economic Development.”
- [380] M. Mascherini, “Development and use of composite Development and use of composite indicators.”
- [381] L. Berges-Muro, D. Galar, P. Sandborn, and U. Kumar, “The need for aggregated indicators in performance asset management,” *Eksploatacja i Niezawodnosc - Maintenance and Reliability*, vol. 16, pp. 120–127, Aug. 2014.
- [382] M. Nardo, M. Saisana, A. Saltelli, and S. Tarantola, “Tools for composite indicators building. European Commission Joint Research Centre,” *Institute for the Protection and the Security of the Citizen, Econometrics and Statistical Support to Antifraud Unit, I-21020 Ispra (VA) Italy, Report number: EUR*, vol. 21682, p. 140, 2005.
- [383] M. Saisana, “Composite indicators: a review,” in *Second workshop on composite indicators of country performance*, 2004.
- [384] M. Nardo, M. Saisana, A. Saltelli, and S. Tarantola, “Tools for Composite Indicators Building.” Aug-2005.
- [385] Oecd and Ocde, “Handbook on Constructing Composite Indicators: Methodology and UserGuide.”

- [386] D. Roman, T. Tarasova, and J. Paniagua, “MethOSM: a methodology for computing composite indicators derived from OpenStreetMap data,” *JOURNAL OF SPATIAL INFORMATION SCIENCE* Number, vol. 19, pp. 3–27, 2019.
- [387] R. Bandura, “A survey of composite indices measuring country performance: 2008 update,” *New York: United Nations Development Programme, Office of Development Studies (UNDP/ODS Working Paper)*, p. 96, 2008.
- [388] L. Pompei *et al.*, “Composite Indicators for Smart Campus: Data Analysis Method,” *Proceedings - 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe, IEEEIC/I and CPS Europe 2018*, Oct. 2018.
- [389] M. Alam, J. Dupras, and C. Messier, “A framework towards a composite indicator for urban ecosystem services,” *Ecol Indic*, vol. 60, pp. 38–44, 2016.
- [390] M. Amadi *et al.*, *2001 Environmental Sustainability Index*. Davos, Switzerland: An Initiative of the Global Leaders of Tomorrow Environment Task Force, World Economic Forum Annual Meeting 2001, 2001.
- [391] PIXEL Project, “D5.1 Environmental factors and mapping to pilots,” 2019. , Online: <https://pixel-ports.eu/wp-content/uploads/2020/05/D5.1-Environmental-aspects-and-mapping-to-pilots.pdf>.
- [392] S. Tarantola, M. Saisana, and A. Saltelli, “Internal Market Index 2002: Technical Details of the Methodology. European Commission: Joint Research Centre,” *Institute for the Protection and Security of the Citizen Technological and Economic Risk Management Unit I-21020 Ispra (VA) Italy*, 2002.
- [393] “Economic sentiment indicator - Eurostat.” , Online: <https://ec.europa.eu/eurostat/web/products-datasets/product?code=teibs010>.
- [394] “Human Development Index (HDI) | Human Development Reports.” , Online: <http://hdr.undp.org/en/content/human-development-index-hdi>.
- [395] P. Conceição, *Human Development Report 2019. Beyond income, beyond averages, beyond today: Inequalities in human development in the 21st century*. United Nations Development Programme, 2019.
- [396] J. Fagerberg, “Europe at the Crossroads: The Challenge from Innovation-Based Growth,” in *Centre for Technology, Innovation and Culture, University of Oslo, Working Papers on Innovation Studies*, Oslo, Norway: University of Oslo, 2001, pp. 45–60.
- [397] S. Dutta and B. Lanvin, “How digital technologies can make the post-COVID world more equal Shaping the Global Recovery The Network Readiness Index 2021.”



- [398] “Composite Indicators & Scoreboards Explorer.” , Online: <https://composite-indicators.jrc.ec.europa.eu/explorer>.
- [399] B. COLLEN, J. LOH, S. WHITMEE, L. McRAE, R. AMIN, and J. E. M. BAILLIE, “Monitoring Change in Vertebrate Abundance: the Living Planet Index,” *Conservation Biology*, vol. 23, no. 2, pp. 317–327, Apr. 2009.
- [400] “OpenStreetMap.” , Online: <https://www.openstreetmap.org/#map=6/40.007/-2.488>.
- [401] C. Botella-Mascarell, J. Perez, J. Soria, and S. Roger, “On the Use of Composite Indicators for Mobile Communications Network Management in Smart Sustainable Cities,” *Applied Sciences 2021, Vol. 11, Page 181*, vol. 11, no. 1, p. 181, Dec. 2020.
- [402] A. Molavi, G. J. Lim, and B. Race, “A framework for building a smart port and smart port index,” *Int J Sustain Transp*, vol. 14, no. 9, pp. 686–700, 2020.
- [403] M. D. Serrano, F. J. B. Peral, F. M. G. Casas, and M. G. Lozano, “Una revisión crítica para la construcción de indicadores sintéticos // A Critical Review to Construct Composite Indicators,” *Revista de Métodos Cuantitativos para la Economía y la Empresa*, vol. 11, pp. 41–70, 2011.
- [404] W. Becker, M. Domínguez-Torreiro, A. R. Neves, C. Tacão Moura, and M. Saisana, “Exploring the link between Asia and Europe connectivity and sustainable development,” *Research in Globalization*, vol. 3, p. 100045, Dec. 2021.
- [405] “Knowledge for policy.” , Online: [https://knowledge4policy.ec.europa.eu/home\\_en](https://knowledge4policy.ec.europa.eu/home_en).
- [406] “2021 - JRC Week on Composite Indicators and Scoreboards | Knowledge for policy.” , Online: [https://knowledge4policy.ec.europa.eu/composite-indicators/2021-jrc-week-composite-indicators-scoreboards\\_en](https://knowledge4policy.ec.europa.eu/composite-indicators/2021-jrc-week-composite-indicators-scoreboards_en).
- [407] M. Nardo, M. Saisana, A. Saltelli, and S. Tarantola, “Tools for Composite Indicators Building.” Aug-2005.
- [408] “COIN | Competence Centre on Composite Indicators and Scoreboards.” , Online: <https://composite-indicators.jrc.ec.europa.eu/>.
- [409] G. Gyomai and E. Guidetti, “OECD SYSTEM OF COMPOSITE LEADING INDICATORS,” 2012.
- [410] “UNITED NATIONS ECONOMIC COMMISSION FOR EUROPE Guidelines on producing leading, composite and sentiment indicators,” 2019.
- [411] “Your 10-Step Pocket Guide to Composite Indicators & Scoreboards | Knowledge for policy.” , Online: [https://knowledge4policy.ec.europa.eu/publication/your-10-step-pocket-guide-composite-indicators-scoreboards\\_en](https://knowledge4policy.ec.europa.eu/publication/your-10-step-pocket-guide-composite-indicators-scoreboards_en).

- [412] A. Sharpe, “Literature Review of Frameworks for Macro-indicators,” Aug. 2004.
- [413] M. Mazziotta and A. Pareto, “Synthesis of Indicators: The Composite Indicators Approach,” in *Complexity in Society: From Indicators Construction to their Synthesis*, 2017, pp. 159–191.
- [414] M. Mazziotta and A. Pareto, “METHODS FOR CONSTRUCTING COMPOSITE INDICES: ONE FOR ALL OR ALL FOR ONE? 1,” 2013.
- [415] S. M. Fischer, M. K. Joy, W. Abrahamse, T. L. Milfont, and L. M. Petherick, “The Use and Misuse of Composite Environmental Indices,” *bioRxiv*, p. 2022.03.15.484501, Mar. 2022.
- [416] F. Panella, “Selection of Indicators,” *18th JRC Annual training on Composite Indicators and Scoreboards*. Dec-2021.
- [417] M. Dominguez-Torreiro, “Data treatment,” *18th JRC Annual training on Composite Indicators and Scoreboards*. Dec-2021.
- [418] “Measures of Skewness and Kurtosis.” , Online: <https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm>.
- [419] “Identifying outliers with the 1.5xIQR rule (article) | Khan Academy.” , Online: <https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/box-whisker-plots/a/identifying-outliers-iqr-rule>.
- [420] R. Little and D. Rubin, *Statistical Analysis with Missing Data, Third Edition*. Wiley, 2019.
- [421] C. E. Palau *et al.*, “D5.2. PEI Definition and Algorithms v1,” Sep. 2021.
- [422] C. E. Palau *et al.*, “D5.3 - PEI Definition and Algorithms v2,” Apr. 2020.
- [423] P. de Pedraza, “Normalisation,” *18th JRC Annual training on Composite Indicators and Scoreboards*. Dec-2021.
- [424] X. Gan *et al.*, “When to use what: Methods for weighting and aggregating sustainability indicators,” *Ecol Indic*, vol. 81, pp. 491–502, 2017.
- [425] M. Kovacic, “Weighting methods,” *18th JRC Annual training on Composite Indicators and Scoreboards*. Dec-2021.
- [426] M. C. Wilson and J. Wu, “The problems of weak sustainability and associated indicators,” *International Journal of Sustainable Development & World Ecology*, vol. 24, no. 1, pp. 44–51, 2017.
- [427] F. Vidoli and E. Fusco, “Compind: Composite indicators functions based on frontiers in R (Compind package version 2.0),” 2018.
- [428] X. Gan *et al.*, “When to use what: Methods for weighting and aggregating sustainability indicators,” *Ecol Indic*, vol. 81, pp. 491–502, Oct. 2017.

- [429] L. Cherchye, W. Moesen, N. Rogge, and T. van Puyenbroeck, “An introduction to ‘benefit of the doubt’ composite indicators.”
- [430] R. W. Saaty, “The analytic hierarchy process—what it is and how it is used,” *Mathematical Modelling*, vol. 9, no. 3–5, pp. 161–176, Jan. 1987.
- [431] G. Caperna, “Aggregation,” *18th JRC Annual training on Composite Indicators and Scoreboards*. Dec-2021.
- [432] G. Munda, “Multiple criteria decision analysis and sustainable development,” *International Series in Operations Research and Management Science*, vol. 78, pp. 953–986, 2005.
- [433] “A Comparison of Compensatory and Non- Compensatory Decision Making Strategies in IT Project Portfolio Management | Semantic Scholar.”, Online: <https://www.semanticscholar.org/paper/A-Comparison-of-Compensatory-and-Non-Compensatory-Lee-Anderson/7247acc6cb496f24bc9d96fbfdd7c632661e97bc>.
- [434] V. Montalto, “Data sensemaking,” *18th JRC Annual training on Composite Indicators and Scoreboards*. Dec-2021.
- [435] E. Papadimitriou, “Statistical coherence,” *18th JRC Annual training on Composite Indicators and Scoreboards*. Dec-2021.
- [436] V. Alberti, “Communication,” *18th JRC Annual training on Composite Indicators and Scoreboards*. Dec-2021.
- [437] “SCADA. ¿Qué es y qué permite hacer? - Sothis.”, Online: <https://www.sothis.tech/scada-que-es-y-que-permite-hacer/>.
- [438] “Community: Cyclical Analysis and Composite Indi... |OECD.”, Online: <https://community.oecd.org/community/cacis>.
- [439] “EViews.com.”, Online: <https://eviews.com/home.html>.
- [440] I. Global Inc, “EViews 12 User’s Guide II,” 1994.
- [441] “GitHub - Geogouz/Composite-Index-Modeler: CIM.”, Online: <https://github.com/Geogouz/Composite-Index-Modeler>.
- [442] “CITYkeys Project.”, Online: <http://www.citykeys-project.eu/citykeys/home>.
- [443] P. Bosch, S. Jongeneel, V. Rovers, H.-M. Neumann, M. Airaksinen, and A. Huovila, “CITYkeys D1-4 Indicators for smart city projects and smart cities,” 2017., Online: <http://nws.eurocities.eu/MediaShell/media/CITYkeysD14Indicatorsforsmartcityprojectsandsmartcities.pdf>.
- [444] “Make Healthy Choices Easier Options - Scientific American.”, Online: <https://www.scientificamerican.com/podcast/episode/make-healthy-choices-easier-options-12-09-20/>.

- [445] M. Airaksinen, I. Pinto-Seppä, A. Huovila, H.-M. Neumann, B. Iglar, and P. Bosch, “Smart city performance measurement framework CITYkeys,” in *2017 International Conference on Engineering, Technology and Innovation*, 2018, pp. 718–723.
- [446] “Compind web app.” , Online: [https://fvidoli.shinyapps.io/compind\\_app/](https://fvidoli.shinyapps.io/compind_app/).
- [447] F. Vidoli and E. F. Maintainer, “Package ‘Compind’ Type Package Title Composite Indicators Functions,” 2022.
- [448] D. Lindén, M. Cinelli, M. Spada, W. Becker, and P. Burgherr, “Composite Indicator Analysis and Optimization (CIAO) Tool,” Nov. 2018.
- [449] W. Becker *et al.*, “COIN Tool User Guide, EUR 29899 EN, Publications Office of the European Union,” pp. 1–53, 2019.
- [450] “On Automating and Extending Construction of Business Cycle Composite Indicators - Theses - University of Economics, Prague.” , Online: <https://vskp.vse.cz/english/74860>.
- [451] “PhD-Lenka Vrana / GitHub.” , Online: [https://github.com/LenkaV/CIF/blob/develop/thesis/vrana\\_CIF\\_2019.pdf](https://github.com/LenkaV/CIF/blob/develop/thesis/vrana_CIF_2019.pdf).
- [452] “FIWARE e interoperabilidad para Smart Cities.” , Online: <https://www.apegr.org/images/descargas/J7OctESMARTCITY/2PresentacionFIWARE.pdf>.
- [453] “Composite Indicator Development and Analysis in R with COINr.” , Online: <https://bluefoxr.github.io/COINrDoc/index.html#demo>.
- [454] “KeyPerformanceIndicator - FIWARE Data Models.” , Online: <https://fiware-datamodels.readthedocs.io/en/latest/KeyPerformanceIndicator/doc/spec/index.html>.
- [455] C. Fajardo, O. Kosheleva, and V. Kreinovich, “One Needs to Be Careful When Dismissing Outliers: A Realistic Example,” *Journal of Innovative Technology and Education*, vol. 3, no. 1, pp. 205–214, 2016.
- [456] “Elasticsearch Java API Client [8.3] | Elastic.” , Online: <https://www.elastic.co/guide/en/elasticsearch/client/java-api-client/current/index.html>.
- [457] “MongoDB Java Drivers — MongoDB Drivers.” , Online: <https://www.mongodb.com/docs/drivers/java-drivers/>.
- [458] “Trees in Java | Java Program to Implement a Binary Tree | Edureka.” , Online: <https://www.edureka.co/blog/java-binary-tree>.
- [459] “ObjectAid UML Explorer | Eclipse Plugins, Bundles and Products - Eclipse Marketplace.” , Online: <https://marketplace.eclipse.org/content/objectaid->

- uml-explorer.
- [460] I. Lacalle, A. Belsa, R. Vaño, and C. E. Palau, “Framework and Methodology for Establishing Port-City Policies Based on Real-Time Composite Indicators and IoT: A Practical Use-Case,” *Sensors*, vol. 20, no. 15, p. 4131, Jul. 2020.
- [461] “Exploratory Data Analysis: A Practical Guide and Template for Structured Data | by Jiahao Weng | Towards Data Science.” , Online: <https://towardsdatascience.com/exploratory-data-analysis-eda-a-practical-guide-and-template-for-structured-data-abfbf3ee3bd9>.
- [462] “Entropía de la información en algoritmos de Machine Learning.” , Online: <https://platzi.com/tutoriales/1457-machine-learning/3512-entropia-de-la-informacion-en-algoritmos-de-machine-learning/>.
- [463] “pickle — Python object serialization — Python 3.10.5 documentation.” , Online: <https://docs.python.org/3/library/pickle.html>.
- [464] “Serialization and Deserialization Techniques in Python | by Asha | Medium.” , Online: <https://medium.com/@ashaicy/serialization-and-deserialization-techniques-in-python-deserialization-69beed1ed3ef>.
- [465] “Aprendizaje supervisado y no supervisado - healthdataminer.com.” , Online: <https://healthdataminer.com/data-mining/aprendizaje-supervisado-y-no-supervisado/>.
- [466] S. V. Kumar and L. Vanajakshi, “Short-term traffic flow prediction using seasonal ARIMA model with limited input data,” *European Transport Research Review*, vol. 7, no. 3, p. 21, 2015.
- [467] “Prophet: forecastig at scale.” , Online: <https://facebook.github.io/prophet/>.
- [468] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, “Semi-supervised Learning with Deep Generative Models.”
- [469] “pandas - Python Data Analysis Library.” , Online: <https://pandas.pydata.org/>.
- [470] “RDS Serialization Prophet - Facebook.” , Online: [https://facebook.github.io/prophet/docs/additional\\_topics.html#saving-models](https://facebook.github.io/prophet/docs/additional_topics.html#saving-models).
- [471] Q. Li *et al.*, “A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection,” Jul. 2019.
- [472] D. J. Beutel *et al.*, “Flower: A Friendly Federated Learning Research Framework,” Jul. 2020.
- [473] S. Xuan *et al.*, “An incentive mechanism for data sharing based on blockchain with smart contracts,” *Computers & Electrical Engineering*, vol. 83, p. 106587, May 2020.
- [474] “¿Qué Es Un Hash Y Cómo Funciona? | Blog oficial de Kaspersky.” , Online:

- <https://latam.kaspersky.com/blog/que-es-un-hash-y-como-funciona/2806/>.
- [475] “Manage data in Docker | Docker Documentation.” , Online: <https://docs.docker.com/storage/>.
- [476] “GitHub - j-easy/easy-rules: The simple, stupid rules engine for Java.” , Online: <https://github.com/j-easy/easy-rules>.
- [477] “Creating PDF Files in Java | Baeldung.” , Online: <https://www.baeldung.com/java-pdf-creation>.
- [478] “Overview of Docker Compose | Docker Documentation.” , Online: <https://docs.docker.com/compose/>.
- [479] L. Garcia-Alonso, T. G. Z. Moura, and D. Roibas, “The effect of weather conditions on port technical efficiency,” *Mar Policy*, vol. 113, p. 103816, Mar. 2020.
- [480] P. Shang, R. Li, Z. Liu, and X. Li, “Inclement Weather Impacts on Urban Traffic Conditions,” *15th COTA International Conference of Transportation Professionals Chinese Overseas Transportation Association (COTA) Beijing Jiaotong University Transportation Research Board Institute of Transportation Engineers (ITE) American Society of Civil Engineers*;, pp. 2213–2227, 2015.
- [481] A. Koźlak and D. Wach, “Causes of traffic congestion in urban areas. Case of Poland,” *SHS Web of Conferences*, vol. 57, p. 01019, Nov. 2018.
- [482] “TrafficThess - LIVE Traffic in Thessaloniki, Greece,” <https://www.traffidthess.imet.gr/>, 10-Jul-2022. .
- [483] “National Observatory of Athens - Meteo - Stations’ live data and database.” , Online: <http://stratus.meteo.noa.gr/front>.
- [484] PIXEL Project, “D5.2 PEI Definition and Algorithms v1,” 2019. , Online: <https://pixel-ports.eu/wp-content/uploads/2020/05/D5.2-PEI-Definition-and-Algorithms-v1.pdf>.
- [485] M. Široka, S. Piličić, T. Milošević, I. Lacalle, and L. Traven, “A novel approach for assessing the ports’ environmental impacts in real time – The IoT based port environmental index,” *Ecol Indic*, vol. 120, p. 106949, Jan. 2021.
- [486] ESPO, *ESPO Environmental Report – EcoPorts in Sights 2019*. 2019.
- [487] “Review of Maritime Transport 2021 | UNCTAD.” , Online: <https://unctad.org/es/node/35541>.
- [488] “Port Performance I Port Performance Indicators Selection and Measurement indicators Selection and PPRISM Port Performance Indicators: Selection and Measurement Project Executive report (PPRISM WP4 D4.2).”

- [489] “PORTOPIA Project – World Port Sustainability Program.” , Online: <https://sustainableworldports.org/project/portopia-project/>.
- [490] A. Micallef and A. Williams, Eds., *Beach Management*. Routledge, 2009.
- [491] M. Puig, C. Wooldridge, A. Michail, and R. M. Darbra, “Current status and trends of the environmental performance in European ports,” *Environ Sci Policy*, vol. 48, pp. 57–66, Apr. 2015.
- [492] E. Peris-Mora, J. M. D. Orejas, A. Subirats, S. Ibáñez, and P. Alvarez, “Development of a system of indicators for sustainable port management,” *Mar Pollut Bull*, vol. 50, no. 12, pp. 1649–1660, Dec. 2005.
- [493] M. Puig, “Methodology for the selection and implementation of environmental aspects and performance indicators in ports,” Universitat Politècnica de Catalunya, 2016.
- [494] “European Port Industry Sustainability Report 2016.”
- [495] O. le Brun, D. Mille, I. Kegalj, T. Milosevic, S. Pilicic, and L. Traven, “D5.1 – Environmental factors and mapping to pilots,” Sep. 2021.
- [496] “International Convention for the Prevention of Pollution from Ships (MARPOL).” , Online: [https://www.imo.org/en/About/Conventions/Pages/International-Convention-for-the-Prevention-of-Pollution-from-Ships-\(MARPOL\).aspx](https://www.imo.org/en/About/Conventions/Pages/International-Convention-for-the-Prevention-of-Pollution-from-Ships-(MARPOL).aspx).
- [497] “COMMISSION OF THE EUROPEAN COMMUNITIES EUROSTAT Directorate E: Sectoral and regional statistics Unit E-3: Environmental and forestry statistics Guidance on classification of waste according to EWC-Stat categories Supplement to the Manual for the Implementation of the Regulation (EC) No 2150/2002 on Waste Statistics,” 2010.
- [498] “HOP - Smart Cities.” , Online: <https://smartcities.hopu.eu/smart-spot.html>.
- [499] “OPT3001 data sheet, product information and support | TI.com.” , Online: <https://www.ti.com/product/OPT3001>.
- [500] W. Bourgeois, A.-C. Romain, J. Nicolas, and R. M. Stuetz, “The use of sensor arrays for environmental monitoring: interests and limitations,” *Journal of Environmental Monitoring*, vol. 5, no. 6, p. 852, 2003.
- [501] “Predictor-LimA - Softnoise GmbH.” , Online: <https://softnoise.com/products/predictor-lima/>.
- [502] M. Jović, E. Tijan, D. Brčić, and A. Pucihar, “Digitalization in Maritime Transport and Seaports: Bibliometric, Content and Thematic Analysis,” *J Mar Sci Eng*, vol. 10, no. 4, p. 486, Apr. 2022.
- [503] “European Maritime Transport Environmental Report 2021 — European Environment Agency.” , Online:

- <https://www.eea.europa.eu/publications/maritime-transport/>.
- [504] “Lightweight M2M (LWM2M) – OMA SpecWorks.” , Online: <https://omaspecworks.org/what-is-oma-specworks/iot/lightweight-m2m-lwm2m/>.
- [505] “Eclipse Mosquitto.” , Online: <https://mosquitto.org/>.
- [506] “Docker Hub Container Image Library | App Containerization.” , Online: <https://hub.docker.com/>.
- [507] “Home - Operational Tools.” , Online: <https://docs-hub-ot.readthedocs.io/en/latest/>.
- [508] “Webinar 5 - PIXEL Closure event. Demo - YouTube.” , Online: [https://www.youtube.com/watch?v=PC\\_4WYjWvyE](https://www.youtube.com/watch?v=PC_4WYjWvyE).
- [509] “Arduino Uno Rev3 — Arduino Official Store.” , Online: <https://store.arduino.cc/products/arduino-uno-rev3>.
- [510] “ESP-01 ESP8266 WiFi-Serial.” , Online: <https://naylampmechatronics.com/esp8266-wifi-serial.html>.
- [511] “Network Layer | Layer 3 | The OSI-Model.” , Online: <https://osi-model.com/network-layer/>.
- [512] “K3s: Lightweight Kubernetes.” , Online: <https://k3s.io/>.
- [513] “Mobility Package I.” , Online: [https://transport.ec.europa.eu/transport-modes/road/mobility-package-i\\_en](https://transport.ec.europa.eu/transport-modes/road/mobility-package-i_en).
- [514] “AIS vessel tracking | AIS data exchange | JSON/XML ship positions.” , Online: <https://www.aishub.net/>.
- [515] “PIXEL: Energy Demand Model - YouTube.” , Online: <https://www.youtube.com/watch?v=RztRazlI2Gk>.
- [516] E. Simon, C. Garnier, I. Lacalle, J. P. Costa, and C. E. Palau, “Small and medium ports’ activities modelling: Introduction to the pixel approach,” in *WIT Transactions on the Built Environment*, 2019, vol. 187, pp. 149–163.
- [517] “Apache ECharts: An Open Source JavaScript Visualization Library.” , Online: <https://echarts.apache.org/en/index.html>.
- [518] “A Roadmap for the Next-Generation IoT in Europe | Shaping Europe’s digital future.” , Online: <https://digital-strategy.ec.europa.eu/en/news/roadmap-next-generation-iot-europe>.
- [519] “ERC: European Research Council |.” , Online: <https://erc.europa.eu/>.
- [520] “Home | Marie Skłodowska-Curie Actions.” , Online: <https://marie-sklodowska-curie-actions.ec.europa.eu/>.
- [521] “Horizon Europe.” , Online: <https://research-and->



- innovation.ec.europa.eu/funding/funding-opportunities/funding-programmes-and-open-calls/horizon-europe\_en.
- [522] “HORIZON EUROPE - Cliffnotes.” , Online: <https://www.horizon-eu.eu/>.
- [523] “Next Generation EU | Fondos de Recuperación para Europa.” , Online: <https://nexteugeneration.com/>.
- [524] “Actualización de instrumentos financieros públicos para startups.” , Online: [https://www.eldiario.es/canariasahora/impulsa-startups/actualizacion-startups-instrumentos-financieros-impulsa-startups-impulsa-ventures\\_1\\_9120609.html](https://www.eldiario.es/canariasahora/impulsa-startups/actualizacion-startups-instrumentos-financieros-impulsa-startups-impulsa-ventures_1_9120609.html).
- [525] “Centro para el Desarrollo Tecnológico Industrial / Ayudas a la I+D+I / Proyectos de investigación y desarrollo.” , Online: <https://www.cdti.es/index.asp?MP=100&MS=802&MN=2>.
- [526] A.-D. Suciú, A. Tudor, I. Chițu, L. Dovleac, and G. Brătucu, “IoT Technologies as Instruments for SMEs’ Innovation and Sustainable Growth,” *Sustainability*, vol. 13, p. 6357, Aug. 2021.
- [527] A.-I. Consortium, “D9.6 – Business Models and Marketing Operations – First Version,” Apr. 2022.
- [528] “EU funding programmes | European Commission.” , Online: [https://ec.europa.eu/info/funding-tenders/find-funding/eu-funding-programmes\\_en](https://ec.europa.eu/info/funding-tenders/find-funding/eu-funding-programmes_en).
- [529] “The Digital Europe Programme | Shaping Europe’s digital future.” , Online: <https://digital-strategy.ec.europa.eu/en/activities/digital-programme>.
- [530] “EIC Accelerator.” , Online: [https://eic.ec.europa.eu/eic-funding-opportunities/eic-accelerator\\_en](https://eic.ec.europa.eu/eic-funding-opportunities/eic-accelerator_en).
- [531] “InvestEU: programa europeo para impulsar la inversión | Noticias | Parlamento Europeo.” , Online: <https://www.europarl.europa.eu/news/es/headlines/eu-affairs/20210225STO98708/investeu-programa-europeo-para-impulsar-la-inversion>.
- [532] “Ayudas NEOTEC - CDTI.” , Online: [https://www.cdti.es/index.asp?MP=4&MS=0&MN=1&TR=C&IDR=3092&r=1920\\*1080](https://www.cdti.es/index.asp?MP=4&MS=0&MN=1&TR=C&IDR=3092&r=1920*1080).
- [533] “Matriz de prioridades - RIS3CV - Generalitat Valenciana.” , Online: <https://ris3cv.gva.es/es/matriz-de-prioridades>.
- [534] “DAFO.” , Online: <http://www.ipyme.org/es-ES/DecisionEmprender/Paginas/DAFO.aspx>.
- [535] “SWOT (Strengths, Weakness, Opportunities, Threats) | Capacity4dev.” , Online: [https://europa.eu/capacity4dev/evaluation\\_guidelines/wiki/swot-](https://europa.eu/capacity4dev/evaluation_guidelines/wiki/swot-)

- strenghts-weakness-opportunities-threats-0.
- [536] “PESTLE Analysis.” , Online: <https://digital-transformation-tool.eu/training/mod/hvp/view.php?id=37>.
- [537] “Business Model CANVAS.” , Online: <https://digital-transformation-tool.eu/training/mod/hvp/view.php?id=45>.
- [538] “Porter’s Five Forces of Competitive Position Analysis.” , Online: <https://www.cgma.org/resources/tools/essential-tools/porters-five-forces.html>.
- [539] “Problem-Solution Fit vs. Product-Market Fit.” , Online: <https://leannovators.tech/articulos/problem-solution-fit-vs-product-market-fit/>.
- [540] “El programa ‘Fit for 55’ de la UE impulsará la innovación política para la acción climática - Spain.” , Online: <https://spain.climate-kic.org/news/el-programa-fit-for-55-de-la-ue-impulsara-la-innovacion-politica-para-la-accion-climatica/>.
- [541] “Quality Education | Composite Indicators & Scoreboards Explorer.” , Online: <https://composite-indicators.jrc.ec.europa.eu/explorer/explorer/categories/16/goal-4-quality-education>.
- [542] M. E. Porter, “Competitive Strategy: Techniques for Analyzing Industries and Competitors.” 1980.
- [543] “What Are Porter’s 5 Forces?” , Online: <https://www.investopedia.com/terms/p/porter.asp>.
- [544] “Horizon Results Booster.” , Online: <https://www.horizonresultsbooster.eu/>.
- [545] “An Easy-to-Understand Guide to Tiered Pricing.” , Online: <https://blog.hubspot.com/sales/tiered-pricing>.
- [546] “Atos Tech Trends Radar 2022.” , Online: <https://atos.net/en/lp/techradar/edge>.
- [547] “Internet of Things (IoT) - Key Business Insights | Gartner.” , Online: <https://www.gartner.com/en/information-technology/insights/internet-of-things>.
- [548] “The 5 Biggest Internet Of Things (IoT) Trends In 2022.” , Online: <https://www.forbes.com/sites/bernardmarr/2021/12/13/the-5-biggest-internet-of-things-iot-trends-in-2022/?sh=59d5218c5aba>.
- [549] “The next generation Internet of Things | Shaping Europe’s digital future.” , Online: <https://digital-strategy.ec.europa.eu/en/policies/next-generation-internet-things>.
- [550] J. Peng, E. Jury, P. Dönnies, and C. Ciurtin, “Machine Learning Techniques

- 
- for Personalised Medicine Approaches in Immune-Mediated Chronic Inflammatory Diseases: Applications and Challenges,” *Front Pharmacol*, vol. 12, Aug. 2021.
- [551] I. H. Sarker, “Machine Learning: Algorithms, Real-World Applications and Research Directions,” *SN Comput Sci*, vol. 2, no. 3, pp. 1–21, May 2021.
- [552] “Linear Regression Analysis in SPSS Statistics - Procedure, assumptions and reporting the output.” , Online: <https://statistics.laerd.com/spss-tutorials/linear-regression-using-spss-statistics.php>.
- [553] P. Sugiartawan, R. Pulungan, and A. Sari, “Prediction by a Hybrid of Wavelet Transform and Long-Short-Term-Memory Neural Network,” *International Journal of Advanced Computer Science and Applications*, vol. 8, Aug. 2017.
- [554] N. Laptev, J. Yosinski, L. Erran Li, and S. Smyl, “Time-series Extreme Event Forecasting with Neural Networks at Uber.”
- [555] “Multicloud Federated Learning.” , Online: <https://acuratio.com/vertical-learning>.



# Apéndices

# Apéndice A – Diagramas de clases del servicio CIC

En las siguientes imágenes se muestran los diagramas de clases de los elementos programáticos más importantes del servicio CIC desarrollado (en Java):

- **Node:** Un nodo del árbol, que posee hijos y padres y cuyo valor se calcula siguiendo los métodos matemáticos correspondientes y en función de su nivel en la estructura global (árbol).
- **Edge:** Representa las relaciones entre los nodos y la direccionalidad de dicha relación.
- **KPI:** Indicadores iniciales (cKPI). Elementos principales de modelado de los datos de entrada.
- **Tree:** La estructura en árbol mencionada, incluyendo los nodos correspondientes y su relación desde los KPIs.
- Vista conjunta de todas las clases incluidas.

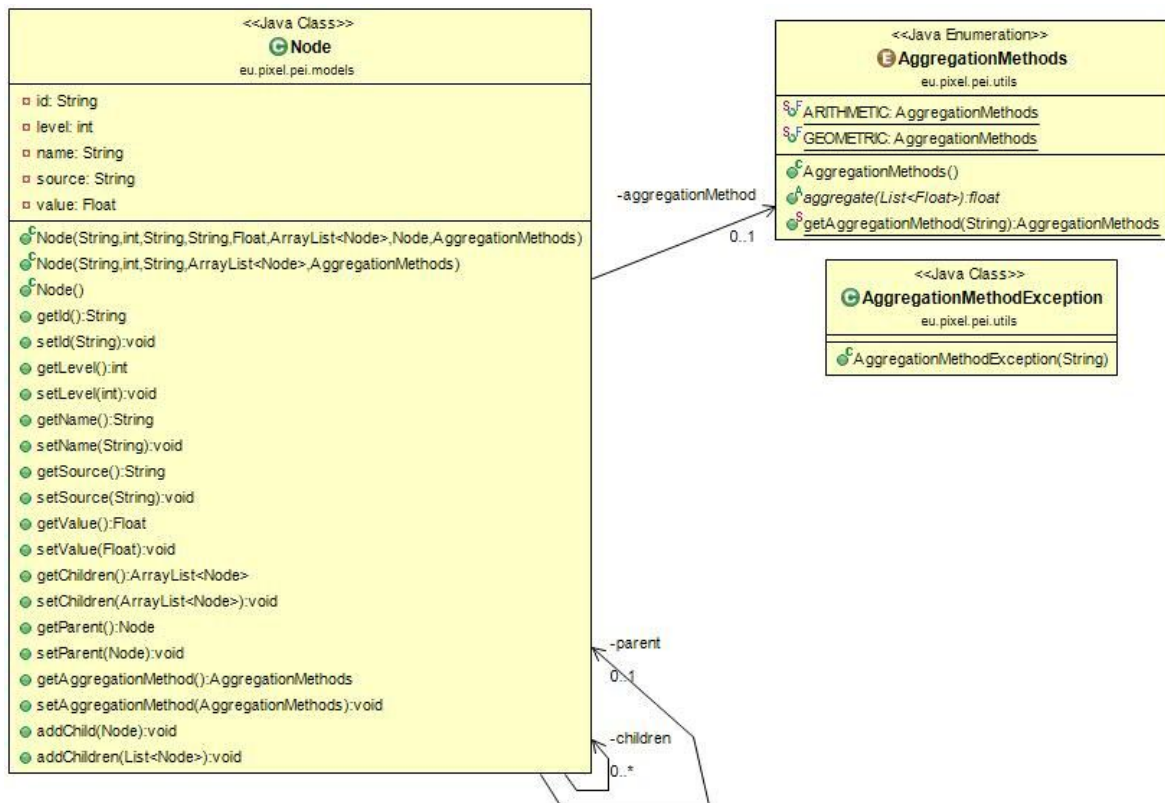


Figura A.1: Diagrama de clases del servicio CIC - Nodee

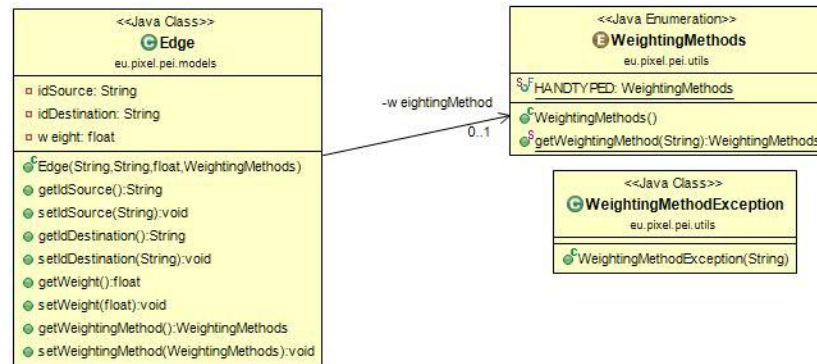


Figura A.2: Diagrama de clases del servicio CIC - Edge

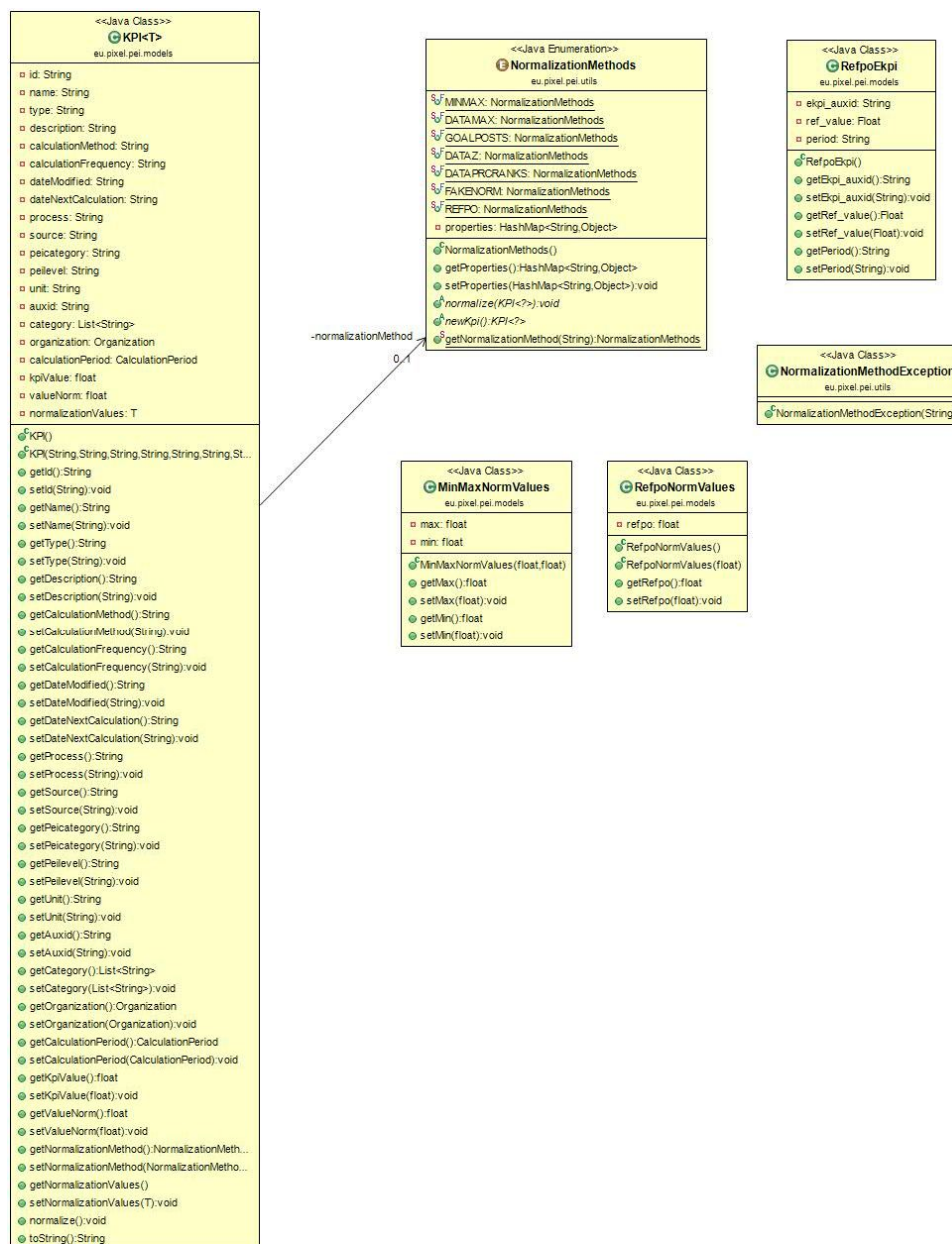


Figura A.3: Diagrama de clases del servicio CIC - KPI

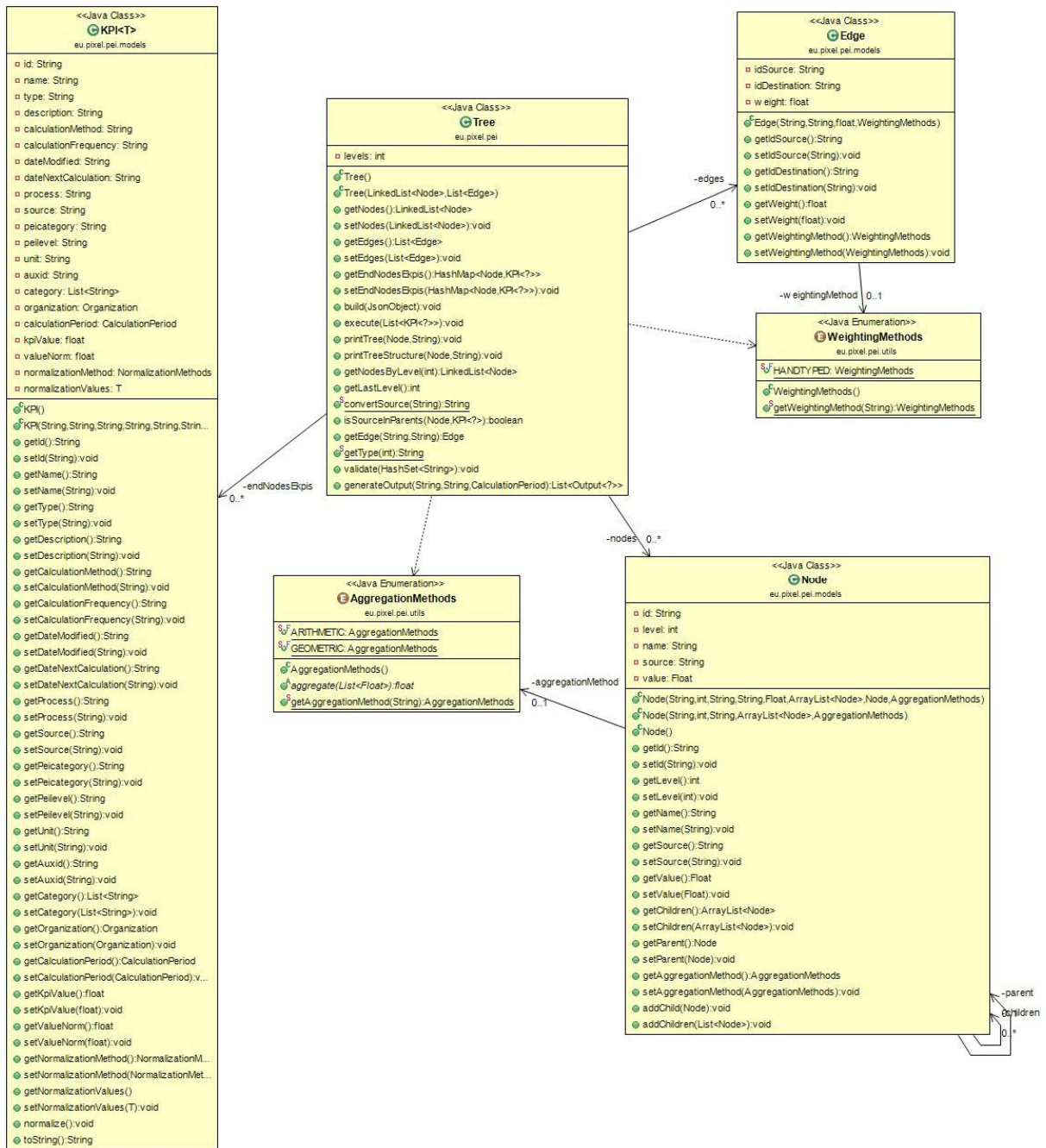


Figura A.4: Diagrama de clases del servicio CIC - Tree





## Apéndice B – Características de las fuentes de datos

A la hora de realizar un índice de fiabilidad asociado al cálculo de un indicador compuesto, se debe considerar el origen de las fuentes de datos que lo alimentan. Para ello, durante la tesis doctoral, el candidato elaboró una serie de tablas comparativas que son adjuntas en este apéndice. Estas tablas tratan de compilar la “bondad” relativa de cada tipo de fuente de datos de cara a alimentar un indicador (*cKPI*) basado en sus puntos débiles y fuertes en diferentes categorías. Tras lo analizado en capítulos del manuscrito y lo observado en las experiencias de validación, estos “tipos de fuentes” fueron resumidos en cinco: sensores, servicios web o API externas, herramientas *proxy* o internas, introducción manual (a través de formularios) y recopilación a partir de valores medios de informes o de estimaciones disponibles en la literatura. Las reflexiones expuestas en la siguiente tabla están fundamentadas en el conocimiento adquirido por el doctorando junto con un soporte de búsqueda de referencias en el estado del arte. Estas indagaciones sirvieron como punto de partida para la definición de algunos pesos en los pasos de ponderación del CI así como para el diseño de las encuestas circulares a los expertos para la realización del método de asignación de presupuesto (BAM).

|   |  |
|---|--|
| <b>Sensores</b>   | Mediciones de ruido, sensor de luz, ocupación de un espacio... |
| <b>Puntos fuertes:</b> Tiempo real, precisión y automatización  |  |
| <p><u>En tiempo real:</u> Se dispondrá de datos más actualizados. Los agentes conectarán con esta información y mantendrán actualizado el módulo de persistencia. Se trata de la fuente de datos ideal para remarcar la orientación IoT de la solución y del cálculo del CI. Sin embargo, hay datos que no son propensos a ser recogidos por los sensores, por lo que el tiempo real no puede ser una característica de evaluación en esos casos.</p> <p><u>Precisión:</u> Los sensores comerciales suelen venir con hojas de datos y especificaciones muy detalladas. En función de las necesidades de precisión de cada <i>cKPI</i> puede elegirse un sensor u otro (si hay varios disponibles). Algunos datos son más sensibles a la precisión, mientras que otros pueden no ser muy críticos o su contribución de peso relativo al CI puede permitir relajar la precisión requerida.</p> <p><u>Automatización:</u> Es el principal punto fuerte de este tipo de fuente de datos. Una vez que el sensor está instalado y los técnicos han sido capaces de integrarlo a través de agentes, el proceso es transparente para el usuario final del CI. Además, los valores automatizados son considerados fiables si se aplican la seguridad y los métodos adecuados, por lo que el resultado final del CI será sensiblemente más confiable.</p> |  |
| <b>Puntos débiles</b> Coste, instalación, disponibilidad, conocimiento requerido.   |  |
| <p><u>Coste:</u> Normalmente, los sensores validados por la industria resultan relativamente costosos. A pesar de que en los escenarios de validación se han planteado se ha conseguido desplegar la solución a bajo coste (prototipo, búsqueda exhaustiva, entornos no productivos), los dispositivos disponibles en el mercado pueden alcanzar cantidades considerables de dinero. Esta reflexión depende enormemente del ámbito de aplicación objetivo de la solución.</p>   |  |

**Instalación:** Una vez adquirido el sensor, es necesario un esfuerzo no trivial por parte de los técnicos para la instalación. Las sesiones *in situ*, las pruebas de disponibilidad, el mantenimiento y el soporte son necesarios si la entidad adoptante de la solución quiere que la fuente de datos sea fiable en el cálculo del CI. De nuevo, estas tareas deben ser realizadas por personal especializado que no siempre está presente en dichas entidades.

**Disponibilidad:** Incluso con una buena instalación, los sensores físicos son siempre susceptibles de sufrir cortes de servicio, problemas de batería, difícil alcance de la red, errores de rango dinámico, mediciones falsas o daños por las condiciones climáticas o eventos inesperados. Todo ello podría convertir a los sensores en una fuente no fiable si no se toman ciertas medidas de contingencia.

**Falta de conocimientos:** Las entidades adoptantes no siempre disponen de personal especializado para realizar la instalación, conexión o mantenimiento. En este sentido, se debería crear un grupo de trabajo con experiencia para cubrir adecuadamente todo el espectro de sensores. Una solución en este punto sería contar con los creadores del producto y realizar una subcontratación para conseguir dicha conexión e integración (relacionado con la sección 6).

|                                      |  |
|--------------------------------------|--|
| <b>Servicios web o APIs externas</b> | Datos AIS, escalas de buques, OpenWeatherMap, HERE, Twitter... |
|--------------------------------------|--|

**Puntos fuertes:** Tiempo (casi) real, automatización, instalación.

**Tiempo casi real:** Normalmente, los servicios web se caracterizan por su alta disponibilidad y un tiempo de respuesta razonablemente bajo. Retrasos en la red, el tiempo de consulta de la base de datos y el reenvío de paquetes implican varias centésimas de segundo, considerado suficiente en caso general para servicios de CI.

**Automatización:** En caso general, basta con ejecutar un cliente HTTP a través de un agente para automatizar esta fuente, una de las características más importantes en el cálculo de CIs. Este enfoque puede ser incluso mejor que la sensorización si se necesita información descriptiva/compleja (por ejemplo, escalas de buques durante una semana), o para conocer el cumplimiento de los límites legales.

**Facilidad de instalación:** Mientras que los sensores necesitan tiempo y esfuerzo para ser instalados, desplegados e integrados correctamente, los servicios web externos sólo necesitan una conexión HTTP y las credenciales adecuadas para acceder a la información. Esta es una de las principales ventajas de este tipo de fuentes de datos.

**Puntos débiles** Heterogeneidad, restricciones de acceso.

**Heterogeneidad:** No existe una metodología estandarizada o una referencia consolidada sobre cómo acceder a los servicios web externos. Aunque muchos servicios ofrecen sus APIS siguiendo convenciones REST y documentándolas con Swagger, esto es todavía un punto débil de este enfoque, ya que cada entidad puede requerir datos de varias webs externas y no se puede crear un agente estándar para gestionar estas conexiones.

**Depende de las políticas de cada entidad:** Según la experiencia adquirida, las políticas de acceso a los servicios externos e internos son bastante heterogéneas y deben ser analizadas individualmente. En algunos casos, los datos son gestionados por diferentes departamentos y no existe interoperabilidad (ni mandato de compartirlos) entre ellos.

**Restricciones de acceso:** En ocasiones, las políticas de seguridad de las entidades no permiten la entrada de datos externos, o deben pasar por determinados controles de software. Este problema puede ser de aplicación con esta fuente de datos siempre que dicha fuente no sea gestionada por la propia entidad.

|   |   |
|---|---|
| <b>Herramientas proxy o internas</b>  | Factor de emisión de CO2 a partir de una simulación (como el servicio PAS), información resultante de la aplicación de motor de reglas... |
| <b>Puntos fuertes:</b>  |   |
| <p><u>Automatización y conocimiento del procedimiento:</u> Si se trata de servicios internos gestionados desde el propio sistema, suele bastar con definir qué campos y en qué condiciones han de ser accedido y configurar los índices/colecciones apropiadamente a la hora de instanciar los servicios. Al ser un servicio que (probablemente) ha sido instalado/desarrollado por el mismo equipo, el conocimiento de la información y provista y cómo ha sido generada es conocido.</p>  |   |
| <b>Puntos débiles</b> Heterogeneidad, dependencia a terceros, confiabilidad.  |   |
| <p><u>No está validada:</u> Al ser servicios que pueden haber sido desarrollados por el propio equipo de instalación o por otros grupos, es probable que no hayan sido validados. Por lo tanto, no se puede garantizar su consistencia y precisión. Esto será un punto débil para este tipo de fuente de datos en la fase de evaluación de la calificación de fiabilidad.</p> <p><u>Dependencia a configuraciones externas al CI:</u> Al no ser controlados por el ciclo establecido en el CI (agente-árbol-indicador-configuración de métodos matemáticos), si los datos y configuraciones empleadas en este servicio no se facilitan o son defectuosos, los <i>cKPIs</i> generados a partir de ellos podrían estar corruptos.</p> <p><u>Se necesitan cálculos adicionales:</u> A pesar de considerarse automatizado, requerirá una configuración y unos acuerdos por parte del módulo de ejecución de servicios. Esto puede llevar un tiempo, esfuerzo e imprecisiones que podría afectar a la corrección, a la cercanía al tiempo real y a la visualización de los result.</p> |   |

|   |  |
|---|--|
| <b>Introducción manual</b>  | (a través de formularios) Formulario web UI para conocer el tiempo medio de permanencia de un tren en un puerto... |
| <b>Puntos fuertes:</b> Personalización, robusto ante fallos tecnológicos, replicable  |  |
| <p><u>Ad-hoc para el usuario:</u> Una de las ventajas de utilizar formularios para la inserción de datos es su capacidad para adaptarse al usuario final. A través de los formularios, pueden implementarse técnicas como límites y restricciones en los campos.</p> <p><u>Menos propenso a los fallos de disponibilidad:</u> El hecho de ser una utilidad de software desarrollada dentro del propio entorno de la solución (módulo de visualización) es menos propenso a problemas de disponibilidad.</p> <p><u>Replicable:</u> Una vez que un formulario ha sido desarrollado, puede ser replicado con ligeras modificaciones para rellenar otros datos necesarios para el CI. Además, los cambios se pueden realizar de forma muy sencilla, por lo que la extensión a otros escenarios no debería suponer un gran esfuerzo (escalabilidad).</p> |  |
| <b>Puntos débiles</b> Error humano (intencional o no), baja automatización.   |  |
| <p><u>Factor humano:</u> El mayor defecto de este tipo de imputación de datos es la intervención humana. Evidentemente, un indicador automatizado no debe emanar de datos tecleados por los humanos. Las personas cometen, la información no se coteja automáticamente y puede haber incorrecciones intencionales. Sin embargo, en algunos casos, para ciertos datos, podría ser necesario/útil para el usuario del CI la posibilidad de imputación manual. Además, esta flexibilidad permite el análisis de escenarios (<i>what-if</i>).</p> <p><u>Es necesario intervenir:</u> La automatización es baja, ya que se requiere la introducción de estos datos de forma manual. Esto podría ralentizar el proceso y crear diversos problemas (valores ausentes) que deriven en un indicador único CI no muy fiable.</p>                              |  |

Transparente para el gestor de contexto: La información procedente de los formularios se almacenará en el módulo de persistencia tras su inserción, sin su registro por la capa de adquisición de datos, por lo que en ningún caso podría existir una suscripción automatizada a dicha fuente.

Menor control de la actualización de datos: Los datos podrían estar totalmente desactualizados en los índices de persistencia. Además, puede ocurrir que el personal para cumplimentar los formularios no esté disponible en el momento de la solicitud (*cron*) por lo que aumentarían los valores ausentes y se obtendría un CI poco fiable.

Necesita una validación previa antes de aceptar los datos: Para evitar errores, falsos positivos, imputaciones erróneas y el incumplimiento de las restricciones, es necesario un preanálisis para establecer una estrategia de control de calidad de los datos en el momento de rellenar los formularios. Esto supone un esfuerzo adicional tanto para el usuario, los diseñadores de las políticas y los instaladores de la solución.

|  |  |
|--|--|
| <b>Valores medios de la literatura</b>   | Media del total de basura desembarcada de los barcos, tipo de combustible... |
| <b>Puntos fuertes:</b> Fácil despliegue de MVPs, independiente de la entidad   |  |
| <p><u>Bueno para un despliegue extremadamente rápido:</u> Podrían desarrollarse agentes que introdujeran datos estándar de una manera muy rápida. Esto permitiría crear demostradores eficaces de cara a una posible exposición tentativa de la herramienta.</p> <p><u>Fácil de depurar:</u> Una vez que se haya creado un producto inicial, será más fácil para el equipo técnico comprender los fallos en el enfoque tecnológico (agentes, métodos matemáticos) y mejorar iterativamente la implementación respectiva. Además, facilitará análisis de sensibilidad e incertidumbre sobre los resultados CI, al ser valores fijos.</p> <p><u>Disponibilidad nula - propensa a fallos:</u> La utilización de datos medios o de valores de referencia extraídos de las fuentes válidas (informes de la CE, informes de años anteriores de la entidad...) garantizará plena disponibilidad de los datos (sin valores ausentes).</p> <p><u>Replicable:</u> El mismo enfoque de selección de datos promedio (ver punto anterior) puede aplicarse a diferentes indicadores para mantener el mismo nivel de fiabilidad (RR). Se considera replicable ya que este método puede ser ejecutado por cualquier entidad independientemente de su tamaño, nivel de digitalización y disponibilidad de recursos.</p> |  |
| <b>Puntos débiles</b> Pobre precisión, no es IoT, nula automatización.   |  |
| <p><u>Falta de precisión:</u> Por muy fiable que sea la fuente de literatura seleccionada, no se basará en mediciones actuales ni en aportaciones reales, por lo que hay que poner en duda la fiabilidad del valor. Esta es una de las principales desventajas de utilizar datos medios o estimados procedentes de la experiencia/literatura.</p> <p><u>Digitalización cero:</u> Este enfoque no implica ningún proceso de digitalización, desarrollo o integración. El indicador compuesto generado a partir de los datos medios no dependería de ningún componente generador de valor.</p> <p><u>No es IoT:</u> Los datos se almacenarán a mano en la base de datos de persistencia sin ninguna validación. La información no cumplirá con ninguno de los conceptos de IoT.</p> <p><u>Requiere conocimientos:</u> Al igual que para la instalación de los sensores, el instalador o miembro de la entidad que inserte los valores de referencia manualmente debería tener amplios conocimientos del sector para garantizar la correcta selección de la fuente de información.</p>  |  |

# Apéndice C – Docker compose de instalación de la arquitectura

```

networks:
  cic_net:
    name: cic_net
    ipam:
      config:
        - subnet: 172.18.1.0/24

services:
  cic_mongodb_orion_svc:
    container_name:
cic_mongodb_orion_svc
    image: mongo:4.4.6
    restart: always
    networks:
      cic_net:
        ipv4_address: 172.18.1.2
    volumes:
      -
"./cic_mongodb_orion_vol/configdb:/data/con
figdb"
      -
"./cic_mongodb_orion_vol/db:/data/db"

  cic_orion_svc:
    container_name: cic_orion_svc
    image: fiware/orion:3.3.1
    hostname: orion
    restart: always
    depends_on:
      - cic_mongodb_orion_svc
    networks:
      cic_net:
        ipv4_address: 172.18.1.4
    command: -dbhost
cic_mongodb_orion_svc
    ports:
      - "1026:1026"

  cic_keyrock_svc:
    container_name: cic_keyrock_svc
    image: fiware/idm:8.1.0
    hostname: keyrock
    restart: always
    depends_on:
      - cic_mysql-server_svc
      - cic_orion_svc
      - cic_apirest_svc
    networks:
      cic_net:
        ipv4_address: 172.18.1.5

```

```

volumes:
  -
"./cic_keyrock_vol:/opt/fiware-
idm/certs"
  ports:
    - "8006:443"
  environment:
    - IDM_DB_HOST=cic_mysql-
server_svc
server_svc
  - IDM_DB_USER=root
  - IDM_DB_PASS=****
  - IDM_HOST=https://localhost
  - IDM_PORT=8006
  - IDM_CORS_ENABLED=true
  - IDM_PDP_LEVEL=basic
  - IDM_AUTHZFORCE_ENABLED=false
  - IDM_ADMIN_USER=admin
  - IDM_ADMIN_EMAIL=admin@cic.com
  - IDM_ADMIN_PASS=cic

  cic_mysql-server_svc:
    container_name: cic_mysql-
server_svc
server_svc
    image: mysql:5.7.37
    hostname: mysql-server
    restart: always
    networks:
      cic_net:
        ipv4_address: 172.18.1.6
    volumes:
      -
"./cic_mysql-
server_vol:/var/lib/mysql"
    environment:
      - MYSQL_ROOT_HOST=172.18.1.5
      - MYSQL_ROOT_PASSWORD=****

  cic_apirest_svc:
    container_name: cic_apirest_svc
    image: apirest:latest
    hostname: apirest
    restart: always
    depends_on:
      - cic_mongodb_historical_svc
      - cic_orion_svc
    networks:
      cic_net:
        ipv4_address: 172.18.1.8
    ports:
      - "8000:8000"

```

```

cic_historical-server_svc:
  container_name: cic_historical-
server_svc
  image: historical-server:latest
  restart: always
  depends_on:
    - cic_mongodb_historical_svc
    - cic_orion_svc
  networks:
    cic_net:
      ipv4_address: 172.18.1.9
  ports:
    - "8003:8003"
cic_wilma-for-orion_svc:
  container_name: cic_wilma-for-
orion_svc
  image: fiware/pep-proxy:8.1.0
  hostname: wilma-for-orion
  restart: always
  networks:
    cic_net:
      ipv4_address: 172.18.1.10
  depends_on:
    - cic_keyrock_svc
  ports:
    - "1027:1027"
  volumes:
    - "./cic_wilma_vol:/opt/fiware-
pep-proxy/cert"
  environment:
    - PEP_PROXY_PORT=1027
    - PEP_PROXY_HTTPS_ENABLED=true
    - PEP_PROXY_HTTPS_PORT=1027
    - PEP_PROXY_AUTH_ENABLED=true
    - PEP_PROXY_APP_ID=370092ea-7b94-
4d42-b7c3-60632ffb9cb9
    - PEP_PROXY_USERNAME=pep_proxy_34966ae8-d89a-
48fe-ab02-edde1d66c4c9
    - PEP_PASSWORD=*****
      - PEP_PROXY_PDP=idm
      - PEP_PROXY_MAGIC_KEY=*****
      - PEP_PROXY_APP_HOST=orion
      - PEP_PROXY_APP_PORT=1026
      - PEP_PROXY_APP_SSL_ENABLED=false
      - PEP_PROXY_IDM_HOST=keyrock
      - PEP_PROXY_IDM_PORT=443
      - PEP_PROXY_IDM_SSL_ENABLED=true

cic_wilma-for-apirest_svc:
  container_name: cic_wilma-for-
apirest_svc
  image: fiware/pep-proxy:8.1.0
  hostname: wilma-for-apirest
  restart: always

networks:
  cic_net:
    ipv4_address: 172.18.1.11
  depends_on:
    - cic_keyrock_svc
  ports:
    - "1028:1028"
  volumes:
    - "./cic_wilma_vol:/opt/fiware-
pep-proxy/cert"
  environment:
    - PEP_PROXY_PORT=1028
    - PEP_PROXY_HTTPS_ENABLED=true
    - PEP_PROXY_HTTPS_PORT=1028
    - PEP_PROXY_AUTH_ENABLED=true
    - PEP_PROXY_APP_ID=5a6f90bf-0559-
462e-81a1-a848a1ea159c
    - PEP_PROXY_USERNAME=pep_proxy_85ab7729-e665-
4954-a3b6-f92da1e00f1b
    - PEP_PASSWORD=*****
      - PEP_PROXY_PDP=idm
      - PEP_PROXY_MAGIC_KEY=*****
      - PEP_PROXY_APP_HOST=apirest
      - PEP_PROXY_APP_PORT=8000
      - PEP_PROXY_APP_SSL_ENABLED=false
      - PEP_PROXY_IDM_HOST=keyrock
      - PEP_PROXY_IDM_PORT=443
      - PEP_PROXY_IDM_SSL_ENABLED=true

cic_visualization_api_svc:
  container_name:
visualization_api_svc
  image: visualization_api:latest
  restart: always
  depends_on:
    - cic_orion_svc
  networks:
    cic_net:
      ipv4_address: 172.18.1.12
  ports:
    - "8001:8001"
    - "8002:8002"

cic_web-client_svc:
  container_name: cic_web-client_svc
  image: web-client:latest
  restart: always
  depends_on:
    - cic_visualization_api_svc
  networks:
    cic_net:
      ipv4_address: 172.18.1.13
  ports:
    - "8005:8005"

```

```

# Versión normal (recomendada)
cic_elasticsearch_svc:
    container_name:
cic_elasticsearch_svc
    image: elasticsearch:8.3.2
    restart: always
    networks:
        cic_net:
            ipv4_address: 172.18.1.17
    volumes:
-
"/cic_elasticsearch_vol:/usr/share/elasticsearch/data"
    environment:
        - discovery.type=single-node
        - http.host=0.0.0.0
-
ingest.geoip.downloader.enabled=false
-
xpack.security.transport.ssl.enabled=false
    - xpack.security.enabled=true
    - http.cors.enabled=true
    - http.cors.allow-origin="*"
    - ELASTIC_USERNAME=cic
    - ELASTIC_PASSWORD=cic
    ports: "9200:9200"

#versión alternativa - con MongoDB
# cic_mongodb_historical_svc:
#     container_name:
cic_mongodb_historical_svc
#     image: mongo:4.4.6
#     restart: always
#
#     networks:
cic_net: ipv4_address:172.18.1.3
#     volumes:
#         - ./cic_mongodb_historical_vol
#           /configdb:/data/configdb"
#
-
"/cic_mongodb_historical_vol/db:/data/db"

```

```

cic_kibana_svc:
    container_name: cic_kibana_svc
    image: kibana:8.3.2
    restart: always
    depends_on:
        - cic_elasticsearch_svc
    networks:
        cic_net:
            ipv4_address: 172.18.1.18
    environment:
-
server.publicBaseUrl=http://158.42.161.177:9140
-
ELASTICSEARCH_HOSTS=http://158.42.161.177:9155
    - ELASTICSEARCH_USERNAME=cic
    - ELASTICSEARCH_PASSWORD=cic
    ports:
        - "5601:5601"

# Versión con motor de reglas como
# componente externo en lugar de dentro del
# código del servicio.
cic_rules-engine_svc:
    container_name: cic_rules-
engine_svc
    image: rules-engine:latest
    restart: always
    depends_on:
        - cic_orion_svc
    networks:
        cic_net:
            ipv4_address: 172.18.1.19
    ports:
        - "8007:8007"

```