



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Arquitectura de IoT para la implementación de servicios cognitivos

Departamento de Comunicaciones
Universitat Politècnica de València

Tesis presentada para la obtención del grado de
Doctor por la Universitat Politècnica de València
Valencia, diciembre de 2023

Autor:
Clara Isabel Valero López

Director:
Dr. Carlos Enrique Palau Salvador

“Recuerda mirar arriba, a las estrellas, y no abajo, a tus pies”.
Stephen Hawking

Resumen

Internet of Things (IoT) se ha convertido en una realidad omnipresente que está transformando la forma en la que interactuamos con el entorno y cómo los dispositivos y objetos se conectan, comunican e intercambian información, abriendo un abanico de posibilidades y oportunidades sin precedentes. Sus aplicaciones son casi infinitas y gracias a su gran potencial, hoy en día está presente en todo tipo de ámbitos.

Artificial Internet of Things (AIoT) se considera la siguiente etapa de Internet. Se trata de un nuevo paradigma de red que combina IoT con la inteligencia artificial (IA) dando lugar a sistemas altamente inteligentes y autónomos, capaces de comprender, razonar y aprender de manera similar a los seres humanos. Sin embargo, a pesar de su potencial, uno de los desafíos clave que enfrenta AIoT desde sus inicios es la falta de una arquitectura de referencia estandarizada que proporcione un conjunto de funcionalidades básicas, estructuras de información y mecanismos que sirvan de modelo para desarrollar e implementar estos sistemas. Esta falta de estandarización está dando lugar a una serie de problemas y desafíos en su implementación y adopción generalizada. En primer lugar, la falta de estándares comunes dificulta la interoperabilidad entre diferentes dispositivos y sistemas AIoT. Además, la ausencia de una arquitectura de referencia complica el desarrollo coherente y eficiente. Por último, se observan dificultades para escalar los sistemas. Sin una estructura común y estándares interoperables, resulta más complicado integrar y administrar grandes cantidades de dispositivos y datos.

En esta tesis se presenta una arquitectura de referencia AIoT multidominio. El diseño de la arquitectura considera las recomendaciones de la ITU propuestas en UIT-T Y.2066. Además, la arquitectura propuesta se alinea a su vez con otras dos arquitecturas de referencia: IIRA y RAMI 4.0. Como consecuencia, la arquitectura presenta una naturaleza versátil y adaptable, lo que le permite ajustarse a las necesidades y requerimientos de diferentes contextos y dominios.

La arquitectura presentada ha sido implementada y validada en cuatro casos de uso desarrollados en el contexto de cuatro proyectos de investigación. En los

RESUMEN

proyectos ACTIVAGE - H2020 LSP y DIATOMIC Open Call n^o2 - H2020 la arquitectura facilita la implementación de servicios cognitivos de asistencia para personas mayores. En el proyecto COSIBAS - Programa Internacional ITEA3 incorpora a un sistema IoT heredado capacidades cognitivas para mejorar la gestión del tráfico marino. Por último, en OPTIMAI - H2020 se incorporan capacidades cognitivas a una línea de producción de antenas para detectar anomalías.

Resum

Internet of Things (IoT) s'ha convertit en una realitat omnipresent que està transformant la manera com interactuem amb l'entorn i com els dispositius i objectes es connecten, comuniquen i intercanvien informació, obrint un ventall de possibilitats i oportunitats sense precedents. Les seves aplicacions són quasi infinites i gràcies al seu gran potencial, avui dia és present en tot tipus d'àmbits.

Artificial Internet of Things (AIoT) es considera la següent etapa de la Internet. Es tracta d'un nou paradigma de xarxa que combina IoT amb la intel·ligència artificial (IA) donant lloc a sistemes altament intel·ligents i autònoms, capaços de comprendre, raonar i aprendre de manera similar a els éssers humans. Tot i això, malgrat el seu potencial, un dels desafiaments clau que enfronta AIoT des dels seus inicis és la manca d'una arquitectura de referència estandarditzada que proporcioni un conjunt de funcionalitats bàsiques, estructures d'informació i mecanismes que serveixin de model per desenvolupar implementar aquests sistemes. Aquesta manca d'estandardització dona lloc a una sèrie de problemes i desafiaments en la seva implementació i adopció generalitzada. En primer lloc, la manca d'estàndards comuns dificulta la interoperabilitat entre diferents dispositius i sistemes AIoT. A més a més, l'absència d'una arquitectura de referència complica el desenvolupament coherent i eficient. Per acabar, s'observen dificultats per escalar els sistemes. Sense una estructura comuna i estàndards interoperables, és més complicat integrar i administrar grans quantitats de dispositius i dades.

En aquesta tesi es presenta una arquitectura de referència AIoT multidomini. El disseny de l'arquitectura considera les recomanacions de la ITU proposades a UIT-T Y.2066. A més, l'arquitectura proposada s'alinea alhora amb dues arquitectures de referència més: IIRA i RAMI 4.0. Com a conseqüència, l'arquitectura presenta una naturalesa versàtil i adaptable, cosa que permet ajustar-se a les necessitats i requeriments de diferents contextos i dominis de la IoT.

L'arquitectura presentada ha estat implementada i validada en quatre casos d'ús desenvolupats en el context de quatre projectes de recerca. Als projectes

RESUM

ACTIVAGE - H2020 LSP i DIATOMIC Open Call n^o2 - H2020 l'arquitectura facilita la implementació de serveis cognitius d'assistència per a gent gran. Al projecte COSIBAS - Programa Internacional ITEA3 incorpora a un sistema IoT legacy capacitats cognitives per millorar la gestió del trànsit marí. Per acabar, a OPTIMAI - H2020 s'incorporen capacitats cognitives a una línia de producció d'antenes per detectar anomalies.

Abstract

Internet of Things (IoT) has become an omnipresent reality that is transforming the way we interact with the environment and how devices and objects connect, communicate, and exchange information, opening a range of unprecedented possibilities and opportunities. Its applications are almost infinite, and thanks to its tremendous potential, it is now present in all kinds of fields.

Artificial Internet of Things (AIoT) is considered the next stage of the Internet. It is a new network paradigm that combines IoT with Artificial Intelligence (AI), resulting in highly intelligent and autonomous systems capable of understanding, reasoning, and learning similar to humans. However, despite its potential, one of the key challenges that AIoT has faced since its inception is the lack of a standardized reference architecture that provides a set of basic functionalities, information structures, and mechanisms to serve as a model for developing and implementing these systems. This lack of standardization has led to a series of problems and challenges in its implementation and widespread adoption. Firstly, the lack of common standards hinders interoperability between different AIoT devices and systems. Additionally, the absence of a reference architecture complicates coherent and efficient development. Lastly, scaling the systems poses difficulties. Without a common structure and interoperable standards, integrating and managing large amounts of devices and data becomes more complex.

This thesis presents a multidomain AIoT reference architecture that incorporates recommendations from the ITU, as proposed in UIT-T Y.2066. Furthermore, the proposed architecture aligns with two other reference architectures: IIRA and RAMI 4.0. As a result, the architecture exhibits versatility and adaptability, allowing it to meet the needs and requirements of different IoT contexts and domains.

The presented architecture has been implemented and validated in four use cases developed in the context of four research projects. In the ACTIVAGE - H2020 LSP and DIATOMIC Open Call n^o2 - H2020 projects, the architecture facilitates the implementation of cognitive assistance services for the elderly.

ABSTRACT

In the COSIBAS - International ITEA3 Program, it incorporates cognitive capabilities into a legacy IoT system to improve maritime traffic management. Finally, in OPTIMAI - H2020, cognitive capabilities are integrated into an antenna production line to detect anomalies.

Agradecimientos

Primero quiero expresar mi más sincero agradecimiento a mi director de tesis, el Dr. Carlos E. Palau, por sus consejos y guía experta a través de todos estos años en los que he tenido el placer de trabajar con él.

A mi familia, por su respaldo y amor incondicional. Pero, sobre todo, por enseñarme que tu vida no queda restringida por tus limitaciones; el optimismo y la determinación marcan la diferencia.

Asimismo, deseo agradecer a mis amigos y seres queridos por su comprensión, apoyo y ánimo durante estos años. Quiero dar las gracias especialmente a mi amiga Carmen, por animarme a lanzarme a esta aventura y por sus generosos consejos. Sus palabras de aliento me han dado fuerza para seguir adelante. También quiero dedicarle unas palabras a mi amigo Diego, quien siempre encuentra la manera de sacar lo mejor de cualquier situación y su optimismo es contagioso.

Quiero extender mi gratitud a todos mis compañeros de laboratorio, incluidos aquellos que ya han completaron su ciclo aquí, con los que he compartido muchas horas de trabajo y también muchos buenos momentos. Su apoyo y amistad han hecho de esta una experiencia tanto enriquecedora como gratificante.

Y por último a mi compañero de aventuras, Álex. Su apoyo incondicional y su infinita paciencia han sido pilares fundamentales en cada etapa de este recorrido.

A todos, muchas gracias.

Clara Isabel Valero López
Valencia, diciembre de 2023

AGRADECIMIENTOS

Índice

Índice de figuras	XIII
Índice de tablas	XV
Acrónimos	XVII
1. Introducción	1
1.1. Introducción	1
1.2. Motivación	4
1.3. Metodología	6
1.4. Objetivos de la tesis	7
1.5. Principales aportaciones	8
1.5.1. Artículos	8
1.5.2. Congresos	9
1.5.3. Capítulos de libro	9
1.5.4. Proyectos de investigación	9
1.6. Estructura de la tesis	10
2. Estado del arte	13
2.1. Introducción	13
2.2. Internet of Things	14
2.2.1. Perspectiva general	14
2.2.2. Definición	17
2.2.3. Tecnologías de comunicación	20
2.2.4. Plataformas IoT	21
2.2.5. Arquitecturas de referencia	23
2.2.6. Interoperabilidad en IoT	28
2.2.7. Otros temas de investigación abiertos	32
2.3. Inteligencia Artificial	33
2.3.1. Perspectiva general	33

ÍNDICE

2.3.2. Definición	35
2.3.3. Tipos de aprendizaje	37
2.3.4. Algoritmos	39
2.3.5. Frameworks de IA	42
2.3.6. Artificial Intelligence as a Service	44
3. Definición de la arquitectura de referencia multidominio para AIIoT	49
3.1. Introducción	49
3.2. Análisis de requisitos	50
3.3. Arquitectura	53
3.3.1. Vista funcional	53
3.3.2. Vista de implementación	62
3.4. Conclusiones	64
4. Arquitectura cognitiva para la creación de servicios comunes en entornos de datos IIoT heterogéneos	67
4.1. Introducción	67
4.2. Arquitectura	68
4.3. Caso de uso: Creación de un ecosistema IIoT para envejecimiento activo	70
4.4. Implementación	73
4.4.1. Relación con la arquitectura de referencia	73
4.4.2. AIIoTES	75
4.5. Validación	80
4.5.1. Fase 1: Verificación de componentes	80
4.5.2. Fase 2: Validación en pilotos	80
4.5.3. Escenarios de validación	81
4.6. Resultados y discusión	82
4.6.1. Verificación de componentes	82
4.6.2. Validación en pilotos	83
4.7. EBASI	87
4.8. Conclusiones	104
5. Arquitectura AIIoT para la creación de servicios cognitivos en el ecosistema FIWARE	107
5.1. Introducción	107
5.2. Arquitectura	108
5.3. Componente cognitivo	114
5.3.1. Flujo de mensajes	116
5.3.2. Interfaces NGSI REST	119

5.4. Caso de uso: gestión del tráfico marítimo	121
5.5. Implementación	125
5.5.1. Relación con la arquitectura de referencia	125
5.5.2. Arquitectura COSIBAS	127
5.6. Validación	133
5.7. Resultados y discusión	134
5.7.1. Componente cognitivo	134
5.7.2. Servicios cognitivos	136
5.8. Conclusiones	142
6. Arquitectura AIoT para la creación de servicios cognitivos avanzados	145
6.1. Introducción	145
6.2. Arquitectura	146
6.2.1. Agente IoT	147
6.2.2. Context broker	148
6.2.3. Componente cognitivo	148
6.2.4. Base de datos de series temporales	150
6.2.5. Interfaz de usuario	151
6.3. Caso de uso: detección de anomalías en la tasa de recepción de datos OEE	151
6.4. Implementación	154
6.4.1. Relación con la arquitectura de referencia	154
6.4.2. Arquitectura OPTIMAI	156
6.5. Validación	160
6.6. Resultados y discusión	162
6.6.1. Agente IoT	162
6.6.2. Servicio cognitivo	163
6.7. Conclusiones	166
7. Conclusiones	169
7.1. Conclusiones finales	169
7.2. Líneas futuras de investigación	173
A. Entidades NGSI	177
Referencias	193

ÍNDICE

Índice de figuras

1.1. Aplicaciones AIoT.	3
1.2. Evolución cognitiva de las arquitecturas.	7
2.1. Número de dispositivos conectados a IoT en todo el mundo de 2019 a 2030.	15
2.2. Dominios de aplicación IoT.	16
2.3. La definición de IoT en su forma más simple.	19
2.4. Definición más completa de IoT.	19
2.5. Arquitecturas IoT - a) tres niveles, b) cinco niveles.	24
2.6. Las dimensiones de la interoperabilidad.	29
2.7. Visualización de la estructura del algoritmo <i>Decision Tree</i>	40
2.8. Visualización de la estructura del algoritmo <i>Random Forest</i>	40
2.9. Visualización del algoritmo <i>Support Vector Machines</i>	41
2.10. Visualización del algoritmo <i>k-Nearest Neighbor</i>	42
3.1. Vista funcional.	54
3.2. Vista funcional. Flujos de datos.	61
3.3. Vista de implementación.	63
4.1. Arquitectura para la creación de servicios comunes en entornos de datos heterogéneos.	69
4.2. AIoTES - Comparativa con arquitectura de referencia.	74
4.3. AIoTES.	76
4.4. Modelo de soledad.	89
4.5. Arquitectura EBASI.	91
4.6. Flujo de trabajo.	93
4.7. Arquitectura EBASI - Comparativa con arquitectura de referencia.	95
4.8. Implementación del sistema.	97
4.9. Asistente domótico.	98

ÍNDICE DE FIGURAS

4.10. Piloto.	100
5.1. Vista global de la arquitectura.	109
5.2. Arquitectura AIoT para la creación de servicios cognitivos en el ecosistema FIWARE.	110
5.3. Adaptador de contexto.	111
5.4. Adaptador NGSI.	112
5.5. Componente cognitivo.	115
5.6. Flujo del componente cognitivo.	118
5.7. Flujo de datos del demostrador.	124
5.8. Arquitectura COSIBAS - Relación con la arquitectura de referencia.	126
5.9. Arquitectura COSIBAS.	128
5.10. Flujo de mensajes.	129
5.11. Distribución de error en modelo ETA Valencia.	137
5.12. Interfaz de usuario - puerto de Valencia.	137
5.13. Predicción de ETA.	138
5.14. Distribución de error en modelo ETA Algeciras.	140
5.15. Interfaz de usuario - puerto de Algeciras.	141
5.16. Resultado de cálculo de ETA.	141
6.1. Arquitectura AIoT para la creación de servicios cognitivos. . .	147
6.2. Componente cognitivo.	149
6.3. Arquitectura OPTIMAI - Relación con la arquitectura de referencia.	155
6.4. Arquitectura OPTIMAI.	157
6.5. TVES OEE API swagger.	158
6.6. Diseño e implementación del agente IoT.	158

Índice de tablas

4.1. Resultados de la evaluación de los componentes.	82
4.2. Perfil de los participantes en la validación,	84
4.3. Complejidad subjetiva percibida (de 0 a 10) de los EV (columnas) en los pilotos (filas) según sus representantes.	85
4.4. Configuración por defecto del umbral de los parámetros ADL y SADL.	89
4.5. Configuración del umbral de los parámetros ADL y SADL. Usuario A.	100
4.6. Configuración del umbral de los parámetros ADL y SADL. Usuario B.	101
4.7. Datos reportados.	102
4.8. Interacciones del sistema.	103
5.1. Datos AIS.	131
5.2. Datos meteorológicos.	131
5.3. Importancia de los atributos del modelo ETA Valencia,	136
5.4. Resultados de las métricas de evaluación del modelo ETA Valencia,	137
5.5. Importancia de los atributos del modelo ETA Algeciras,	139
5.6. Resultados de las métricas de evaluación del modelo ETA Algeciras,	139
6.1. Datos de una sesión de fabricación.	153
6.2. Atributos seleccionados.	160
6.3. Matriz de confusión para un problema de clasificación de dos clases.	162
6.4. IF - Matriz de confusión (contamination parameter = 0,0054).	164
6.5. IF - Matriz de confusión (contamination parameter = 0,005).	164
6.6. IF - Matriz de confusión (contamination parameter = 0,0058).	164
6.7. Experimento 1 - Resultados de las métricas de evaluación,	164

ÍNDICE DE TABLAS

6.8. NB - Matriz de confusión.	165
6.9. DT - Matriz de confusión.	165
6.10. RF - Matriz de confusión.	165
6.11. KNN - Matriz de confusión.	166
6.12. Experimento 2 - Resultados de las métricas de evaluación, . . .	166

Acrónimos

ACTIVAGE	<i>ACTivating InnoVative IoT smart living environments for AGEing well</i>
ADL	<i>Activities of Daily Living</i>
AIOTI	<i>Alliance for the Internet of Things Innovation</i>
AIaaS	<i>AI as a Service</i>
AI PaaS	<i>AI Platform as a Service</i>
AIS	<i>Automatic Identification System</i>
AIoT	<i>Artificial Intelligence of Things</i>
AIoTES	<i>ACTIVAGE IoT Ecosystem Suite</i>
API	<i>Application Programming Interface</i>
AR	<i>Arquitectura de Referencia</i>
ATA	<i>Actual Time of Arrival</i>
CIoT	<i>Cognitive Internet of Things</i>
CEP	<i>Complex Event Processing</i>
COSIBAS	<i>Cognitive services for IoT-based scenarios</i>
COG	<i>Course Over Ground</i>
DL	<i>Deep Learning</i>
DT	<i>Decision Tree</i>
EBASI	<i>iot based Emotion and Behaviour recognition Against elderly people Social Isolation</i>
ETA	<i>Estimated Time of Arrival</i>
HDFS	<i>Hadoop Distributed File System</i>
HLA	<i>High Level Architecture</i>

ACRÓNIMOS

IDM	<i>Identity Management</i>
IETF	<i>Internet Engineering Task Force</i>
IF	<i>Isolation Forest</i>
IIC	<i>Industrial Internet Consortium</i>
IIoT	<i>Industrial Internet of Things</i>
IIRA	<i>Industrial Internet Reference Architecture</i>
IA	<i>Inteligencia Artificial</i>
IoT	<i>Internet of Things</i>
IPSM	<i>Inter Platform Semantic Mediator</i>
ITU	<i>International Telecommunication Union</i>
KNN	<i>K-Nearest Neighbor</i>
KPI	<i>Key Performance Indicator</i>
LoRAWAN	<i>Long Range Wide Area Network</i>
LR	<i>Linear Regression</i>
MAE	<i>Mean Absolute Error</i>
ML	<i>Machine Learning</i>
MMSI	<i>Maritime Mobile Service Identity</i>
NB	<i>Naïve Bayes</i>
NB-IoT	<i>Narrow band-Internet of Things</i>
NLP	<i>Natural Language Processing</i>
NFC	<i>Near Field Communication</i>
NGSI	<i>Next Generation Service Interface</i>
NMEA	<i>National Marine Electronics Association</i>
OCB	<i>Orion Context Broker</i>
OCF	<i>Open Connectivity Foundation</i>
OEE	<i>Overall equipment effectiveness</i>
OMI	<i>Organización Marítima Internacional</i>
OMS	<i>Organización Mundial de la Salud</i>
PaaS	<i>Platform as a Service</i>
PAP	<i>Policy Administration Point</i>
PDP	<i>Policy Decision Point</i>

PEP	<i>Policy Enforcement Point</i>
PIP	<i>Policy Information Point</i>
RAMI 4.0	<i>Reference Architectural Model Industrie 4.0</i>
RF	<i>Random Forest</i>
RGPD	Reglamento General de Protección de Datos
RMSE	<i>Root Mean Squared Error</i>
SADL	<i>Social Activities of Daily Living</i>
SATRD	Sistemas y Aplicaciones de Tiempo Real Distribuidos
STH	<i>Short Time Historic Framework</i>
SOLAS	<i>Safety Of Life At Sea</i>
SOG	<i>Speed Over Ground</i>
SVM	<i>Support Vector Machine</i>
TDD	<i>Test-Driven Development</i>
THD	Tecnologías Habilitadoras Digitales
TIC	Tecnologías de la Información y la Comunicación
UPV	Universidad Politécnica de Valencia

ACRÓNIMOS

Capítulo 1

Introducción

1.1. Introducción

Internet es una red informática mundial formada por un conjunto descentralizado de redes de comunicaciones interconectadas. Con la llegada de *Internet of Things* (IoT), su visión y alcance se amplió. Esta nueva tecnología permite recoger e intercambiar datos de forma automática y sistemática de cualquier dispositivo conectado a Internet. A menudo, IoT es descrito como una extensión de Internet enfocada hacia los objetos y lugares físicos del mundo. Sus aplicaciones son casi infinitas y gracias a su gran potencial, actualmente está presente en todo tipo de ámbitos como la logística, el transporte, la agricultura, la salud, el hogar inteligente, la industria o los vehículos. Esta rápida integración se debe no solo por sus múltiples beneficios, sino también a su bajo coste. Así, se ha reducido la distancia entre las estructuras físicas y el ciberespacio, permitiendo una monitorización más efectiva de los sistemas. Gracias a ello, nos encontramos en una nueva etapa donde las organizaciones pueden controlar y gestionar sus datos de manera inteligente.

IoT ha significado un gran avance en la adquisición de datos. Con el avance de la tecnología, se ha incrementado exponencialmente las cantidades de datos que deben ser almacenadas, analizadas y procesadas [1], [2]. Un análisis profundo de los datos da lugar a una interpretación sofisticada que permite generar valor en términos de una detección más rápida, mejores pronósticos y decisiones con una mayor flexibilidad general. Sin embargo, actualmente una gran cantidad de los datos obtenidos en redes IoT no se utilizan [1]. Los volúmenes son tan grandes y heterogéneos que su análisis resulta un desafío y generalmente solo se examina un pequeño porcentaje de los datos recogidos.

CAPÍTULO 1. INTRODUCCIÓN

En este punto, tecnologías como *big data* [3], analíticas predictivas [4], Inteligencia Artificial (IA) [5] y *Machine Learning* (ML) [6] facilitan la extracción del conocimiento. Permiten relevar tendencias, patrones no vistos, correlaciones ocultas y extraer nueva información [7]. Sin embargo, la convivencia de IoT con estas tecnologías es un reto aún presente. Por un lado, la unificación de los datos es un factor especialmente crítico y complejo [1], [8].

Durante las primeras etapas de IoT, *big data* ha sido una tecnología clave para el análisis de datos IoT. Sin embargo, a medida que IoT ha evolucionado, se ha evidenciado la necesidad de enfoques más avanzados y flexibles para el análisis de datos. La arquitectura dinámica, descentralizada y heterogénea de los sistemas IoT impulsa la demanda de soluciones que puedan adaptarse a entornos cambiantes y procesar datos de manera más ágil. En este contexto, la incorporación de IA emerge como una solución esencial para abordar los desafíos cada vez más complejos de los sistemas IoT. ML ofrece la capacidad de aprender y adaptarse a patrones de datos en tiempo real, permitiendo una toma de decisiones más rápida y precisa. Aplicar algoritmos de ML a los datos generados por los dispositivos IoT, permite identificar tendencias, anomalías y correlaciones que podrían pasar desapercibidas mediante enfoques tradicionales.

Además, otra cuestión importante es la automatización de la toma de decisiones para el posterior análisis de datos. Con volúmenes tan grandes resulta crítico reducir la intervención humana al mínimo posible. Para ello es conveniente emplear algoritmos genéticos [9], redes neuronales [10] y distintos métodos de aprendizaje [11] de ML y *Deep Learning* (DL). Las redes neuronales profundas destacan por su capacidad para analizar grandes volúmenes de información y tomar decisiones complejas [12].

Cognitive Internet of Things (CIoT) [13] es un nuevo paradigma de red, donde los objetos (físicas/virtuales) están interconectados y se comportan como agentes con una intervención humana mínima. Utiliza una metodología que permite aprender tanto del entorno físico como del entorno virtual. Almacena semánticamente los conocimientos aprendidos en bases de datos y se adapta a los cambios mediante mecanismos de toma de decisiones. En sus primeras etapas, los sistemas CIoT se integran con tecnologías como la computación en la nube (*cloud*) o *big data* para almacenar y analizar los datos. Estas tecnologías permiten a las redes CIoT observar el entorno y responder de acuerdo con reglas preestablecidas. No se trata de sistemas totalmente autónomos, ya que solo pueden tomar sus decisiones en función del contexto inmediato. Este se puede considerar el nivel de inteligencia más bajo de un sistema IoT. Alcanzar el máximo potencial implica que este sea capaz de percibir, interpretar y comprender, de forma muy similar a los seres humanos, así como de asimilar nuevos conocimientos construidos a partir de los datos de la red.

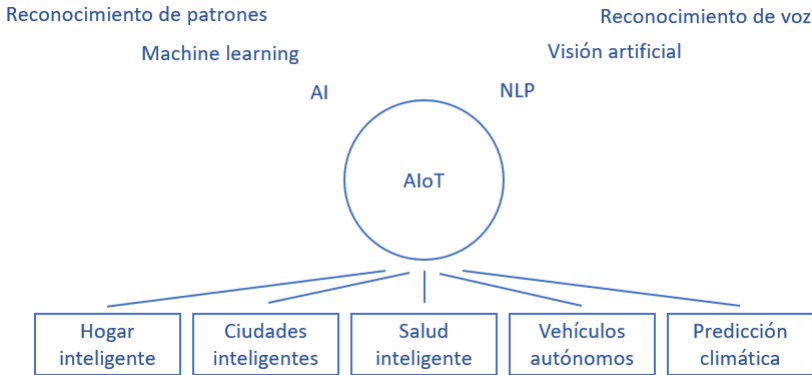


Figura 1.1: Aplicaciones AIoT.

Por otro lado, tras años de estancamiento y gracias a una combinación de factores clave, la IA ha experimentado un gran avance. Uno de los principales motivos es el aumento en la disponibilidad y accesibilidad de grandes volúmenes de datos, que permite entrenar modelos más precisos. Además, el desarrollo y la adopción generalizada de técnicas de aprendizaje profundo también ha sido clave para el avance en áreas como la visión por computadora, *Natural Language Processing* (NLP) o la robótica. Otro factor que ha contribuido significativamente ha sido el aumento en la capacidad de procesamiento de las computadoras. Los avances en el *hardware*, como las GPUs y las TPUs, permiten realizar cálculos de manera mucho más rápida y eficiente. La capacidad de recopilar y analizar grandes cantidades de datos ha sido fundamental para mejorar la precisión y la generalización de los sistemas de IA. Estos avances han dado lugar a la adopción generalizada de la IA en una amplia gama de industrias y sectores como la salud, la energía o la educación. Además, está presente en una gran cantidad de servicios que usamos en nuestro día a día. Por ejemplo, en forma de asistentes de voz en nuestros dispositivos móviles o también como *chats* inteligentes (ChatGPT [14], Phind [15] o Bard [16]). También la encontramos en sistemas de reconocimiento facial [17], sistemas de recomendaciones en plataformas de entretenimiento [18] o de conducción autónoma [19].

Impulsado por los rápidos avances de la IA, IoT ha empezado a integrarse con esta tecnología. Pese a que actualmente esta integración se encuentra en sus primeras etapas, ya se está presenciando la convergencia gradual de ambas, dando lugar a *Artificial Intelligence of Things* (AIoT). La infraestructura habilitada por AIoT ofrece una mejora notable en las capacidades predictivas de los sistemas. Ambas tecnologías se benefician mutuamente la una de la otra.

La IA añade flexibilidad e inteligencia a IoT a través de algoritmos avanzados, mientras que IoT proporciona datos a la IA. Se espera que la combinación de ambas tecnologías permita mejorar los procesos, optimizar la toma de decisiones y potenciar la eficiencia en diversos campos y aplicaciones. La Figura 1.1 muestra algunas aplicaciones basadas en AIoT.

1.2. Motivación

Los orígenes de la IA se remontan a los años 50. Sin embargo, este campo ha experimentado varios periodos de estancamiento [20] debido, entre otros problemas, a la limitación de datos disponibles para el entrenamiento de los algoritmos. Esta escasez de datos deriva en dos problemáticas. Por un lado, si la cantidad de datos es insuficiente, los modelos no logran capturar toda la complejidad y diversidad del problema que intentan resolver. El resultado es un modelado parcial del problema bajo estudio. Por otro lado, la limitación de los datos también puede provocar un sobreajuste en los modelos. El algoritmo memoriza los datos disponibles en lugar de aprender patrones generalizables. Esto lleva a que funcione bien con los datos de entrenamiento, pero no puede generalizar bien para nuevos datos, perdiendo su utilidad en situaciones prácticas.

Los datos son la base fundamental de IoT y la IA. IoT facilita la integración y almacenamiento de datos de diferentes fuentes. Esto permite un acceso más rápido y sencillo a la información para su posterior análisis. Por otro lado, la IA se basa en el aprendizaje a partir de información. Los modelos de IA aprenden de una amplia variedad de ejemplos para capturar las relaciones y patrones subyacentes en los datos.

Se trata de dos tecnologías complementarias que se benefician mutuamente. Los datos son el pilar fundamental de la IA, lo que implica la necesidad de mantener altos estándares de calidad en su recopilación. Un problema habitual en los *datasets* de entrenamiento de los algoritmos es su baja calidad. A menudo, estos *datasets* están incompletos o contienen ruido, lo cual afecta negativamente el rendimiento de los modelos. La capacidad de IoT para recoger y etiquetar datos de múltiples fuentes, como sensores, dispositivos y sistemas conectados, crea una fuente rica y diversa de datos de entrenamiento para la IA. Además, IoT contribuye a la recopilación continua de datos en tiempo real, lo cual permite a los modelos de IA actualizarse y adaptarse a medida que se generan nuevos datos. Esta separación de responsabilidades es clave para generar sistemas eficientes donde cada uno de sus elementos se especializa en un solo objetivo. Por ejemplo, etiquetar los datos correctamente y rápidamente es una tarea que consume mucho tiempo y energía. Es preferible que esta sea

abordada de forma específica en módulos IoT, dejando a los módulos de IA encargados del entrenamiento e inferencia de los algoritmos. También es importante señalar que la IA permite explotar los datos recopilados por IoT de una manera más exhaustiva. La IA detecta patrones, tendencias y anomalías en los datos, lo que permite identificar oportunidades de mejora, prever eventos futuros y tomar decisiones informadas de manera más efectiva que otras tecnologías.

Para aprovechar al máximo las sinergias presentes entre IoT e IA, es clave integrar ambas tecnologías a través de una visión unificada. Para ello es esencial proporcionar una base común que sirva como guía para el diseño y desarrollo de sistemas AIoT. Las Arquitectura de Referencia (AR) surgieron para tal fin. Proporcionan un conjunto de funcionalidades básicas, estructuras de información y mecanismos que sirven de modelo para desarrollar e implantar con éxito arquitecturas. Deben ser lo suficientemente flexibles como para poder aplicarse en diferentes dominios e implementarse teniendo en cuenta diferentes tecnologías, pero sin llegar a ser tan abstractas como para resultar poco prácticas.

Uno de los problemas a los que se enfrentan las arquitecturas IoT es su heterogeneidad y la falta de un estándar unificado. Simplemente hay demasiadas topologías, redes de acceso, protocolos, dispositivos, tipos de datos y tecnologías implicados. La naturaleza diversa y variada de los dispositivos, protocolos y plataformas utilizados en IoT hace que sea difícil lograr una interoperabilidad eficiente entre ellos. Para asegurar la calidad de los datos, una arquitectura AIoT debe tener en cuenta este hecho e incorporar mecanismos de interoperabilidad que permitan extraer el máximo conocimiento de estos sistemas.

Por otro lado, la mayoría de las arquitecturas propuestas IoT se definen de forma específica para cumplir los requerimientos particulares de cada dominio de aplicación y caso de uso [21], [22], [23]. La mayoría no consideran un mecanismo para describir la arquitectura de destino de manera separada de las transformaciones, sino que los autores definen la arquitectura y las tecnologías a emplear de tal manera que la lógica de modelado de la arquitectura queda combinada con las transformaciones. Esto lleva a que la arquitectura diseñada sea totalmente dependiente del dominio de aplicación y de sus requerimientos específicos y no se pueda reusar en otros casos.

En la actualidad, según el conocimiento del autor, no existe ningún trabajo publicado que presente una arquitectura de referencia AIoT que se esté imponiendo como estándar o que sea ampliamente aceptada en el ámbito académico o industrial. La mayoría de las arquitecturas AIoT presentadas están fuertemente acopladas a los casos de uso a los que dan solución [24]. Y, pese a que algunas de las arquitecturas AIoT presentadas son genéricas [25], no están ali-

CAPÍTULO 1. INTRODUCCIÓN

neadas con organismos de estandarización [26]. La falta de adhesión a estándares reconocidos puede limitar la aceptación y la confianza en las arquitecturas.

Así, la presente tesis está motivada por la necesidad de proporcionar una arquitectura de referencia AIoT que no solo unifique ambas tecnologías de forma eficiente, sino que también esté alineada con organismos de estandarización de prestigio y sirva como base para la creación de futuras arquitecturas AIoT. La arquitectura busca proporcionar un conjunto de requisitos, directrices y recomendaciones que guíen el diseño, el desarrollo y la implantación para facilitar la realización de estos sistemas. En resumen, la motivación principal de esta tesis es proponer una arquitectura de referencia AIoT integral, flexible y adaptable que permita eliminar la desconexión que existe entre estas dos tecnologías.

1.3. Metodología

En esta Sección se presenta la metodología seguida durante el presente trabajo. En primer lugar, se lleva a cabo una revisión del estado del arte. Posteriormente, se define el objetivo general junto con los objetivos específicos. Estos objetivos tienen como propósito contribuir al avance del campo de estudio a partir de las brechas identificadas en la revisión del estado del arte.

A continuación, se define una primera versión de la arquitectura de referencia. Esta es construida basándose en los fundamentos teóricos y conceptuales recopilados durante la revisión del estado del arte, así como en las necesidades específicas de la investigación.

La validación de una arquitectura es un paso crucial, ya que garantiza que la solución propuesta cumple con los objetivos establecidos y se ajusta adecuadamente a los requisitos propuestos. Sin embargo, la validación directa de una arquitectura de referencia no es posible. Debido a su naturaleza teórica, no puede ser validada en el sentido tradicional de realizar pruebas empíricas o experimentales para confirmar su funcionalidad o rendimiento. En su lugar, esta tesis presenta una validación *use case driven*. En lugar de validar la arquitectura de referencia, se validan sus distintas instanciaciones. Estas representan implementaciones concretas de la arquitectura propuesta, adaptadas a los requerimientos específicos de cada caso de uso. Esta metodología permite evaluar la efectividad de la arquitectura en diversos escenarios y asegurar que cada instanciación satisface sus requisitos particulares (cualitativos o cuantitativos). Al validar cada instancia individualmente, se obtiene una visión completa y detallada de la adaptabilidad de la arquitectura en situaciones prácticas. Este proceso de validación proporciona información para retroalimentar el diseño de la arquitectura, permitiendo ajustes y mejoras en su formulación para futuras implementaciones.

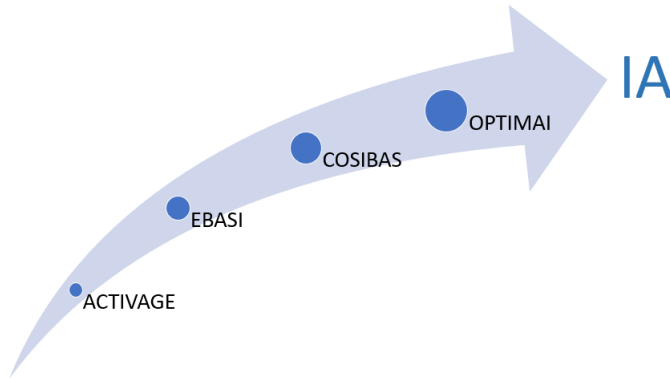


Figura 1.2: Evolución cognitiva de las arquitecturas.

Por lo tanto, a partir de esta primera versión de la arquitectura se inicia un proceso iterativo formado por los siguientes pasos: instanciación, validación de la arquitectura y retroalimentación.

La arquitectura de referencia evoluciona según lo hacen los avances tecnológicos y también en función de las necesidades de las arquitecturas específicas generadas. En total se instancian cuatro arquitecturas: ACTIVAGE, EBASI, COSIBAS y OPTIMAI. La Figura 1.2 muestra la evolución cognitiva de las arquitecturas, reflejo del progreso de la arquitectura de referencia. En las primeras instanciaciones, la arquitectura presenta un estado embrionario y sus capacidades cognitivas son débiles (arquitectura ACTIVAGE). Esta pone el foco en definir adecuadamente los niveles inferiores, responsables de permitir una extracción de datos de calidad. A partir de ese momento las capacidades cognitivas de las nuevas instancias crecen. EBASI presenta una inteligencia débil, basada en *Key Performance Indicator* (KPI) y reglas. Por último, se definen COSIBAS y OPTIMAI. Ambas incorporan una inteligencia más sólida, basada en ML. Es en este punto donde se completa la definición de la arquitectura de referencia. Estas últimas variantes representan un importante avance en la definición de la arquitectura, y permiten evolucionar de una arquitectura CIoT a una arquitectura AIoT.

1.4. Objetivos de la tesis

Teniendo en cuenta las motivaciones anteriores, el objetivo principal de la presente tesis es:

Definir una arquitectura de referencia multidominio AIoT. La arquitectura debe integrar ambas tecnologías de manera unificada, apoyándose en estándares ampliamente aceptados. Además, debe ofrecer una estructura clara y bien definida que guíe el diseño y desarrollo de este tipo de sistemas. Igualmente, debe promover la coherencia, reutilización y eficiencia de sus componentes en el proceso de creación de soluciones específicas.

Sobre este objetivo principal se derivan los siguientes objetivos específicos:

- **O1.** Identificar y enmarcar la problemática de la extracción de conocimiento en redes IoT mediante el razonamiento y el aprendizaje sobre los datos.
- **O2.** Analizar las metodologías, plataformas y tecnologías habilitadoras de IoT e IA para asentar las bases en el desarrollo de la solución al problema planteado.
- **O3.** Definir una arquitectura de referencia multidominio que habilite el uso de servicios cognitivos sobre IoT.
- **O4.** Instanciar y validar la arquitectura de referencia en al menos un entorno real perteneciente al dominio de la salud.
- **O5.** Instanciar y validar la arquitectura de referencia en al menos un entorno real perteneciente al dominio del transporte marítimo.
- **O6.** Instanciar y validar la arquitectura de referencia en al menos un entorno real perteneciente al dominio de la industria 4.0.

1.5. Principales aportaciones

1.5.1. Artículos

- **Valero, C. I.,** Gil, A. M. M., Gonzalez-Usach, R., Julian, M., Fico, G., Arredondo, M. T., ... & Palau, C. E. (2021). *AIoTES: Setting the principles for semantic interoperable and modern IoT-enabled reference architecture for Active and Healthy Ageing ecosystems*. Computer Communications, 177, 96-111.

- **Valero, C. I.**, Ivancos Pla, E., Vaño, R., Garro, E., Boronat, F., & Palau, C. E. (2021). *Design and Development of an AIoT Architecture for Introducing a Vessel ETA Cognitive Service in a Legacy Port Management Solution*. *Sensors*, 21(23), 8133.
- **Valero, C. I.**, Martínez, Á., Oltra-Badenes, R., Gil, H., Boronat, F., & Palau, C. E. (2022). *Prediction of the Estimated Time of Arrival of container ships on short-sea shipping: A pragmatistical analysis*. *IEEE Latin America Transactions*, 20(11), 2354-2362.

1.5.2. Congresos

- **Valero, C. I.**, Gravina, S., Ciuffetti, A., Di Gioia, M., Gravina, R., Boronat, F., & Palau, C. E. (2022). *EBASI: IoT-Based Emotion and Behaviour Recognition System Against Elderly People Social Isolation*. In *International Symposium on Intelligent and Distributed Computing* (pp. 3-13). Springer, Cham.
- **Valero, C. I.**, Boronat, F., Esteve, M., & Palau, C. E. (2023). *AI for detecting variations in the OEE data reception rate in the manufacturing industry*. In 10th ECCOMAS thematic conference on smart structures and materials. Pre-print.

1.5.3. Capítulos de libro

- **Valero, C. I.**, Belsa, A., Fornes-Leal, A., Boronat, F., Llorente, M. A., & Montesinos, M. (2021). *INTER-Framework: An Interoperability Framework to Support IoT Platform Interoperability*. In *Interoperability of Heterogeneous IoT Platforms* (pp. 167-193). Springer, Cham.
- Belsa, A., Fornes-Leal, A., **Valero, C. I.**, Olivares, E., Suárez de Puga, J., Boronat, F., & Fuart, F. (2021). *INTER-Layer: A Layered Approach for IoT Platform Interoperability*. In *Interoperability of Heterogeneous IoT Platforms* (pp. 95-132). Springer, Cham.

1.5.4. Proyectos de investigación

La presente tesis doctoral se ha desarrollado en el contexto del grupo de investigación Sistemas y Aplicaciones de Tiempo Real Distribuidos (SATRD) del Departamento de Comunicaciones de la Universidad Politécnica de Valencia (UPV).

Durante el desarrollo de la tesis doctoral, se han establecido colaboraciones en diversos proyectos de investigación y desarrollo. Estas colaboraciones han

CAPÍTULO 1. INTRODUCCIÓN

permitido validar la arquitectura propuesta en distintos dominios de aplicación. Algunas de las publicaciones asociadas a la tesis han recibido financiamiento de estos proyectos:

- Proyecto ACTIVAGE - H2020 LSP. Acuerdo de subvención nº 732679.
- Proyecto DIATOMIC, *Open Call* nº2 - H2020. Acuerdo de subvención nº 761809.
- Proyecto COSIBAS - Programa Internacional ITEA3. Acuerdo de subvención nº .EXP 00110912/INNO-20181033.
- Proyecto OPTIMAI - H2020. Acuerdo de subvención nº 958264.

1.6. Estructura de la tesis

La memoria de la presente tesis doctoral está compuesta de siete Capítulos estructurados de la siguiente manera:

- En el Capítulo 1 se realiza una breve introducción de los términos IoT, CIoT e IA. Tras esto, se describen las motivaciones y los objetivos de la presente tesis. Por último, se detallan las contribuciones derivadas de este trabajo, así como su estructura. El Capítulo está relacionado con el objetivo **O1**.
- En el Capítulo 2 se revisa la tecnología sobre la que se basa la presente tesis, IoT e IA. En primer lugar, se da una visión general de ambas tecnologías. Seguidamente se definen y por último, se presentan sus características, arquitecturas y plataformas. El Capítulo está relacionado con el objetivo **O2**.
- En el Capítulo 3 se define una arquitectura de referencia de AIoT para la implementación de servicios cognitivos. Primero se detallan sus requisitos y a partir de estos se definen los niveles que forman la arquitectura de alto nivel. El Capítulo está relacionado con el objetivo **O3**.
- En el Capítulo 4 se presenta una primera transformación de la arquitectura de referencia para la creación de servicios comunes en entornos de datos IoT heterogéneos. Esta arquitectura es instanciada en el dominio de la salud, en concreto dentro del paradigma del envejecimiento activo. Se presenta una arquitectura embrionaria, AIoTES, que permite la creación de un ecosistema de soluciones y servicios para envejecimiento

activo. En una de las Secciones del Capítulo se detalla una nueva arquitectura, EBASI, que permite reconocer el sentimiento de soledad en adultos mayores. Esta arquitectura se diseña para ser integrada dentro del ecosistema anterior, pero también para ser independiente a él. Ambas arquitecturas introducen una inteligencia débil. ACTIVAGE se basa en *big data* mientras que EBASI implementa un sistema de IA basado en reglas. El Capítulo está relacionado con el objetivo **O4**.

- En el Capítulo 5 se introduce una segunda transformación de la arquitectura de referencia para la creación de servicios cognitivos en el ecosistema FIWARE. Esta arquitectura introduce un componente cognitivo que permite la creación de servicios cognitivos basados en IA en entornos IoT. La arquitectura se implementa y valida en el dominio del transporte marítimo. La validación se realiza a través de la extensión de un sistema heredado para la gestión de tráfico marino. Este caso de uso busca añadir nuevos servicios cognitivos que permitan predecir con exactitud la hora de llegada de los buques a puerto a partir de distintas fuentes de datos IoT. El Capítulo está relacionado con el objetivo **O5**.
- En el Capítulo 6 se presenta una última transformación de la arquitectura de referencia. En este caso, la arquitectura implementa un componente cognitivo genérico y no acoplado a ningún ecosistema IoT. La validación se realiza en el dominio de la industria 4.0. Se busca crear e integrar un servicio cognitivo en un entorno industrial real para la detección de anomalías. La arquitectura OPTIMAI ofrece un servicio cognitivo para la detección de anomalías en la tasa de recepción de datos en una línea de producción robotizada de antenas. El Capítulo está relacionado con el objetivo **O6**.
- En el Capítulo 7 se exponen las principales conclusiones, así como las líneas futuras de investigación más relevantes.

CAPÍTULO 1. INTRODUCCIÓN

Capítulo 2

Estado del arte

2.1. Introducción

La computación cognitiva recibe una atención considerable tanto en el ámbito académico como en el industrial. En el ámbito académico, IEEE la define como “un campo interdisciplinario de investigación y aplicación”, que “utiliza métodos de la psicología, la biología, el procesamiento de señales, la física, la teoría de la información, las matemáticas y la estadística” en un intento de construir “máquinas que tengan capacidades de razonamiento análogas a las de un cerebro humano” [27].

IoT se centra en la adquisición de datos, pero a menudo el análisis de esos datos queda en un segundo plano. Aplicar computación cognitiva a sistemas IoT recibe el nombre de CIIoT o AIIoT, dependiendo del grado de inteligencia del sistema. Estos sistemas aprovechan la ubicuidad de los dispositivos IoT para recopilar información a bajo coste. La IA permite incorporar nuevas capacidades avanzadas de análisis de datos en los ecosistemas.

Este Capítulo proporciona una breve introducción a conceptos clave para la definición de una arquitectura de IoT que permita la implementación de servicios cognitivos. La primera parte del capítulo presenta IoT, mientras que la segunda parte se centra en la IA.

2.2. Internet of Things

2.2.1. Perspectiva general

IoT es una red heterogénea formada por dispositivos inteligentes conectados a Internet que permite la comunicación entre los dispositivos que la forman proporcionando una identidad única a cada uno de ellos. Las palabras Internet y *Things* significan una red mundial interconectada basada en sensores, comunicación, redes y tecnologías de procesamiento de la información. Tal y como se señala en [28], IoT se considera clave en las Tecnologías de la Información y la Comunicación (TIC). Pese a que el término IoT fue acuñado principios de este siglo, esta tecnología y sus dispositivos están presentes en todo tipo de ámbitos gracias a los continuos avances de la tecnología entre los cuales destacan: i) el acceso a la tecnología de sensores de bajo coste y baja potencia, ii) mejora de la conectividad gracias a la creación de gran cantidad de protocolos de red para Internet y iii) el aumento en la disponibilidad de plataformas *cloud*.

El informe *Hype Cycle* del año 2012 identificó al IoT como una de las tecnologías emergentes del momento. Como constata el informe publicado en 2015, en ese año se consideraba que las expectativas de IoT estaban sobredimensionadas y se estimaba que la tecnología tardaría entre 5 y 10 años en madurar y ser adoptada de forma mayoritaria por el mercado. Sin embargo, desde su aparición IoT ha evolucionado con rapidez y vivimos rodeados de sensores y dispositivos que de forma autónoma recopilan y procesan datos de nuestro entorno, los transmiten a la nube donde son procesados para la toma de decisiones y reciben respuesta través de Internet.

IoT está presente en múltiples dominios de aplicación [29]. Hay gran cantidad de aplicaciones a nuestro alcance donde está presente; transporte, *smart home*, asistencia sanitaria, industria, medio ambiente, *smart city*, así como los robots de juegos sociales y personales. Un ejemplo son los hogares inteligentes, que se han convertido en una realidad gracias a la incorporación dispositivos inteligentes en las viviendas. Aparatos como Google Home Hub, Google Home Mini, las bombillas inteligentes GE C-Life, Amazon Echo Plus y Echo Dot son algunos ejemplos. La revolución de Internet ha transformado la forma en que los seres humanos consumen, buscan y comparten información.

IoT se considera la siguiente etapa de la evolución de Internet y se ha convertido en una de las tecnologías más importantes del siglo XXI. De acuerdo con el informe de Statistica publicado en julio de 2023 [30], en 2030 el número total de dispositivos conectados en el mundo será de aproximadamente 29.400 millones (Figura 2.1). Además, IoT ha sido identificado como una de las Tecnologías Habilitadoras Digitales (THD) [31]. Por consiguiente, IoT es considerado una

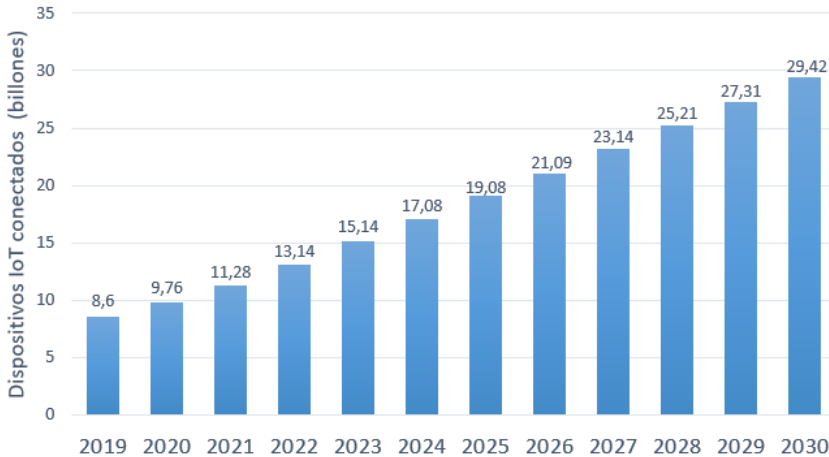


Figura 2.1: Número de dispositivos conectados a IoT en todo el mundo de 2019 a 2030.

tecnología estratégica para el desarrollo y la transformación digital de la economía y la sociedad.

Debido a su carácter transversal, están surgiendo nuevas aplicaciones de IoT para prácticamente todo tipo de industria y dominio de aplicación. Como se muestra en la Figura 2.2, IoT desempeña un papel importante en dominios tales como la sanidad [32], la logística [33], *smart cities* [34] o *smart home* [35], [36], [37]. De manera general, la integración de IoT en los diferentes dominios ha permitido automatizar la comunicación de los dispositivos con las aplicaciones, mejorando el rendimiento y el procesamiento de los datos. El objetivo de incorporar IoT en la sanidad es recoger y examinar los datos médicos en tiempo real para hacer frente a las limitaciones del diagnóstico médico tradicional, como los errores en el análisis médico o el retraso en los diagnósticos debido a que es un proceso que hasta el momento este ha estado limitado a la capacidad de procesamiento humana. La incorporación de IoT en el dominio de la logística permite rastrear rutas, sugerir rutas más eficientes además de ayudar a desarrollar aplicaciones de vigilancia, control de tráfico, detección de anomalías, predicción de tráfico y gestión de situaciones de emergencia. Las ciudades inteligentes utilizan la tecnología para mejorar la vida de las personas mediante aplicaciones inteligentes en diversos campos. Para implementarlas, se despliegan dispositivos IoT para optimizar diversos sistemas y servicios públicos, como la gestión del aparcamiento, la gestión de residuos, la gestión de la

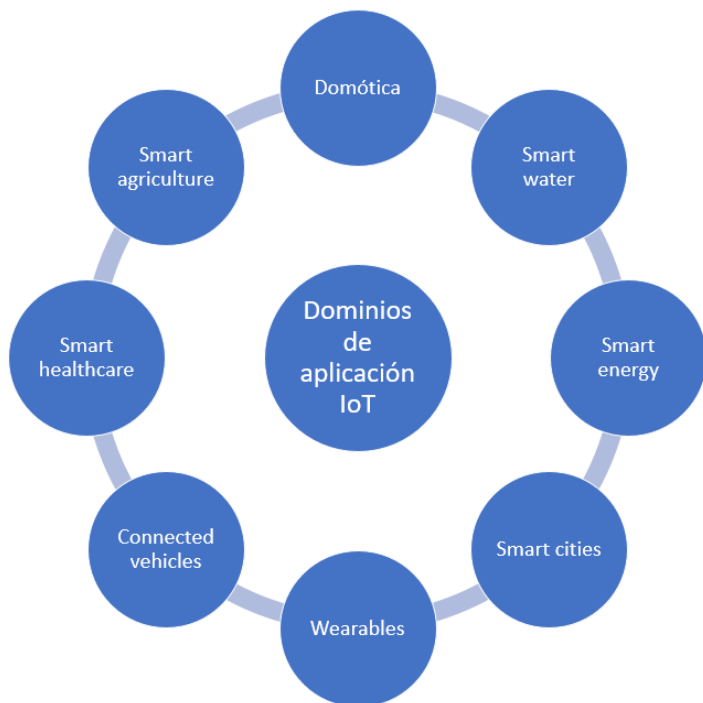


Figura 2.2: Dominios de aplicación IoT.

limpieza, la iluminación de las calles y el control de emergencias en la ciudad. El uso de dispositivos IoT los hogares permite proporcionar servicios como control de la privacidad, control de acceso, digitalización del vídeo, detección de intrusos, alarmas contra incendios o el control de la calidad del aire y de la iluminación.

IoT permite comunicar, procesar y analizar gran cantidad de datos dotando así a los sistemas de inteligencia y permitiendo que los dispositivos respondan automáticamente actuando sobre el medio siguiendo reglas predefinidas. Así, esta tecnología permite capturar y almacenar una enorme cantidad de datos procedentes de múltiples fuentes. Sin embargo, la recopilación, el procesamiento y el análisis de los datos no es trivial. Es difícil transmitir cantidades masivas de datos heterogéneos, analizar entornos complejos a partir de estos datos y luego tomar decisiones automáticas en poco tiempo. Para alcanzar el pleno potencial de IoT su integración con otras tecnologías como el *big data* o la IA está siendo decisiva. La convergencia de estas tecnologías ha originado una

tormenta perfecta que dota a los sistemas de la capacidad de dar sentido y extraer el significado de grandes cantidades de fuentes de datos. Gracias a la gran cantidad de datos es posible entrenar algoritmos o redes neuronales con gran precisión, lo cual permite analizar y predecir tendencias con gran exactitud [38].

2.2.2. Definición

De forma general, IoT se refiere a una red inteligente global que permite las interacciones ciberfísicas mediante la conexión de numerosas cosas con capacidad para percibir, computar, ejecutar y comunicarse con Internet; procesar e intercambiar información entre los objetos, los centros de datos y los usuarios; y ofrecer diversos servicios inteligentes.

Es importante señalar que no existe una definición única para IoT que esté aceptada de forma consensuada por la comunidad científica. De hecho, hay muchos grupos diferentes, entre ellos académicos, investigadores, profesionales, innovadores, desarrolladores y empresarios, que han definido el término. La definición exacta de IoT está todavía en proceso de formación y está sujeta a las perspectivas que se adopten. La presente Sección recopila las definiciones más relevantes, correspondientes a diversos organismos de estandarización o publicaciones de referencia.

A continuación, se listan algunas de las definiciones. Las más aceptadas son aquellas realizadas por organismos de estandarización:

- El IEEE, en su informe especial sobre IoT publicado en marzo de 2014 [39], describió el término como: “una red de elementos, cada uno integrado con sensores, que están conectados a Internet”.
- La *International Telecommunication Union* (ITU), en su recomendación ITU-T Y.2060 [40] lo define como: “Infraestructura mundial para la sociedad de la información que propicia la prestación de servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperabilidad de tecnologías de la información y la comunicación presentes y futuras”.
- La *Internet Engineering Task Force* (IETF) da su propia definición de IoT [41]: “IoT es la red de objetos físicos o “cosas” que llevan incorporada electrónica, *software*, sensores, actuadores y conectividad para permitir que los objetos intercambien datos con el fabricante, el operador u otros dispositivos conectados”.
- El Parlamento Europeo define IoT [42] como “IoT es una red distribuida que conecta objetos físicos capaces de detectar o actuar sobre su entorno

CAPÍTULO 2. ESTADO DEL ARTE

y de comunicarse entre sí, con otras máquinas o con ordenadores. Los datos que comunican estos dispositivos pueden recogerse y analizarse para revelar conocimientos y sugerir acciones que produzcan ahorros de costes, aumenten la eficiencia o mejoren los productos y servicios”.

No obstante, es importante resaltar el papel de la comunidad científica en el proceso de definición del término. Algunas de las definiciones propuestas son las siguientes:

- La definición propuesta por Xia *et al.* [43] es la siguiente: “IoT se refiere a la interconexión en red de los objetos cotidianos, que a menudo están equipados con inteligencia ubicua. IoT aumentará la ubicuidad de Internet al integrar todos los objetos para su interacción a través de sistemas integrados, lo que dará lugar a una red altamente distribuida de dispositivos que se comunicarán con los seres humanos y con otros dispositivos”.
- Dorsemayne *et al.* [44] lo definen como: “Conjunto de infraestructuras que interconectan objetos conectados y permiten su gestión, la extracción de datos y el acceso a los datos que generan”.
- Haller *et al.* [45] propusieron la siguiente definición: “Un mundo en el que los objetos físicos se integran perfectamente en la red de información, y en el que los objetos físicos pueden convertirse en participantes activos en el proceso empresarial”.
- Sarma *et al.* [46] definen las “cosas” de objetos físicos a objetos virtuales que representa como las identidades con conectividad a Internet.
- Rayes *et al.* [47] proponen la siguiente definición: “IoT es la red de las cosas, con una clara identificación de los elementos, dotada de inteligencia *software*, sensores y conectividad ubicua a Internet”.

Rayes *et al.*, tras aportar una definición inicial del término IoT, dan un paso más y aportan la definición minimalista de IoT. Para ello, para ellos resulta esencial definir previamente el término “objeto” con exactitud, pues es la esencia de la tecnología. Para ellos, “objetos” son en realidad “cualquier cosa” y “todo”, desde los electrodomésticos hasta los edificios, pasando por los coches, las personas, los animales, los árboles y las plantas. Por lo tanto, como se muestra en la Figura 2.3, en su forma más simple IoT, puede considerarse como la intersección de Internet, las cosas y los datos.

Tras esto, presentan una definición más compleja (Figura 2.4) que incluye los términos “estándares” y “procesos”. Según ella, los “objetos” se conectan a través de “Internet” para intercambiar “datos” utilizando “estándares” de la industria que garanticen la interoperabilidad y permitiendo “procesos” útiles y en su mayoría automatizados.

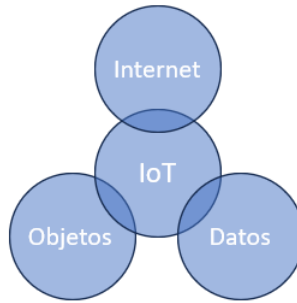


Figura 2.3: La definición de IoT en su forma más simple.

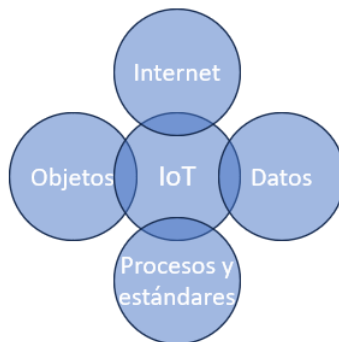


Figura 2.4: Definición más completa de IoT.

2.2.3. Tecnologías de comunicación

IoT incorpora multitud de tecnologías de comunicación. Los protocolos de comunicación son un aspecto clave de los sistemas IoT pues son los elementos encargados de capturar e intercambiar los datos del sistema. Ellos especifican el formato de la codificación e intercambio de datos, además de realizar el enrutamiento de los datos desde el origen hasta el destino. Además, proporcionan control de flujo y retransmisión de paquetes perdidos [48].

A continuación, se presentan brevemente los protocolos de comunicación y estándares más habituales en comunicaciones IoT.

- **ZigBee** [49]

Protocolo propuesto por la *ZigBee Alliance* [50] basado en el estándar de redes inalámbricas de bajo consumo IEEE802.15.4. Este proporciona un protocolo de comunicación de bajo coste para crear redes de área personal. *ZigBee* es el protocolo de comunicación adecuado para aplicaciones que necesitan una baja velocidad de datos, una mayor duración de la batería y dispositivos de red seguros. Además, *ZigBee* puede funcionar con diferentes topologías de red, como malla, estrella y árbol.

- **Bluetooth** [51]

Se trata de un protocolo de comunicación de corto alcance clave en el ámbito de la ropa inteligente. Existe una versión más reciente de *Bluetooth* llamada *Bluetooth Smart* o *Bluetooth Low-Energy*, cuya principal característica es su bajo consumo de energía y su mejorado radio de transferencia respecto al protocolo original.

- **Z-Wave** [52]

Es un protocolo comunicación por radiofrecuencia de baja potencia habitualmente empleado en domótica. Se trata de un protocolo sencillo, fiable y escalable que permite un desarrollo sencillo y rápido. Proporciona una comunicación de baja latencia y soporta velocidades de hasta 100kbit/s.

- **Near Field Communication** [53]

Near Field Communication (NFC) es un protocolo de comunicación inalámbrica de muy corto alcance (menos de 10cm) que proporciona comunicación bidireccional sencilla y segura entre dispositivos electrónicos. Dentro del área de IoT Se usa para conectar dispositivos IoT de forma fácil e intuitiva.

- **6LoWPAN** [54]

Fue el primer protocolo de comunicación de IoT y es el más empleado. Es un protocolo de red que envía paquetes IPv6 sobre redes IEEE 802.15.4 y que permite mecanismos de encapsulado y compresión de cabeceras. También admite direcciones de diferente longitud, bajo ancho de banda, diferentes topologías, incluyendo estrella o malla, consumo de energía, bajo coste, redes escalables, movilidad, falta de fiabilidad y largo tiempo de espera.

- ***Long Range Wide Area Network*** [55]

Long Range Wide Area Network (LoRAWAN) es un protocolo de comunicación de baja potencia y baja tasa de transferencia de datos que ofrece una amplia cobertura en un área extensa. Se basa en una capa de enlace de datos con largo alcance, consumo eficiente de energía y velocidades de transmisión reducidas. Suele utilizarse en entornos con limitaciones de energía y donde la cobertura es crítica.

- ***Narrow band-Internet of Things*** [56]

Narrow band-Internet of Things (NB-IoT) es una tecnología de comunicación inalámbrica recientemente introducida para habilitar específicamente la conectividad entre dispositivos IoT. Está basada en LTE, con la cual comparte muchas funcionalidades. Puede utilizar tanto en redes 4G como en redes 5G.

2.2.4. Plataformas IoT

Una plataforma IoT es un conjunto de entidades habilitadas por la tecnología que incluye objetos físicos inteligentes (por ejemplo, sensores, actuadores, cámaras, etiquetas inteligentes y etiquetas de seguimiento), así como servicios y sistemas *software* que están conectados y funcionan juntos [57].

A continuación, se introducen las plataformas IoT más destacadas presentes en esta tesis.

FIWARE

FIWARE [58] es una plataforma IoT de código abierto impulsada por la Unión Europea y gestionada por FIWARE Foundation [59]. Su objetivo principal es facilitar la comunicación entre aplicaciones y dispositivos IoT. Para ello, ofrece una infraestructura de servicios, basada en bloques de construcción genéricos y reutilizables. Cada bloque utiliza un *Application Programming Interface* (API) estandarizada que permiten la gestión y el intercambio de datos de manera eficiente.

CAPÍTULO 2. ESTADO DEL ARTE

La información de contexto se gestiona a través de *Orion Context Broker*. Es el elemento central e indispensable de la plataforma. Habilita la comunicación entre los productores y los consumidores de la información mediante la API *Next Generation Service Interface* (NGSI). Los modelos de datos en FIWARE están diseñados de forma armonizada, lo que posibilita la transferencia de datos entre diversos ámbitos, como *smart robotics*, *smart agriculture* o *smart health* entre otros.

Con el objetivo de impulsar la creación de un modelo de datos común sobre IoT, recientemente FIWARE Foundation ha creado una nueva iniciativa llamada *Smart Data Models* [60]. Esta iniciativa libre modelos de datos con formato JSON/JSON-LD compatibles con las APIs NGSIv2/NGSI-LD de FIWARE así como con cualquier otra interfaz REST que cumpla con la especificación Open API.

IoTivity

IoTivity [61] es un proyecto de código abierto respaldado por *Open Connectivity Foundation* (OCF) [62] y *Linux Foundation* [63]. Ambas entidades trabajan juntas para promover un conjunto de directrices de interoperabilidad IoT. La plataforma implementa estándares OCF, sin limitarse únicamente a ellos, para proporcionar comunicaciones fáciles y seguras en los dispositivos IoT. La interoperabilidad se consigue mediante el uso de modelos de datos normalizados y consensuados por OCF. Estos modelos abarcan una amplia gama de verticales. IoTivity proporciona capacidades de *discovery*, transmisión de datos y gestión de dispositivos y datos. Este *framework* es independiente del dispositivo, utiliza un protocolo de aplicación restringida (CoAP) como capa de aplicación, pero requiere que la capa de red IP, siendo compatible con protocolos como Wi-Fi, *Ethernet*, *ZigBee*, *Bluetooth*, *Bluetooth Low-Energy* y *Z-wave*.

Apache sensiNact

La plataforma *sensiNact* [64], de la Fundación Eclipse [65], tiene como objetivo integrar y gestionar dispositivos IoT, recopilar sus datos y facilitar el desarrollo de aplicaciones. La plataforma utiliza un *broker* de mensajería MQTT que soporta protocolos *northbound* como HTTP REST, MQTT, XMPP, JSON RPC y CDMI y protocolos *southbound* como *Zigbee*, *EnOcean*, *LoRa* y *XBee*. Además, gestiona la heterogeneidad de los dispositivos a través de un modelo de datos unificado. También proporciona acceso síncrono (bajo demanda) y asíncrono (periódico o basado en eventos) a los datos de los sensores, así como acceso a los datos históricos (almacenados en InfluxDB [66]). Por último,

sensiNact gestiona los derechos de acceso a los dispositivos para usuarios de terceros.

La plataforma incluye la herramienta *sensiNact Studio*, que permite a los desarrolladores crear aplicaciones rápidamente sin necesidad de conocer los detalles subyacentes. Esto permite reducir el tiempo de desarrollo de las aplicaciones IoT además de facilitar su despliegue y gestión en tiempo de ejecución.

2.2.5. Arquitecturas de referencia

Una arquitectura de referencia es un diseño que se utiliza como una plantilla para guiar la definición de otras arquitecturas más específicas. Esta plantilla carece de detalles tecnológicos, pero incluye directrices de alto nivel a seguir, los bloques generales a implementar, así como sus relaciones además de recomendaciones tecnológicas. No hay una arquitectura estandarizada para implementación de IoT, aunque si existen numerosas propuestas de diversos investigadores y organizaciones [67]. Este hecho se debe a varias causas.

Por un lado, a menudo se presentan propuestas de arquitectura mediante el uso de diferentes tipos de sensores y tecnología de acuerdo a las necesidades concretas del propósito de los sistemas donde se implementan que luego son difíciles de trasladar a otros entornos. Por otro lado, pese a que tanto proyectos de investigación representativos como empresas privadas han propuesto arquitecturas de IoT de propósito general [68] y de dominio específico [69], estas no han llegado a convertirse en estándar pues, en lugar de abrir la infraestructura de IoT horizontalmente, muchas empresas están contribuyendo a su fragmentación mediante la creación de pilas verticales de productos integrados (*vendor lock-in* [70]). Como consecuencia, aunque el número de soluciones crece rápidamente se trata de una tecnología muy heterogénea donde ninguno de sus estándares parece ser dominante [71].

De forma general, una arquitectura IoT puede explicarse en tres o cinco niveles. Muchos investigadores coinciden en que el modelo de tres niveles es sencillo de entender, pero deja fuera algunos detalles clave sobre el uso de los datos.

Arquitecturas de tres y cinco niveles

La arquitectura de tres niveles (Figura 2.5a) se considera la arquitectura básica de IoT. Como su nombre indica, consta básicamente de tres niveles: nivel de percepción, nivel de red y nivel de aplicación. Esta clasificación está apoyada por el estándar IEEE P2413 [72]. El nivel de percepción se conoce como nivel físico porque contiene los dispositivos físicos y el nivel de detección debido a los sensores incorporados a los objetos físicos. En este nivel, los sensores que

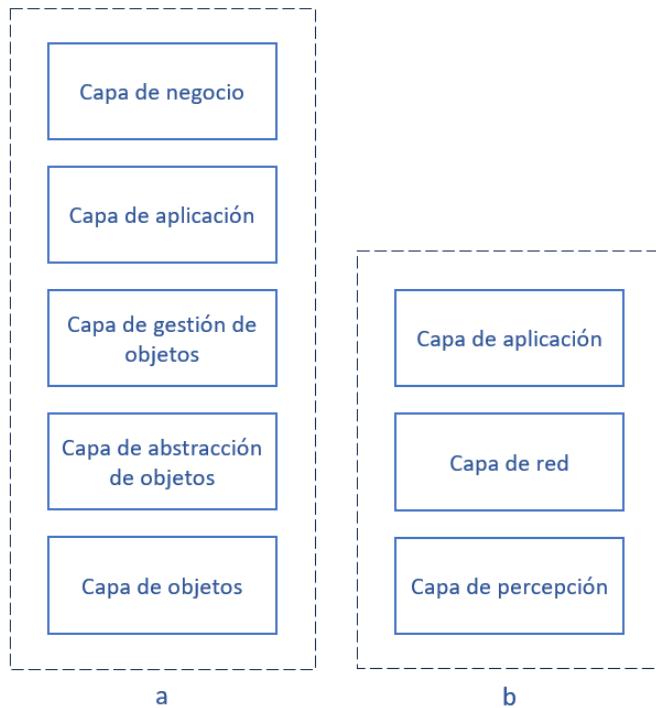


Figura 2.5: Arquitecturas IoT - a) tres niveles, b) cinco niveles.

la integran recogen los datos de los dispositivos y los envían a el nivel de red. El nivel de red se encarga de conectar el nivel de percepción con el nivel físico. Este nivel debe disponer de tecnologías de conexión como la conexión segura por cable o inalámbrica. El nivel de aplicación se considera el nivel superior en la arquitectura tradicional de IoT y proporciona servicios personalizados basados en las necesidades de los usuarios. Los datos recibidos por el nivel de aplicación son analizados y procesados para ofrecer servicios, enviando los resultados al nivel de percepción a través del nivel de red.

La Figura 2.5b presenta el modelo de cinco niveles:

■ Nivel de objetos

El primer nivel, llamado nivel de objetos, dispositivos o percepción representa los sensores físicos de IoT que tienen como objetivo recoger y procesar información. Este nivel incluye sensores y actuadores [73] para realizar diferentes funcionalidades como la consulta de la ubicación, temperatura, peso, aceleración, la humedad, etc. Los sensores son capaces de percibir el entorno que les rodea y recoger información que luego se transmite elementos superiores mientras que los actuadores son capaces de controlar o manipular el entorno físico. Algunos ejemplos son los *smartwatches* o las *Raspberry Pis* [74].

El nivel de percepción debe utilizar mecanismos estandarizados *plug-and-play* para configurar los objetos heterogéneos. Este nivel digitaliza y transfiere los datos a el nivel inmediatamente superior (nivel de abstracción de objetos) a través de canales seguros.

■ Nivel de abstracción de objetos

La abstracción de objetos transfiere los datos producidos por el nivel de objetos al nivel de gestión de servicios a través de canales seguros. Los datos pueden transferirse a través de diversas tecnologías como RFID, 3G, GSM, UMTS, WiFi, *Bluetooth Low Energy*, infrarrojos, *ZigBee*, etc. Además, en este nivel se gestionan otras funciones como la computación *cloud* y los procesos de gestión de datos [75].

■ Nivel de gestión de servicios

El nivel de gestión de servicios o *middleware* empareja un servicio con su solicitante basándose en direcciones y nombres. Este nivel permite a los programadores de aplicaciones IoT trabajar con objetos heterogéneos sin tener en cuenta una plataforma de hardware específica. Además, procesa los datos recibidos, toma decisiones y entrega los servicios requeridos a través de los protocolos de la red [76].

■ Nivel de aplicación

El nivel de aplicación proporciona los servicios solicitados por los usuarios finales. Cada sistema IoT cuenta con unos requisitos diferentes y por lo tanto cada sistema define unas metas y objetivos particulares. Por ejemplo, en el caso una vivienda inteligente el nivel de aplicación puede proporcionar una interfaz de usuario que muestre el estado de los sensores IoT instalados en la vivienda. El nivel de aplicación abarca todos los verticales de IoT, como el hogar inteligente, los edificios inteligentes, el transporte, la automatización industrial o la atención sanitaria inteligente [3], [17] - [19].

■ Nivel de negocio

El nivel de negocio gestiona las actividades y servicios generales de los sistemas IoT. Las responsabilidades de este nivel son construir un modelo de negocio, gráficos y diagramas de flujo que utilizan los datos recibidos del nivel de aplicación. También diseña, analiza, implementa, evalúa, supervisa y desarrolla los elementos relacionados con el sistema IoT. El nivel de negocio permite apoyar los procesos de toma de decisiones basados en el análisis de *big data*. Además, en este nivel se realiza la supervisión y gestión de los cuatro niveles subyacentes. Igualmente, este nivel compara el resultado de cada nivel con el resultado esperado para mejorar los servicios y mantener la privacidad de los usuarios.

Además de las arquitecturas de referencia de tres y cinco capas, existen otras arquitecturas que han ganado reconocimiento en el ámbito de la industria y la investigación. Dos ejemplos destacados son *Reference Architectural Model Industrie 4.0* (RAMI 4.0) e *Industrial Internet Reference Architecture* (IIRA).

RAMI 4.0

RAMI 4.0 [77] es uno de los primeros *frameworks* propuestos para definir una imagen común de la industria 4.0. Fue propuesto por una iniciativa alemana de normalización con el mismo nombre. Su objetivo principal es proporcionar normas, términos y reglas comunes para describir los requisitos y las estructuras relativas a los diferentes casos de uso de las fábricas inteligentes. Desglosa los elementos más críticos que se encuentran en el entorno de la Industria 4.0 en un sistema de coordenadas tridimensional, cuyos ejes corresponden a:

1. Los niveles jerárquicos del sistema de fabricación.
2. El ciclo de vida de los sistemas y productos, incluyendo tanto el desarrollo como el mantenimiento.

3. Los niveles de una fábrica inteligente.

El eje *layers* se utiliza para describir los diferentes elementos de la industria 4.0 presentes en una fábrica inteligente típica. De abajo a arriba, son:

- **Nivel de activos:** Describe los recursos físicos que se encuentran en la planta de la fábrica, por ejemplo, equipos, piezas, etc.
- **Nivel de integración:** Se refiere a los medios de comunicación entre las tecnologías de la información y de la operación.
- **Nivel de comunicación:** Se refiere al tipo de protocolos y canales de comunicación utilizados para la transmisión de datos conforme a las normas.
- **Nivel de información:** Se refiere a los medios de almacenamiento de la información (datos).
- **Nivel funcional:** Se refiere a los procesos de toma de decisiones para establecer las funciones que deben desempeñar los activos de la fábrica.
- **Nivel de negocio:** Se refiere a los procesos organizativos y empresariales.

Industrial Internet Reference Architecture

IIRA [78] es una iniciativa liderada por los Estados Unidos para establecer un marco de referencia común para el desarrollo de soluciones de *Industrial Internet of Things* (IIoT). Está dirigida a las fábricas inteligentes de diferentes sectores industriales y ha sido desarrollado por *Industrial Internet Consortium* (IIC). El objetivo de IIRA es proporcionar orientación para la especificación de arquitecturas de sistemas, soluciones y aplicaciones de IIoT. El documento donde se especifica se actualiza continuamente, siendo la novena versión (v1.9) la última publicada.

IIRA no implica ninguna implementación específica, sino que organiza los casos de uso de IIoT en cuatro niveles (denominados “vistas”), que categorizan los aspectos que el sistema debe abordar en cada caso de uso. Cada vista se utiliza para analizar necesidades y requisitos específicos. Las decisiones tomadas en una vista concreta guían la forma en que se definen los requisitos y las necesidades en la vista siguiente. Del mismo modo, la eventual implementación sugerida en las vistas inferiores sirve como medio para validar y proporcionar retroalimentación a las vistas superiores, introduciendo potencialmente revisiones. Además de enmarcar las operaciones en estas cuatro vistas, IIRA proporciona extensibilidad por medio de puntos de vistas personalizadas que pueden

ser definidas por los arquitectos de sistemas. Esto le permite adaptarse a los requisitos específicos de su propio caso de uso.

Los cuatro puntos de vista que forman el núcleo de IIRA se organizan de la siguiente forma:

- **Vista de negocio**

Esta vista está relacionada con la toma de decisiones a nivel empresarial. En ella los objetivos empresariales específicos se alinean con las capacidades del sistema. Las preocupaciones que se abordan en este punto de vista se refieren, por ejemplo, al retorno de la inversión (ROI), los costes de mantenimiento, etc. Las decisiones que se toman en este punto de vista repercuten en los diseños de la vista “uso”.

- **Vista de utilización**

Esta vista está relacionada con el uso previsto del sistema. Se ocupa de la forma en que se construye y se estructura el sistema IIoT en relación con las actividades de sus componentes (es decir, tanto los recursos humanos como los del sistema). Básicamente, sugiere los casos de uso (en términos de tareas realizadas por una entidad o “rol” en el sistema) que guían la forma en que deben diseñarse los componentes de la vista funcional.

- **Vista funcional**

La vista funcional se utiliza para especificar los componentes funcionales que conforman el sistema global. Además, define las interacciones, las funciones y las responsabilidades de cada uno de ellos. Esta vista forma la columna vertebral de la arquitectura del sistema y guía las actividades definidas en la vista de implementación.

- **Vista de implementación**

La vista de implementación describe la tecnología necesaria para implementar los bloques funcionales identificados en la vista funcional. Especifica tanto los elementos de implementación e integración del sistema, como los protocolos de comunicación de conectividad, las consideraciones de despliegue etc.

2.2.6. Interoperabilidad en IoT

La heterogeneidad de los dispositivos IoT puede presentarse en términos de sus capacidades, características, proveedores o requisitos específicos de sus aplicaciones. Además, existe gran diversidad de tecnologías, protocolos, formatos de datos, semántica, plataformas IoT y estrategias de procesamiento utilizadas.

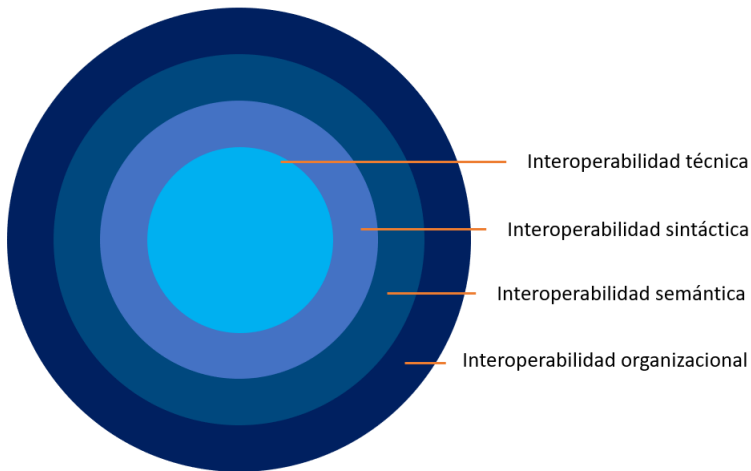


Figura 2.6: Las dimensiones de la interoperabilidad.

Esta naturaleza heterogénea de los sistemas IoT impide a los usuarios consumir los servicios de los dispositivos de forma correcta y eficiente.

Para proporcionar una comunicación fluida entre los dispositivos IoT de los diferentes proveedores, una posible solución sería que todos los proveedores, investigadores, ciudadanos y agencias gubernamentales acordaran utilizar un estándar común de comunicación. También deben estandarizarse aspectos como la creación APIs y la definición de servicios estandarizados que ayuden en un futuro próximo a desplegar las aplicaciones en un entorno uniforme.

Pese a que existen muchas definiciones de interoperabilidad, hay funcionalidades comunes entre ellas; i) obtener información, ii) intercambiar datos, y iii) utilizar la información para comprenderla y poder procesarla. La interoperabilidad puede aportar soluciones para lograr la uniformidad en presencia de la heterogeneidad en varios niveles, como el técnico, el sintáctico y el semántico [79]. Una representación simple de los distintos niveles de interoperabilidad puede verse en la Figura 2.6. Sus dimensiones son las siguientes:

- La **interoperabilidad técnica** se asocia con componentes *hardware/software*, sistemas y plataformas que permiten la comunicación máquina a máquina. Este tipo de interoperabilidad suele centrarse en los protocolos y la infraestructura necesaria para que estos funcionen.
- La **interoperabilidad sintáctica** se asocia con los formatos de datos. Los mensajes transferidos por los protocolos de comunicación deben tener

CAPÍTULO 2. ESTADO DEL ARTE

una sintaxis y codificación bien definidas. Sin embargo, cada protocolo representa los datos utilizando sintaxis diferentes, como HTML o XML.

- La **interoperabilidad semántica** se asocia con el significado del contenido y concierne a la interpretación humana en lugar de la máquina del contenido. Por lo tanto, la interoperabilidad en este nivel implica que hay una comprensión común entre las personas del significado de la información que se intercambia.
- La **interoperabilidad organizacional** es la capacidad de las organizaciones para comunicarse y transferir información de manera efectiva. Esta debe darse, aunque la organización utilice una variedad de sistemas de información diferentes sobre infraestructuras heterogéneas, e incluso aunque se encuentre en regiones geográficas distintas. La interoperabilidad organizacional depende de una interoperabilidad técnica, sintáctica y semántica exitosa.

Durante los últimos años se han llevado a cabo varios esfuerzos de estandarización para definir una arquitectura para la interoperabilidad de los sistemas IoT. Algunos de los más destacados son *Alliance for the Internet of Things Innovation* (AIOTI), INTER-IoT y oneM2M.

AIOTI

AIOTI [80] es un organismo de estandarización lanzado por la Comisión Europea. Su objetivo principal es apoyar y estandarizar el desarrollo del ecosistema IoT en Europa. Se centra especialmente en superar las barreras que existen entre las principales áreas verticales del IoT, permitiendo una mayor interoperabilidad y colaboración. En particular, busca romper los silos de información entre las principales áreas verticales IoT.

La arquitectura desarrollada por AIOTI recibe el nombre de *AIOTI High Level Architecture* (HLA). Esta busca fomentar el uso de estándares y prácticas compartidas. Su objetivo es lograr una integración sin complicaciones entre dispositivos y sistemas. Al romper las barreras de información entre las distintas áreas verticales del IoT, AIOTI busca impulsar la innovación, acelerar el desarrollo de aplicaciones y servicios, y mejorar el impacto positivo del IoT en diversos sectores de Europa, tanto en la industria como en la sociedad en general.

INTER-IoT

INTER-IoT [81] es un proyecto europeo financiado por la iniciativa Horizonte 2020. Proporciona un *framework cloud* que habilita la interoperabilidad

entre diferentes plataformas IoT. A diferencia de otras soluciones de interoperabilidad, INTER-IoT ofrece un enfoque multicapa que integra diferentes dispositivos, redes, plataformas, servicios y aplicaciones, con el objetivo de lograr un continuo global de datos, infraestructuras y servicios que puedan permitir diferentes escenarios IoT. Además, INTER-IoT reutiliza e integra los sistemas IoT existentes y futuros, creando un ecosistema de plataformas IoT interoperables. En ausencia de estándares IoT globales, INTER-IoT permite a las empresas, usuarios o desarrolladores diseñar y desarrollar nuevos dispositivos o servicios IoT, aprovechando el ecosistema existente, e introducirlos rápidamente en el mercado. El *framework* ofrecido por INTER-IoT es conceptualmente genérico y puede extenderse a un gran número de dominios de aplicación diferentes, proporcionando un marco *cloud* para hacer interoperables las plataformas IoT y los objetos inteligentes que operan en el mismo o en diferentes dominios de aplicación.

oneM2M

La iniciativa oneM2M [82] pretende cubrir estos aspectos y ha publicado recientemente sus especificaciones, que definen una capa *software* asentada entre las aplicaciones M2M y el *hardware/software* de comunicación y proporciona transporte de datos.

oneM2M proporciona un *middleware* o capa de servicios IoT que proporciona una serie de funciones a través de APIs uniformes, independientes del proveedor y estandarizadas globalmente, hacia las aplicaciones IoT. Entre las principales características proporcionadas por este conjunto de funciones se encuentran la identificación, autenticación y autorización de usuarios y aplicaciones, el cifrado de datos de extremo a extremo, la gestión de dispositivos, el aprovisionamiento remoto y la activación de servicios, la configuración de la conectividad y la programación de la transmisión de datos, la gestión de grupos y las funciones de descubrimiento de aplicaciones y datos, la agregación de datos, el almacenamiento en búfer en caso de falta de conectividad y la activación cuando se restablece.

La arquitectura utilizada en la norma oneM2M utiliza un modelo de tres niveles: aplicaciones, servicios y redes. El nivel de aplicación reside dentro de las aplicaciones de los dispositivos y proporciona una interfaz estandarizada para la administración. Entre el nivel de aplicación y el nivel de red se encuentra el nivel de servicio, compuesta por entidades de servicio común que desempeñan un papel similar al expuesto en el nivel de aplicación.

2.2.7. Otros temas de investigación abiertos

IoT continúa presentando retos que hasta la fecha no han sido resueltos. A continuación, se presentan los desafíos y líneas de investigación que no han sido presentadas hasta el momento y que guardan relación con la presente tesis.

■ Escalabilidad

La escalabilidad es un gran reto en IoT [83]. Un sistema IoT conecta varios sensores, actuadores y otros dispositivos para permitir el intercambio de información y un gran número de aplicaciones a través de Internet. Esto supone un desafío para el diseño y el crecimiento del sistema para cumplir con la escalabilidad y la adaptabilidad a las necesidades digitales en continuo cambio y evolución.

La escalabilidad significa adaptarse a los cambios y crecer con ellos, así como satisfacer las necesidades específicas cuando surgen. Un sistema escalable es aquel que, tras añadir nuevos dispositivos, permite que el sistema funcione con fluidez, sin retrasos indebidos ni recursos improductivos, y haciendo un uso eficiente de los recursos disponibles. Es deseable que una red pueda continuar gestionando sus recursos con eficiencia cuando estos crecen en cantidad.

■ Cloud IoT

Los dispositivos IoT generan grandes volúmenes de datos que generalmente se envían a la nube para su procesamiento. La distancia entre la nube y los dispositivos IoT provoca retrasos en el envío y la recepción de datos [84]. Asimismo, la recepción tardía de los datos provoca retrasos a la hora de obtener el resultado de los análisis y las posibles acciones asociadas. Así, los sistemas IoT que dependen de la nube para trabajar con aplicaciones de tiempo crítico ven comprometido su rendimiento.

Otro problema intrínseco de los sistemas *cloud-IoT* es que a menudo necesitan la entrada de una amplia gama de dispositivos para ser procesados e implementados. En esta situación, los requisitos de espacio de almacenamiento y la capacidad de computación *cloud* pueden ser muy exigentes.

■ Edge IoT

Debido al crecimiento exponencial en la cantidad de datos que generan los dispositivos digitales, el modelo tradicional de procesamiento y almacenamiento de los datos *cloud* se está volviendo demasiado costoso y lento especialmente para las aplicaciones en tiempo real. Para poner fin a esta problemática, ha surgido la computación *edge*, un nuevo paradigma de

computación descentralizado donde el procesamiento de los datos es más rápido ya que ocurre en la periferia de la red (en el llamado *edge*). Sin embargo, este nuevo paradigma presenta algunos problemas como la falta de seguridad en la transferencia de datos [85]. Además, el *edge* no proporciona un almacenamiento de datos grande como la nube y el tiempo de almacenamiento de los datos es limitado [86].

■ Seguridad, privacidad y gestión de los datos personales

Las aplicaciones de la red IoT recogen datos de los sensores/dispositivos y los envían para su análisis y procesamiento a nodos *edge* o *cloud*. La privacidad de los sistemas es crucial [87], ya que los datos que se transfieren entre los dispositivos pueden contener información sensible, como información sanitaria de un paciente, secretos comerciales de una empresa o información clasificada de una red estatal o militar.

Además de la privacidad, los problemas de seguridad, como los ataques de enrutamiento basados en la red y los ataques de *botnets*, pueden interrumpir los servicios de IoT [88], [89]. Igualmente, las aplicaciones IoT requieren una gestión de la confianza para una fusión de datos fiable y una mayor seguridad de la información [90]. Por ello, la gestión de la seguridad de IoT es fundamental para garantizar la seguridad de las redes y la transmisión eficiente de los datos. Sin embargo, la funcionalidad de la seguridad se vuelve aún más difícil debido a la naturaleza heterogénea de estas redes equipadas con dispositivos IoT con limitaciones de recursos [91]. Por lo tanto, los sistemas tradicionales de seguridad son ineficaces y requieren ser adaptados al marco de las redes IoT creando mecanismos innovadores para hacer frente a los desafíos específicos en la gestión de la seguridad.

2.3. Inteligencia Artificial

2.3.1. Perspectiva general

La IA se refiere a la capacidad de un ordenador o de un sistema robótico con ordenador para procesar información y producir resultados de forma similar al proceso de pensamiento de los humanos en el aprendizaje, la toma de decisiones y la resolución de problemas. Esta tecnología se sitúa en el centro de lo que se ha denominado la “cuarta revolución industrial” [92], que se distingue por el desplazamiento de la toma de decisiones y el control de los seres humanos a la tecnología.

El objetivo del desarrollo de técnicas de IA es lograr la automatización de comportamientos inteligentes que abarcan principalmente seis áreas: ingeniería

CAPÍTULO 2. ESTADO DEL ARTE

del conocimiento, razonamiento, planificación, comunicación, percepción y movimiento [93]. Se trata de un campo en constante desarrollo y cambio que está formado, a su vez, por gran cantidad de subcampos en constante evolución. No existe una clasificación única de los campos que abarca. A continuación, se lista la clasificación propuesta por Frankish *et al.* [94]:

- *Knowledge representation*
- *Heuristic search*
- *Planning*
- *Expert systems*
- *Machine vision*
- *Machine learning*
- *Natural language processing*
- *Software agents*
- *Intelligent tutoring systems*
- *Robotics*

En su informe anual de 2017, Gartner indicó que las expectativas en el área de ciencia de datos y ML se habían elevado en el último año y se encontraban en el pico. Identificó la IA como una de las mega tendencias tecnológicas emergentes y señaló que, en el ámbito empresarial, los arquitectos y los líderes en innovación tecnológica deberían explorar y analizar la IA para comprender el impacto futuro en su negocio e incorporarla en su modelo de negocio. Para la mayoría de las organizaciones, la integración de IA en sus aplicaciones y flujos de trabajo es un reto complejo y de elevado coste. Pese a esto, en 2021 Gartner señaló que, gracias a la rápida madurez detectada en las iniciativas de orquestación de IA, para 2025 el 70 % de las organizaciones habrán puesto en marcha arquitecturas de IA. También es interesante destacar que el Ministerio de Asuntos Económicos y Transformación Digital, a través de la Estrategia España Digital 2026 [95], ha fijado como uno de principales objetivos “impulsar la Inteligencia Artificial como motor de innovación y crecimiento económico social, inclusivo y sostenible”.

ML es una aplicación específica o subconjunto de la IA que permite a las máquinas aprender de datos sin necesidad de ser programadas explícitamente [96]. Esta área explora el estudio y la construcción de algoritmos que pueden aprender de los datos y hacer predicciones en base a ellos. ML se emplea en

aquellas tareas informáticas en las que el diseño y la programación de algoritmos explícitos con buen rendimiento es difícil o inviable. Está estrechamente relacionado (y a menudo se solapa) con la estadística computacional, que también se centra en la elaboración de predicciones mediante el uso de ordenadores. Tiene fuertes vínculos con la optimización matemática, que aporta métodos, teoría y dominios de aplicación al campo. También se confunde a veces con la minería de datos, donde este último subcampo se centra más en el análisis exploratorio de datos. Algunos ejemplos de aplicación son el filtrado de correo electrónico [97], la detección de intrusos en las redes [98], el reconocimiento de caracteres [99], los asistentes virtuales [100] o los sistemas de recomendación [101].

ML y DL han ganado protagonismo en los últimos años [102] y en ocasiones su frontera es confusa. Ambos métodos “capturan patrones de datos brutos complicados y multidimensionales y los utilizan como características discriminatorias de los datos” [103]. ML comprende algoritmos y modelos estadísticos utilizados por las máquinas para realizar tareas y recopilar información mejorando con la experiencia, mientras que DL imita las vías neuronales del cerebro humano en el procesamiento de datos, utilizándolos para la toma de decisiones, la detección de objetos, el reconocimiento del habla y la traducción de idiomas. DL aprende sin supervisión ni intervención humana, a partir de datos no estructurados y sin etiquetar. Además, imita el modo de funcionamiento del cerebro humano, organizando sus elementos en capas y formando redes neuronales para procesar datos y generar patrones para tomar decisiones sobre los problemas planteados. Las redes neuronales interpretan mejor los datos desestructurados y mejoran su eficiencia en función de la cantidad de datos. No ocurre lo mismo con los algoritmos de ML, que tienen un punto de saturación: es decir, llegado un momento, aunque se les proporcionen más datos, no mejoran su eficiencia. Por esto, DL se utiliza para solventar los problemas más complejos y que requieren más datos, como la clasificación de objetos, la detección de defectos, NLP o la transformación de voz humana en texto (*speech to text*) entre otras aplicaciones.

2.3.2. Definición

En la comunidad científica no existe una definición clara y consensuada de IA o ML. De forma general se entiende IA como un conjunto de técnicas destinadas a aproximar algún aspecto de la cognición humana o animal utilizando máquinas.

A continuación, se incluyen definiciones del término IA aportadas por distintos organismos de estandarización:

CAPÍTULO 2. ESTADO DEL ARTE

- IEEE-USA define la IA como: “La teoría y el desarrollo de sistemas informáticos capaces de realizar tareas que normalmente requieren inteligencia humana, como la percepción visual, el reconocimiento del habla, el aprendizaje, la toma de decisiones y el procesamiento del lenguaje natural” [104].
- La ETSI ofrece la siguiente definición: “Sistema informático que utiliza la cognición para comprender la información y resolver problemas” [105].
- La definición de la Comisión Europea es la siguiente: “Los sistemas de inteligencia artificial son sistemas *software* (y posiblemente también *hardware*) diseñados por seres humanos que, dado un objetivo complejo, actúan en la dimensión física o digital percibiendo su entorno mediante la adquisición de datos, interpretando los datos estructurados o no estructurados recogidos, razonando sobre los conocimientos, o procesando la información, derivados de estos datos y decidiendo la mejor acción o acciones a realizar para lograr el objetivo dado. Los sistemas de IA pueden utilizar reglas simbólicas o aprender un modelo numérico, y también pueden adaptar su comportamiento analizando cómo se ve afectado el entorno por sus acciones anteriores. Como disciplina científica, la IA incluye varios enfoques y técnicas, como el aprendizaje automático (del que el aprendizaje profundo y el aprendizaje por refuerzo son ejemplos específicos), el razonamiento automático (que incluye la planificación, la programación, la representación del conocimiento y el razonamiento, la búsqueda y la optimización) y la robótica (que incluye el control, la percepción, los sensores y los actuadores, así como la integración de todas las demás técnicas en sistemas ciberfísicos)” [106].

También se incluyen definiciones del término IA propuestos por diversos autores:

- Alan Turing definió la IA de la siguiente manera: “Si hay una máquina detrás de una cortina y un humano está interactuando con ella (por cualquier medio, por ejemplo, audio o a través de la escritura, etc.) y si el humano siente que está interactuando con otro humano, entonces la máquina es artificialmente inteligente”.
- Fue definida por John McCarthy en 1956, como “Conjunto de técnicas que sirven para dotar de inteligencia a una máquina” [107].
- Haugeland definió la IA de la siguiente forma: “Los nuevos y emocionantes esfuerzos por hacer que los ordenadores piensen (...) máquinas con mentes, en el sentido pleno y literal” [108].

- Bellman lo define como: “La automatización de actividades que asociamos con el pensamiento humano, actividades como la toma de decisiones, la resolución de problemas, el aprendizaje (...)” [109].
- Charniak y McDermott lo definen como: “El estudio de las facultades mentales mediante el uso de modelos computacionales” [110].

Seguidamente se listan múltiples definiciones del término ML proporcionadas por distintos autores:

- ML fue definido por Tom M. Mitchell como: “El estudio de los algoritmos informáticos que permiten a los programas de ordenador mejorar automáticamente a través de la experiencia” [111].
- “*Machine learning* es la capacidad de los ordenadores de ajustar su comportamiento en función de los datos a los que están expuestos” [112].
- Arthur Samuel definió el aprendizaje automático como: “Un campo de estudio que da a los ordenadores la capacidad de aprender sin ser programados explícitamente” [113].

2.3.3. Tipos de aprendizaje

ML permite a los sistemas de computación hacer predicciones o tomar decisiones sin necesidad de intervención humana. A fin de dotar a los sistemas de inteligencia, los algoritmos son entrenados utilizando cantidades masiva de datos (ya sean estructurados o semiestructurados). Los sistemas extraen conocimiento de los datos proporcionados, aprendiendo de ellos. Durante este proceso se entrenan diversos algoritmos y se escoge aquel que es capaz realizar predicciones con la mayor exactitud posible. De esta forma se obtiene lo que se denomina modelo.

Los enfoques tradicionales de ML distinguen tres tipos de aprendizaje: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo.

■ Aprendizaje supervisado

En el aprendizaje supervisado los datos de entrenamiento proporcionados al algoritmo son datos etiquetados. El sistema crea un modelo utilizando los datos etiquetados para comprender los *datasets* y aprender sobre ellos. En concreto, el algoritmo aprende las relaciones entre los parámetros de entrada (variables independientes) y el parámetro de salida (variable dependiente) basándose en los datos de entrenamiento y la retroalimentación humana. Una vez finalizado el entrenamiento, se valida el

rendimiento del modelo resultante. El objetivo principal del aprendizaje supervisado es relacionar los datos de entrada con los de salida.

El aprendizaje supervisado puede agruparse en dos categorías de algoritmos:

- **Algoritmos de clasificación:** Estos algoritmos realizan clasificaciones binarias o multiclase. El ejemplo clásico se basa en clasificar tres especies de la flor Iris (*Iris setosa*, *Iris virginica* e *Iris versicolor*) a partir de sus características físicas (largo y ancho del sépalo y pétalo).
- **Algoritmos de regresión:** Los algoritmos de regresión son empleados para generar modelos capaces de predecir un valor numérico continuo. Un ejemplo habitual en la literatura consiste en predecir el valor de una vivienda dadas sus características (número de habitaciones, índice de criminalidad de la zona etc.).

■ Aprendizaje no supervisado

En el aprendizaje no supervisado los algoritmos extraen tendencias y patrones de los datos de entrenamiento sin supervisión humana. Los datos de entrenamiento no están etiquetados o clasificados de ninguna forma. El objetivo de este tipo de aprendizaje es estructurar o categorizar los datos de entrada en categorías o grupos que compartan patrones o características similares. Por lo tanto, no es posible conocer con antelación la cantidad de categorías que el algoritmo va a detectar.

Frecuentemente los algoritmos detectan subcategorías que inicialmente no se contemplaban o que incluso pueden ser no válidas para el objetivo del análisis, por lo que el posterior análisis e interpretación de los resultados es de gran importancia. Se trata de un método menos preciso y fiable que el aprendizaje supervisado.

■ Aprendizaje por refuerzo

El aprendizaje por refuerzo es un tipo especial de aprendizaje que es tratado por separado a los métodos de aprendizaje supervisado y no supervisado. En este tipo de método están involucrados dos componentes: los agentes y el ambiente. El objetivo de este método es que un agente aprenda a operar en un entorno a partir de datos obtenido del mismo utilizando la retroalimentación. El agente obtiene una recompensa por cada acción correcta y recibe una penalización por cada acción incorrecta. El sistema no cuenta con un *dataset* de entrenamiento fijo, sino que continuamente interactúa con el entorno aprendiendo de él.

2.3.4. Algoritmos

Los algoritmos de ML son métodos matemáticos y lógicos que son capaces de realizar predicciones a partir de unos datos de entrada, generando un modelo. La elección del algoritmo depende de múltiples factores como el tipo de problema que se desea resolver, los recursos informáticos disponibles o la naturaleza de los datos. En la presente Sección se detallan los algoritmos más relevantes.

Linear Regression

Linear Regression (LR) es un ejemplo clásico de los modelos estrictamente lineales. También se denomina ajuste polinómico y es uno de los métodos lineales más sencillos en el aprendizaje automático. Este algoritmo busca ajustar una línea recta a través de los datos que mejor represente la relación entre las variables involucradas. Esta línea recta se determina mediante una fórmula matemática que minimiza la suma de los errores cuadrados entre los valores predichos y los valores reales de la variable dependiente.

Es importante tener en cuenta que este algoritmo asume una relación lineal entre las variables, lo que puede no ser adecuado para todos los casos.

Decision Tree

Decision Tree (DT) es un enfoque de aprendizaje automático supervisado empleado para resolver problemas de clasificación y regresión mediante la división continua de los datos en función de un determinado parámetro. Cada árbol está formado por nodos y ramas. Las decisiones están en las ramas y los datos se dividen en los nodos. En los problemas de clasificación, la variable de decisión es categórica y en los problemas de regresión la variable de decisión es continua.

Tiene las siguientes ventajas: es adecuado tanto para problemas de regresión como de clasificación, es fácil de interpretar, es fácil de manejar valores categóricos y cuantitativos, es capaz de rellenar los valores que faltan en los atributos con el valor más probable y tiene un alto rendimiento debido a la eficacia del algoritmo de recorrido del árbol. Sin embargo, los problemas de sobreajuste son habituales. En este caso, sería recomendable aplicar otros modelos más complejos.

Random Forest

Random Forest (RF) es un método conjunto que combina árboles de decisión y agregación *bootstrap*. Su objetivo es lograr una mayor precisión en com-

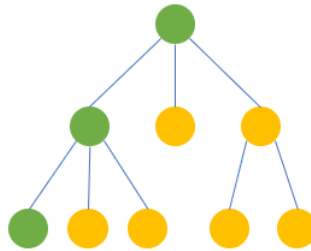


Figura 2.7: Visualización de la estructura del algoritmo *Decision Tree*.

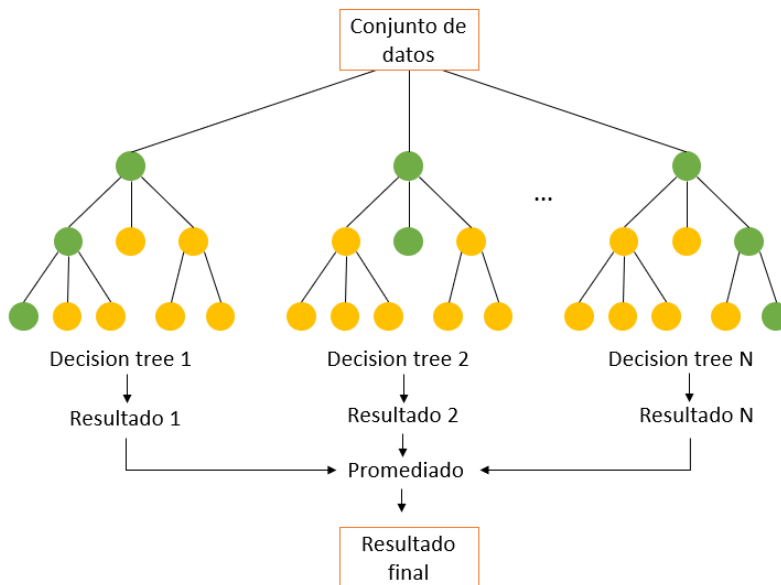


Figura 2.8: Visualización de la estructura del algoritmo *Random Forest*.

paración con los algoritmos individuales que lo componen. Como su nombre indica, el árbol de decisión construye un modelo de clasificación en la estructura de un árbol con un conjunto de reglas *if-then*. Su estructura se presenta en la Figura 2.8. Cada árbol de decisión produce una decisión. El resultado se obtiene teniendo en cuenta todas las decisiones parciales (votación por mayoría o promedio).

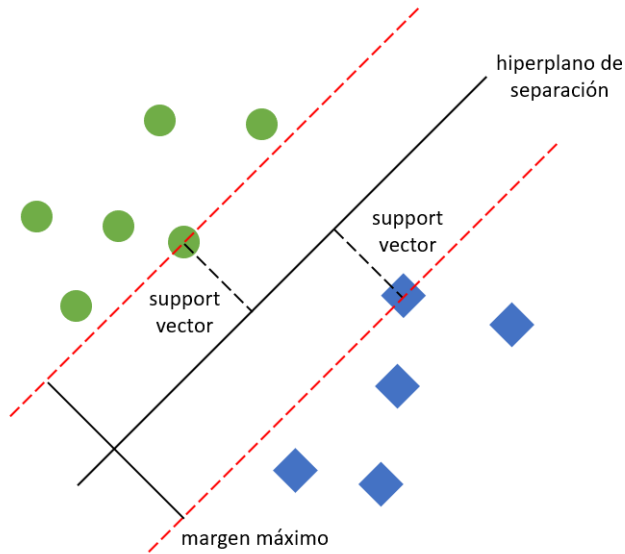


Figura 2.9: Visualización del algoritmo *Support Vector Machines*.

Como es de esperar, cuanto mayor sea el número de árboles, mayor será la precisión y menor la velocidad. Un número de 100 árboles parece ser suficiente para las aplicaciones típicas. El algoritmo RF es superior a los árboles de decisión simples porque supera los problemas de sobreajuste de forma intrínseca y no requiere un ajuste de los parámetros. Por otro lado, RF es generalmente un clasificador lento y, por lo tanto, no está pensado para aplicaciones en tiempo real.

Support Vector Machines

Support Vector Machine (SVM) es otro algoritmo de ML que se utiliza tanto para la clasificación como para la regresión. El objetivo del algoritmo es encontrar un hiperplano que, en el mejor grado posible, separe los puntos de datos de una clase de los de otra clase. El algoritmo encuentra el hiperplano óptimo que maximiza el margen entre el hiperplano y los vectores de soporte de las diferentes clases, como se muestra en la Figura 2.9. El entrenamiento de un algoritmo SVM corresponde a un problema de optimización cuadrática para ajustar un hiperplano, minimizando el margen entre las clases.

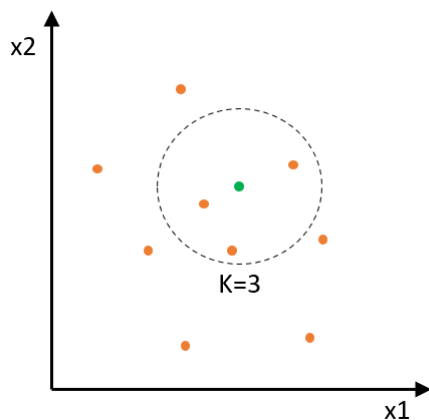


Figura 2.10: Visualización del algoritmo k -Nearest Neighbor.

K-Nearest Neighbor

El algoritmo K -Nearest Neighbor (KNN) no es un ejemplo de método lineal, pero es uno de los algoritmos más sencillos de comprender y puede utilizarse como clasificador o regresor. Este algoritmo no asume ningún tipo de ecuación ni ningún tipo de relación funcional entre la entrada y la salida.

La Figura 2.10 ilustra el modo de proceder del algoritmo en el caso de una tarea de regresión. La Figura muestra una distribución de datos de ejemplo donde los puntos naranjas representan los datos de entrenamiento el punto verde el valor a predecir. Considerando que se utiliza el valor de $k=3$, el algoritmo busca el valor de la variable a predecir de los 3 vecinos más cercanos, promedia ese valor y el resultado obtenido es la predicción.

2.3.5. Frameworks de IA

Los *frameworks* de IA proporcionan a los científicos de datos, desarrolladores e investigadores los componentes básicos para diseñar, entrenar, validar y desplegar modelos a través de una interfaz de programación de alto nivel.

A continuación, se describen los *frameworks* de IA más importantes y reconocidos, tanto en el ámbito industrial como de investigación:

- **TensorFlow** [114] es un *framework* de ML de código abierto desarrollado por Google que se centra en las redes neuronales y DL. Ofrece un ecosistema completo y flexible de herramientas (que se basan principalmente en gráficos y visualización de datos), bibliotecas y recursos de la comunidad.

TensorFlow se distribuye bajo la licencia Apache 2.0 y está disponible para las plataformas móviles y de escritorio más comunes. Dispone de APIs estables para Python y C++, así como de APIs de terceros para varios otros lenguajes. Dado que TensorFlow fue desarrollado como un sistema modular, es posible utilizar sólo algunas partes del *framework* de forma independiente.

- **Keras** [115] es una librería Python de código abierto para la creación de redes neuronales. Keras ha sido diseñada como complemento a otras librerías de ML para ofrecer herramientas de alto nivel más intuitivas para desarrollar aplicaciones de IA. Puede ejecutarse sobre TensorFlow, Theano o CNTK. Keras soporta una amplia gama de capas de redes neuronales, como capas densas, capas recurrentes y capas convolucionales.
- **Microsoft Cognitive Toolkit (CNTK)** [116] es un *framework* de trabajo de código abierto para redes neuronales. Proporciona una API de bajo nivel, destinada a la creación de componentes de redes neuronales, y una API de alto nivel, que contiene un conjunto de bloques de construcción para la definición y el entrenamiento de redes neuronales. También incluye componentes que facilitan la gestión de *datasets* grandes. Dispone de APIs para Python, C++ y C#.
- **Caffe** [117] es un *framework* de DL de código abierto orientado al procesamiento de imágenes implementado en C++ y cuenta con una API de Python. Caffe fue desarrollado por *Berkeley AI Research, The Berkeley Vision and Learning Center* y colaboradores de su comunidad. Es ligero, escalable y rápido y ofrece modelos preentrenados para fines de demostración.
- **Scikit-learn** [118] es una biblioteca de aprendizaje automático en Python de código abierto construida sobre *Scientific Python* [119]. Soporta el aprendizaje supervisado y no supervisado. Cuenta con una comunidad grande y activa lo cual la hace una opción muy atractiva dentro de las opciones *open source*.
- **PyTorch** [120] es un paquete de Python para el cálculo de tensores y la construcción de redes neuronales profundas. Ha sido diseñado para ser eficiente, intuitivo y fácil de usar. Ofrece gráficos de computación dinámicos, que pueden gestionar entradas y salidas que varían en longitud. Torch [121], fue desarrollado originalmente en la Universidad de Nueva York y luego contó con un gran número de paquetes aportados por la comunidad;
- **Deeplearning4j** [122] es un *framework* de DL para Java y Scala. Se integra con Hadoop y Apache Spark y proporciona métodos para la definición

y el entrenamiento de redes neuronales profundas. Este *framework* tiene varios submódulos, que cubren diferentes aspectos, como el entrenamiento del modelo o la visualización.

- **Theano** [123] es una librería de Python que permite la definición, evaluación y optimización de expresiones matemáticas. Está basada en Python y ha sido uno de los *frameworks* de IA más utilizados. Sin embargo, su desarrollo se ha detenido.

2.3.6. Artificial Intelligence as a Service

Con el fin de automatizar y proporcionar inteligencia a los sistemas, la integración de las capacidades de IA en las soluciones TIC se está convirtiendo en algo esencial. En consecuencia, las plataformas y *frameworks* que proporcionan este tipo de servicios han ido evolucionando, ofreciendo servicios cada vez más sofisticados y precisos.

En este contexto ha aparecido *AI as a Service* (AIaaS), que permite a los usuarios acceder a capacidades avanzadas de IA sin necesidad de poseer una infraestructura local dedicada o de un conocimiento profundo en el área.

Plataformas de IA

AIaaS es el tipo de solución más completa proporcionada por plataformas automatizadas y semiautomatizadas *cloud*. La mayoría solucionan los problemas de infraestructura, como el preprocesamiento de datos, el entrenamiento y la evaluación de modelos. Este tipo de plataformas proporcionan módulos y herramientas predefinidas para automatizar el diseño, las pruebas y la implementación de los servicios de IA, así como una infraestructura *hardware*. Esto las convierte en el entorno ideal para los desarrolladores que no tienen suficientes recursos o experiencia.

Algunos proveedores han empezado incluso a ofrecer una solución más útil: *AI Platform as a Service* (AI PaaS). En lugar de ofrecer un servicio independiente que aproveche las capacidades de la IA, proporcionan plataformas de IA a gran escala. AI PaaS resuelve el problema intrínseco de la IA, que es que consume una gran cantidad de recursos, ofreciendo así un ahorro de dinero, tiempo y esfuerzo.

IA PaaS puede entenderse desde la perspectiva del modelo clásico *Platform as a Service* (PaaS). En PaaS, un proveedor de la nube proporciona un entorno para crear, desplegar y mantener aplicaciones. Del mismo modo, aunque no existe una arquitectura o modelo general para AI PaaS, se puede identificar una serie de elementos que son comunes a la mayoría de las plataformas de

servicios de IA y PaaS actuales: infraestructura, almacenamiento de datos, modelos ML preentrenados y API.

Existen numerosas plataformas *cloud* de IA que ofrecen entornos asequibles con un alto rendimiento. La computación *cloud* les proporciona una alta escalabilidad y disponibilidad de recursos, así como de la capacidad de gestionar y almacenar *datasets* grandes. Algunas plataformas destacadas son:

- **IBM Watson** [124] es una plataforma de IA orientada a las empresas que ofrece servicios basados en el procesamiento del lenguaje natural, traducción automática, el reconocimiento visual y ML. Watson ofrece servicios preconstruidos, así como una interfaz de programación de aplicaciones (del inglés API) para desarrollar nuevas soluciones de IA y herramientas para gestionar el ciclo de vida completo de una aplicación basada en IA.
- **AI4EU** [125] es una plataforma europea de IA que tiene como objetivo proporcionar un entorno colaborativo y un conjunto de herramientas avanzadas para el desarrollo, implementación y financiamiento de innovaciones basadas en la IA. La plataforma adopta un enfoque centrado en el ser humano y se fundamenta en cinco principios clave: *explainable AI*, *verifiable AI*, *collaborative AI*, *physical AI* y *secure AI*.
- **Google AI Platform** [126] es una plataforma integral dirigida a científicos de datos, desarrolladores de IA y ML. Esta plataforma cuenta con servicios para abordar el ciclo de vida de los modelos de aprendizaje automático. Esta ofrece bloques de construcción para desarrollar y desplegar modelos complejos de aprendizaje automático, incluyendo, entre otros, un servicio de entrenamiento, un servicio de etiquetado de datos y otro de despliegue de modelos.
- **Microsoft Azure AI** [127] ofrece un conjunto de servicios de aprendizaje automático basados en Python que pueden utilizarse tanto en *cloud* como en *edge*. Incluye un conjunto de servicios de IA preentrenados, a los que se puede acceder a través de una API para crear aplicaciones cognitivas. Algunos de estos servicios son el procesamiento del lenguaje y del habla, la visión por ordenador, la búsqueda cognitiva o la gestión de decisiones. También ofrece un conjunto de herramientas de desarrollo para la creación de modelos de ML personalizados.
- **Amazon AWS AI** [128] es un conjunto de servicios públicos de computación *cloud* que, en conjunto, forman una plataforma de computación *cloud*, ofrecida a través de Internet por Amazon. Proporciona una infraestructura técnica abstracta, bloques de construcción y herramientas de computación de manera distribuida. Además, también ofrece un entorno

de desarrollo integrado para construir, entrenar y desplegar modelos de ML personalizados. Ejemplos de algunos servicios son el procesamiento de voz y texto, la visión por ordenador y la búsqueda cognitiva, entre otros. En cuanto al acceso a los servicios, en lugar de exponer directamente los servicios a los usuarios finales, el acceso se proporciona mediante APIs. El acceso a los servicios web se realiza a través de HTTP, utilizando los protocolos REST y SOAP.

- **Oracle AI** [129] es una plataforma dirigida a empresas, científicos de datos y desarrolladores. Ofrece modelos de IA preentrenados y herramientas de desarrollo que cubren los diferentes pasos del ciclo de vida del desarrollo de la IA, desde la gestión de datos hasta el desarrollo de aplicaciones y la ciencia de datos. También ofrece aplicaciones de IA *cloud* preconstruidas.

Servicios de IA

Los servicios de IA proporcionan una amplia gama de modelos preentrenados, diseñados para abordar desafíos específicos. A través de una API, tanto desarrolladores como usuarios pueden aprovechar sus capacidades y utilizarlas para construir o mejorar aplicaciones.

Los servicios de AI cuentan con numerosos beneficios. En primer lugar, simplifican la integración de capacidades cognitivas en los sistemas. Desarrolladores sin experiencia pueden utilizar e integrar modelos preentrenados. Además, el uso de modelos preentrenados ahorra tiempo y recursos durante la fase de entrenamiento. También destacan por su escalabilidad y alta disponibilidad. El acceso a los servicios *cloud* a través de una API asegura que estén siempre disponibles y ofrezcan un rendimiento óptimo, incluso en situaciones de alta exigencia y tráfico intenso. Una ventaja adicional de utilizar una API para acceder a estos servicios es la facilidad con la que se pueden integrar en otras aplicaciones o sistemas. Por último, los proveedores de servicios de IA actualizan y mejoran constantemente sus modelos.

Existe un abanico amplio de servicios de IA. A continuación, se citan algunos de los más destacados:

- **Servicios de visión artificial:** Estos servicios permiten el procesamiento de imágenes y vídeos. Incluye funcionalidades de clasificación de imágenes, identificación de contenidos y detección y seguimiento de objetos. Algunos ejemplos son Google Vision AI [130] y Microsoft Computer Vision API [131].
- **Chatbots:** Los *chatbots* responden como una entidad inteligente cuando se conversa con ellos a través de texto o voz. Gracias a NLP son capaces de

entender múltiples idiomas. Algunos ejemplos son ChatGPT[14] o Google Bard [16].

- **Análisis de sentimiento:** Servicios como IBM Watson Natural Language Understanding [132] y Lexalytics [133] utilizan IA para analizar el sentimiento expresado en textos, como comentarios o reseñas de productos.
- **Traducción de texto:** Los servicios de traducción de textos más avanzados, como Google Translate API [134], están basados NLP. Permiten utilizar múltiples fuentes de datos como texto plano, páginas webs o documentos (PDF, Word, Excel, entre otros). Además ofrecen sugerencias de corrección y mejoras en la traducción.
- **Gestión de decisiones mediante análisis predictivo:** Algunos servicios como IBM Watson for Oncology [135] y Aidoc [136], se utilizan en el diagnóstico médico para ayudar a los profesionales de la salud a interpretar imágenes médicas y datos clínicos.

CAPÍTULO 2. ESTADO DEL ARTE

Capítulo 3

Definición de la arquitectura de referencia multidominio para AIoT

3.1. Introducción

La creciente cantidad de datos en los sistemas requiere una analítica avanzada que genere valor añadido, proporcionando y flexibilizando la interpretación de los datos para una detección más rápida, una mejor previsión y toma de decisiones.

La utilización de IA en los sistemas IoT habilita la toma de decisiones dinámicas en tiempo real, la presentación de hipótesis imparciales, argumentos razonados y recomendaciones al mismo tiempo que el sistema aprende de su entorno. Los elementos IoT actúan como receptores del entorno físico y la información es procesada por la parte cognitiva del sistema. Los resultados del análisis se devuelven a los sensores y a las interfaces de usuario. De esta forma se forma un bucle cerrado completo que abarca la toma de decisiones, el aprendizaje y la acción.

Una arquitectura de referencia es un diseño que se utiliza como una plantilla para guiar la definición de otras arquitecturas más específicas. Carece de detalles tecnológicos, pero incluye directrices de alto nivel a seguir, los bloques generales a implementar, así como sus relaciones, además de recomendaciones tecnológicas.

El objetivo de este Capítulo es la presentación de una arquitectura de referencia AIoT, personalizable y multidominio que permita integrar servicios

CAPÍTULO 3. DEFINICIÓN DE LA ARQUITECTURA DE REFERENCIA MULTIDOMINIO PARA AIOT

cognitivos en sistemas IoT utilizando estándares comunes y generalmente aceptados. La arquitectura propuesta tiene como objetivo superar la brecha entre las complejas arquitecturas IoT y la IA, aunando ambas tecnologías en una única arquitectura a través de la integración de tecnologías semánticas y cognitivas.

En contraste con otras arquitecturas AIoT, la presentada en este Capítulo se basa en estándares reconocidos, como UIT-T Y.2066, IIRA o RAMI 4.0. para aumentar su nivel de aceptación y confianza. Esta arquitectura proporciona una estructura y directrices para lograr una implementación efectiva y permitir una combinación eficiente de dispositivos IoT y tecnologías de IA. Por lo tanto, esta arquitectura contribuye significativamente a la estandarización y consolidación de un enfoque coherente y robusto para la integración de IoT e IA. Para ello, se fundamenta en una base semántica que posibilita la recopilación y preprocesado de datos de manera fluida. Esto, a su vez, habilita un análisis enriquecido y significativo de la información.

3.2. Análisis de requisitos

La siguiente Sección analiza y presenta los requisitos necesarios para la definición de una arquitectura de referencia multidominio para AIoT que facilite la implementación de servicios cognitivos sobre IoT. En la definición de estos requisitos, se ha tenido en cuenta que los sistemas IoT comparten una serie de características comunes. También se han identificado las necesidades y los requisitos específicos que surgen al integrar capacidades cognitivas en IoT.

Durante la identificación de los requisitos, se ha empleado como punto de partida la recomendación UIT-T Y.2066 [137], que proporciona los requisitos comunes de IoT.

R1. Alineación con arquitecturas previas

La arquitectura de referencia debe alinearse con arquitecturas previamente establecidas y consolidadas. La motivación de este requisito es evitar reinventar un nuevo modelo arquitectónico desde cero y para estar alineado y ser compatible con proyectos existentes.

R2. Acceso a múltiples plataformas de datos IoT

La arquitectura debe ser capaz de conectarse con distintas plataformas de datos IoT heterogéneos. Esto permite el acceso a una amplia gama de dispositivos y sensores, independientemente de su fabricante o especificaciones técnicas (sensores IoT, actuadores o *gateways*). Además, debe ser posible el acceso a datos de componentes y sistemas heredados.

R3. Conectividad y normalización

Existen gran cantidad de protocolos utilizados por los dispositivos y servicios IoT. La arquitectura debe ser capaz de utilizar los protocolos de comunicación IoT más comunes, asegurando una normalización y homogeneización del formato de los datos recibidos.

R4. Interoperabilidad

La arquitectura debe asegurar la interoperabilidad efectiva entre diferentes dominios, permitiendo que los sistemas y servicios trabajen de manera colaborativa y coherente. Para asegurar interoperabilidad entre las distintas fuentes de datos, se debe permitir un intercambio de datos con un significado compartido y sin ambigüedades. El uso de un modelo de datos estándar permite que el significado de los datos puede ser entendido por todos los componentes involucrados. Este requisito es indispensable a la hora de combinar datos de diversas fuentes y dar lugar a *datasets* coherentes y armonizados.

R5. Almacenamiento de datos históricos

La arquitectura debe permitir la consulta de los datos históricos de forma ágil y eficiente. Debe ser capaz de gestionar adecuadamente la gran variedad de formatos y estructuras que pueden presentar estos datos. Además, es preciso que cuente con mecanismos de búsqueda y recuperación de datos que garanticen tiempos de respuesta óptimos, incluso para grandes volúmenes. Este punto es esencial para el proceso de entrenamiento continuo de los modelos de IA.

R6. Procesado de los datos en tiempo real

La arquitectura debe ser capaz de obtener y procesar datos en tiempo real. Los retrasos en la salida del sistema deben ser lo suficientemente cortos para que el usuario no perciba ninguna latencia. Miller [138] fue el primer investigador que abordó sistemáticamente el tiempo de respuesta de los sistemas en relación con la satisfacción del usuario. Según su estudio, si un sistema informático responde menos de 200 ms después de las acciones del usuario, este parecerá fluido y receptivo.

R7. Incorporación de capacidades cognitivas

La arquitectura debe contar con la capacidad de generar servicios cognitivos para realizar un análisis avanzado de los datos recopilados. Asimismo, se plantea una evolución progresiva para generar servicios cognitivos, partiendo de una

CAPÍTULO 3. DEFINICIÓN DE LA ARQUITECTURA DE REFERENCIA MULTIDOMINIO PARA AIOT

inteligencia débil en las etapas iniciales y avanzando hacia una inteligencia más avanzada en etapas posteriores.

R8. Soporte en la integración y distribución de servicios

La arquitectura debe ofrecer soporte en la integración y distribución de sus servicios. Para facilitar su integración en interfaces de usuario o en otros sistemas, los servicios deben ser accesibles a través de API. A su vez, la arquitectura debe ser capaz de escalar y gestionar eficientemente el tráfico generado por las peticiones a las API, especialmente en escenarios de alto rendimiento y alta concurrencia.

R9. Extensibilidad

La arquitectura debe ser capaz de evolucionar y adaptarse a nuevos requisitos y necesidades sin afectar a las funcionalidades existentes. Este requisito es indispensable, ya que la metodología utilizada durante el diseño y validación de la arquitectura requiere que esta se actualice con cada instanciación. Para lograr esta flexibilidad, la arquitectura debe ser diseñada de forma modular. Se trata de un requisito esencial para mantener su relevancia y utilidad en un entorno tecnológico en constante evolución.

R10. Escalabilidad

La arquitectura debe ser escalable. El sistema debe soportar un crecimiento sostenido de la carga de trabajo y la demanda de recursos sin degradar su rendimiento.

R11. Independencia del dominio

La arquitectura debe ser diseñada para ser general, flexible e independiente del dominio de aplicación. En lugar de adaptarse a dominios particulares, debe favorecer la reutilización de componentes y servicios comunes. Su versatilidad y adaptabilidad debe facilitar que esta sea instanciada en diferentes dominios sin requerir modificaciones significativas.

R12. Independencia tecnológica

La arquitectura no debe estar vinculada a proveedores o soluciones concretas. Debe permitir la incorporación de diversas tecnologías y herramientas de acuerdo con las necesidades y preferencias de los casos de uso. Este desacople de tecnologías específicas reduce el riesgo de que en un futuro quede atrapada en tecnologías o enfoques obsoletos.

3.3. Arquitectura

La arquitectura de referencia se define desde una perspectiva que abarca varias de las vistas definidas por IIRA y RAMI 4.0. En concreto, se define a partir de la vista funcional y la vista de implementación

3.3.1. Vista funcional

Se propone una arquitectura por niveles donde cada nivel gestiona una parte diferente del sistema y proporciona un conjunto de servicios al resto. La Figura 3.1 muestra la vista funcional.

Nivel de adquisición de datos

El nivel de adquisición de datos es el nivel de partida. Se encarga de la obtención de datos de distintas fuentes. Al igual que en el nivel de percepción de las arquitecturas clásicas, permite percibir las propiedades físicas de los objetos (como la temperatura, la ubicación, etc.) mediante diversos sensores y transmitir esta información utilizando el protocolo de comunicación más apropiado. También permite actuar sobre el mundo real modificando el comportamiento de los objetos conectados. Adicionalmente, gestiona la representación de los datos no estructurados recibidos de los dispositivos físicos, lo que incluye la generación de información de contexto (metadatos). Esta información contextual es esencial para comprender mejor la procedencia, calidad y relevancia de los datos.

Además de interactuar con dispositivos físicos, este nivel también se ocupa de acceder a datos de servicios de terceros. Por un lado, permite establecer conexiones con servicios web a través de sus API para obtener datos en tiempo real. Además, facilita el acceso a datos históricos de fuentes externas. Esto permite que el sistema pueda analizar y comprender patrones, tendencias y comportamientos pasados, fundamentales para la realización de análisis predictivos y la toma de decisiones inteligentes.

Este mecanismo de adquisición de datos combinado, que integra información tanto de sensores IoT como de servicios externos, permite crear una base sólida de conocimiento acumulado. De esta forma se enriquece la capacidad de aprendizaje del sistema y su habilidad para adaptarse a situaciones cambiantes y tomar decisiones informadas en tiempo real.

Los datos obtenidos por el nivel de adquisición de datos son consumidos por el nivel de *middleware*, garantizando una integración perfecta con el sistema y permitiendo que los servicios y aplicaciones superiores sean más precisos.

CAPÍTULO 3. DEFINICIÓN DE LA ARQUITECTURA DE REFERENCIA MULTIDOMINIO PARA AIOT

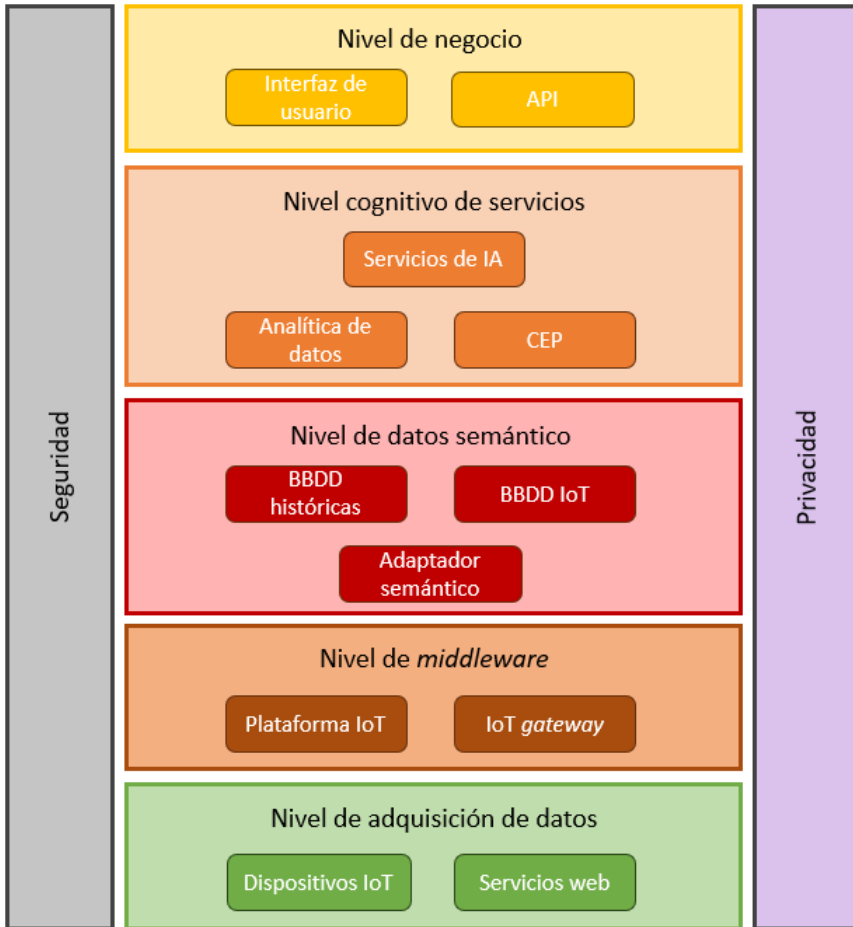


Figura 3.1: Vista funcional.

Nivel de *middleware*

En las arquitecturas IoT clásicas, el nivel de *middleware* es el encargado de facilitar la comunicación, interacción y coordinación entre los dispositivos IoT y las aplicaciones o servicios que utilizan los datos generados por estos dispositivos. Proporciona medios para acceder a la información, incluyendo una interfaz y un modelo de datos común.

Este nivel es compatible con los protocolos de red IoT utilizados para el intercambio y almacenamiento de datos e información. Esto es fundamental para facilitar el desarrollo de servicios en un entorno con muchos componentes de redes y sistemas diferentes.

Uno de los puntos diferenciadores de esta arquitectura respecto a las arquitecturas clásicas es que la integración de fuentes de datos externas se hace en el nivel más bajo. Es por esto que es necesario que el *middleware* sea capaz comunicarse con servicios externos para recuperar sus datos.

De forma general, el nivel de *middleware* despliega una serie de funcionalidades fundamentales para el procesamiento y la distribución eficiente de datos:

- Interfaz común para el suministro y la recuperación de datos: Proporciona una interfaz uniforme que facilita tanto el suministro de datos desde los dispositivos y fuentes de datos conectadas como la recuperación de datos por parte de los servicios y aplicaciones superiores.
- Enrutamiento de la información a los consumidores de datos: El enrutamiento se encarga de direccionar los datos recibidos desde los sensores y servicios de terceros hacia los suscriptores correspondientes. Esto se logra mediante la asignación adecuada de reglas y filtros, lo que permite identificar los intereses y requisitos de cada suscriptor y asegura que solo reciban los datos pertinentes a sus necesidades específicas. Este enrutamiento inteligente evita la sobrecarga de los servicios y reduce la latencia.
- Mecanismos *push* y *pull* para la recuperación de datos: Se ofrecen los métodos de recuperación de datos *push* y *pull* del patrón publicación/-suscripción [139]. El método *push* es especialmente útil en aplicaciones donde la información debe ser procesada y reaccionar rápidamente a eventos en tiempo real.
- Gestión de la información sobre la disponibilidad del contexto: Se monitorea y gestiona la disponibilidad y confiabilidad de los datos contextuales, lo que permite informar a los servicios superiores sobre la calidad y actualidad de la información recibida.
- Gestión de la información de contexto: El *middleware* se encarga de almacenar y administrar la información contextual relacionada con los datos

CAPÍTULO 3. DEFINICIÓN DE LA ARQUITECTURA DE REFERENCIA MULTIDOMINIO PARA AIOT

adquiridos, como metadatos, ubicación geográfica, tiempo, estado de los dispositivos, entre otros. Esta información es esencial para comprender y analizar adecuadamente los datos recibidos.

En función de los requerimientos del sistema y su complejidad, el *middleware* puede estar implementado mediante una plataforma IoT o una *gateway*.

Nivel de datos semántico

El nivel de datos semántico proporciona el almacenamiento de datos y los servicios de recogida, suministro y transformación de estos, incluida la agregación de datos.

La representación semántica de la información también forma parte de los servicios prestados por este nivel. Esta permite soportar interoperabilidad entre múltiples objetos y sistemas heterogéneos. Esto proporciona una estructura común de conocimiento que es clave para que los mecanismos cognitivos de la misma sean capaces de identificar los datos útiles, extraer conocimiento de forma inequívoca y fiable de los sistemas, identificar patrones existentes, aprender de ellos y, en última instancia, ofrecer servicios cognitivos.

Los datos deben ser almacenados utilizando modelos de datos semánticos. Estos modelos describen los datos y metadatos de los sensores, además de incluir una descripción del dominio. Almacenar los datos utilizando un modelo semántico unificado garantiza que el sistema pueda extraer información significativa de los datos no estructurados, dándole sentido a estos.

Las principales funcionalidades son:

- Almacenamiento de datos: Permite capturar, gestionar y procesar la información obtenida por el nivel de adquisición de datos. El almacenamiento debe ser altamente escalable y tener una gran capacidad para acomodar la creciente cantidad de información generada por los dispositivos conectados. Además, debe permitir almacenar tanto datos estructurados como no estructurados.
- Acceso a datos históricos: Permite acceder, guardar y gestionar datos o históricos generados por el nivel de adquisición de datos. Junto con los datos históricos, es importante conservar metadatos relevantes, como la fecha y hora de registro, la fuente de datos.
- Agregación de datos: Permite recopilar y combinar datos de múltiples fuentes o fuentes individuales para obtener una visión más completa y significativa de la información.

- Conversión de los datos utilizando un modelo de datos común: Permite transformar y estandarizar los datos provenientes de diferentes fuentes en un formato común y coherente. Esto es esencial para asegurar que los datos sean interoperables y puedan ser procesados y comprendidos de manera uniforme por todos los componentes del sistema.
- Representación semántica de los datos: Se debe permitir una representación semántica de los datos. La incorporación de modelado ontológico permite definir jerarquías, relaciones y propiedades entre los conceptos y entidades relevantes del sistema. Esto ayuda a los servicios cognitivos a interpretar con mayor precisión los datos recogidos.

Nivel cognitivo de servicios

El nivel cognitivo de servicios habilita la creación de soluciones cognitivas en la arquitectura. Este nivel proporciona un conjunto de servicios que permiten una extracción y análisis avanzado de los datos. Entre sus capacidades clave se incluyen: *Complex Event Processing* (CEP), *data analytics* y servicios de IA.

Los servicios son utilizados por el nivel de negocio para extraer información de los datos o hacer predicciones. La integración de servicios cognitivos en sistemas heredados permite aumentar su valor. Adicionalmente, los servicios pueden integrarse entre sí dentro del nivel de servicios para dar lugar a servicios más complejos.

Con el uso de CEP, los flujos de datos se analizan en tiempo real y se comparan con reglas predefinidas. Sin embargo, debido a su limitada memoria, no explotan los datos históricos. Por lo tanto, sus soluciones son únicamente reactivas. Por otro lado, *data analytics* y los servicios de IA sí que cuentan con un carácter predictivo.

Las principales funcionalidades son:

- CEP: Permite procesar, analizar y correlacionar flujos de sucesos procedentes de distintas fuentes de datos para deducir eventos más complejos a medida que se producen.
- *Data analytics*: La arquitectura habilita la incorporación de mecanismos de análisis de *big data* para soportar el análisis de datos no estructurados o semiestructurados. Permite caracterizar los eventos pasados y anticipar las probabilidades y tendencias futuras. También permite identificar patrones anómalos o tendencias, extraer información significativa a partir de texto no estructurado o extraer contenido semántico en datos multimedia.
- Servicios de IA: Los servicios cognitivos basados en IA permiten realizar un análisis avanzado de datos generados. A medida que reciben nueva

CAPÍTULO 3. DEFINICIÓN DE LA ARQUITECTURA DE REFERENCIA MULTIDOMINIO PARA AIOT

información, ganan nuevo conocimiento y valor, generando predicciones y recomendaciones más precisas. Además, estos servicios pueden automatizar procesos complejos, realizar tareas repetitivas con precisión y velocidad, y mejorar la toma de decisiones mediante la generación de conocimientos a partir de datos.

- **Procesamiento de datos en tiempo real:** Permite procesar la información en un periodo de tiempo muy corto, a menudo en cuestión de milisegundos. Es una funcionalidad especialmente importante en entornos donde la interacción con humanos es crucial, como en aplicaciones médicas, entornos de producción o sistemas de seguridad.
- **Procesamiento de datos por lotes:** Facilita el procesado de grandes volúmenes de datos cuando los tiempos de procesamiento no son una limitación.
- **Servicios *cloud*:** Ofrecen una amplia capacidad de almacenamiento y procesamiento, al tiempo que ofrecen escalabilidad y flexibilidad. Son idóneos para la inferencia de modelos de IA basados en redes neuronales.
- **Interacción entre servicios:** Los servicios deben ser capaces de interactuar entre sí para ofrecer funcionalidades avanzadas. Para facilitar su interacción, los servicios deben estar débilmente acoplados y ser altamente interoperables.

Nivel de negocio

Este nivel orquesta los servicios ofrecidos por el nivel cognitivo de servicios en función del dominio de aplicación y los requisitos del sistema. Actúa como enlace entre los diferentes sistemas y permite las operaciones de extremo a extremo. Además, proporciona API para aplicaciones específicas del dominio e interfaces de usuario para la visualización de datos.

Sus principales funciones son:

- **Orquestación de servicios:** Es esencial para lograr ecosistemas interconectados y eficientes. Permite la convergencia de sistemas, aplicaciones y sensores independientes en una red común de gestión de servicios. Mediante la orquestación, se logra una integración fluida y coherente de los diversos servicios, permitiendo la comunicación y coordinación entre ellos de manera eficiente.
- **Visualización de datos:** Permite analizar, explorar y visualizar los datos recopilados por el sistema. Esta representación gráfica facilita la identi-

ficación de patrones, tendencias y relaciones ocultas. Esta funcionalidad debe ser fácil de utilizar, tanto para usuarios técnicos como no técnicos.

- **Interfaces de usuario:** El sistema debe proporcionar un conjunto de interfaces visuales e interactivas que permitan a los usuarios interactuar con él. Las interfaces deben permitir acceder a las funcionalidades específicas de la implementación concreta.
- **API:** La definición y desarrollo de API es esencial para garantizar la flexibilidad, adaptabilidad y escalabilidad de la arquitectura. Estas facilitan la integración de nuevas funcionalidades, módulos y servicios en el sistema, sin tener que modificar el núcleo de la aplicación. Además, permiten la comunicación y la interoperabilidad con otros sistemas y aplicaciones.

Niveles de seguridad y privacidad

Los niveles de seguridad y privacidad actúan como pilares que sustentan la confianza y la integridad de la arquitectura. Abarcan de manera integral todos los niveles de la arquitectura y no deben ser tratados de forma aislada, ya que la falta de protección en un nivel puede debilitar la integridad de toda la arquitectura.

La seguridad y la privacidad son elementos críticos de la implantación del IoT. Muchos dispositivos IoT requieren la recopilación, el análisis y la transmisión de datos potencialmente sensibles. Es esencial que estos datos estén adecuadamente protegidos en todo momento. El nuevo Reglamento General de Protección de Datos (RGPD) del Parlamento Europeo y la Comisión Europea, otorga pleno control al individuo sobre el procesamiento de datos personales, y se necesita un consentimiento válido para el uso de datos, incluso si los datos están seudonimizados. El consentimiento del usuario, la transparencia, la responsabilidad, la rendición de cuentas y el derecho al olvido figuran entre los requisitos del reglamento. La privacidad también implica confidencialidad.

El nivel de seguridad tiene como objetivo proporcionar un entorno fiable que permita confiar en la arquitectura global. Esta confianza se refleja en la capacidad de garantizar tres cualidades:

- **Confidencialidad:** Solo el destinatario legítimo (o poseedor) de un bloque de datos o de un mensaje puede tener una visión inteligible del mismo.
- **Autenticación:** Al enviar un bloque de datos o un mensaje o al conectarse a un sistema, se conoce la identidad del remitente o la identidad del usuario.
- **Integridad:** Garantiza que un bloque de datos o un mensaje enviado no ha sido modificado, accidental o intencionadamente.

CAPÍTULO 3. DEFINICIÓN DE LA ARQUITECTURA DE REFERENCIA MULTIDOMINIO PARA AIOT

En lo que respecta a la privacidad del usuario, se han de garantizar los siguientes aspectos: i) minimizar el tratamiento de datos personales; ii) pseudonimizar los datos personales lo antes posible; iii) transparencia en relación con las funciones y el tratamiento de datos personales; iv) permitir que el interesado controle el tratamiento de los datos; v) permitir al controlador crear y mejorar los dispositivos de seguridad.

Flujos de datos

La Figura 3.2 muestra los flujos de datos de la vista funcional. Todos los intercambios de datos siguen el paradigma productor-consumidor [140]. Este es útil en fases posteriores, cuando se utiliza la arquitectura de referencia para definir arquitecturas concretas. Durante estas fases, la relación entre los componentes funcionales de la arquitectura ayuda a definir interfaces concretas entre los componentes funcionales a implementar.

La arquitectura proporciona dos mecanismos para introducir información al sistema y enviarla al *middleware*: i) adquisición de datos a través de sensores IoT; ii) adquisición de datos a través de servicios web externos. Además, en función de los requerimientos del sistema, los datos pueden ser enviados directamente a una plataforma IoT, a una plataforma IoT pasando previamente por una *gateway* o simplemente a una *gateway*.

Antes de ser almacenados, los datos son transformados a un formato de datos común. Los datos son almacenados en dos tipos de bases de datos, las propias de las plataformas IoT y las históricas. Las bases de datos de las plataformas IoT son utilizadas por CEP para el análisis de datos en tiempo real, mientras que las bases de datos históricas se utilizan para la analítica de datos a lo largo del tiempo. Los servicios de IA utilizan ambas bases de datos. Los datos históricos son indispensables para su entrenamiento, mientras que la obtención de datos en tiempo real se utiliza para las inferencias y la toma de decisiones en tiempo real.

Por último, los servicios se ofrecen a los usuarios a través de interfaces de servicios y/o de API.

La privacidad y la seguridad están presentes en todos los niveles de la arquitectura. En todos ellos existen riesgos de interceptación y manipulación de la información. La privacidad es crucial para garantizar que los datos personales se manejan de manera adecuada y legal, mientras que la seguridad protege contra ataques que pueden comprometer la integridad y disponibilidad de los datos capturados.

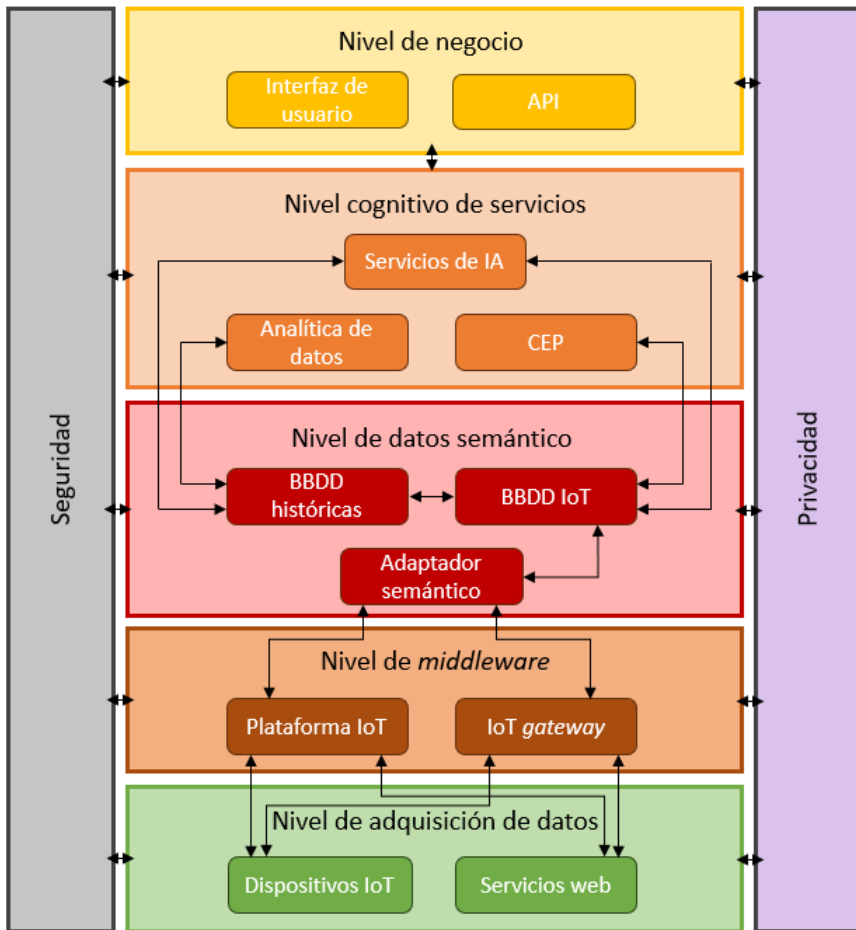


Figura 3.2: Vista funcional. Flujos de datos.

3.3.2. Vista de implementación

La vista de implementación de la arquitectura AIoT se ilustra en la Figura 3.3. Está compuesta por los siguientes niveles:

- **Dispositivos y servicios:** Dispositivos IoT y servicios web externos de los cuales la arquitectura extrae información.
- **Red:** Se refiere a la red desplegada para dar soporte tanto a la intercomunicación de los recursos físicos como a la conexión de dichos recursos con *edge computing* y *cloud computing*.
- **Cloud/Edge:** *Edge computing* permite reducir la latencia y los problemas de conectividad al procesar datos cerca de donde se crean. Sin embargo, la capacidad de procesamiento *edge* es significativamente menor que el procesamiento *cloud*. Por otro lado, *cloud computing* ofrece soluciones de infraestructura, plataforma y *software* como servicio (IaaS, PaaS, SaaS) mediante la virtualización de las capacidades informáticas y de almacenamiento que pueden ampliarse o reducirse en función de la demanda. Este nivel abarca servicios *cloud* privados o públicos en función de las preocupaciones estratégicas o de seguridad.
- **Terminales de usuario:** Incluye todos los terminales de usuario final (p. ej., PC, ordenadores portátiles, *tablets*, etc.) que pueden acceder a los resultados del procesamiento y a los datos de almacenamiento alojados en el *cloud*, y mostrarlos a un usuario final.

El nivel inferior de la vista de implementación está formado por todos aquellos sensores, actuadores y servicios web necesarios para implementar los casos de uso definidos por el sistema objetivo.

El nivel inmediatamente superior es el nivel de red. Este es responsable de la interconexión de dispositivos y sistemas dentro de la arquitectura.

Seguidamente se encuentra el nivel *cloud/edge*. Los servicios que requieren procesado en tiempo real se benefician de las ventajas que ofrece *edge computing*. *Cloud computing* se utiliza para implementar servicios que no presentan requerimientos críticos de latencia o que quieran aprovechar la gran capacidad de almacenamiento y potencia de cómputo *cloud*. Este enfoque híbrido permite una distribución más equilibrada de la carga de trabajo en la red. Así, se optimiza la utilización de recursos y mejora la capacidad de respuesta en tiempo real. La arquitectura permite procesar y filtrar los datos localmente *edge computing* antes de enviar solo la información relevante a la nube. De esta forma se reduce el ancho de banda requerido y se mejora la eficiencia general del sistema. El procesado de los datos puede realizarse en un módulo de análisis basado en *big data*, un módulo de IA o un módulo dedicado a procesar eventos (CEP).

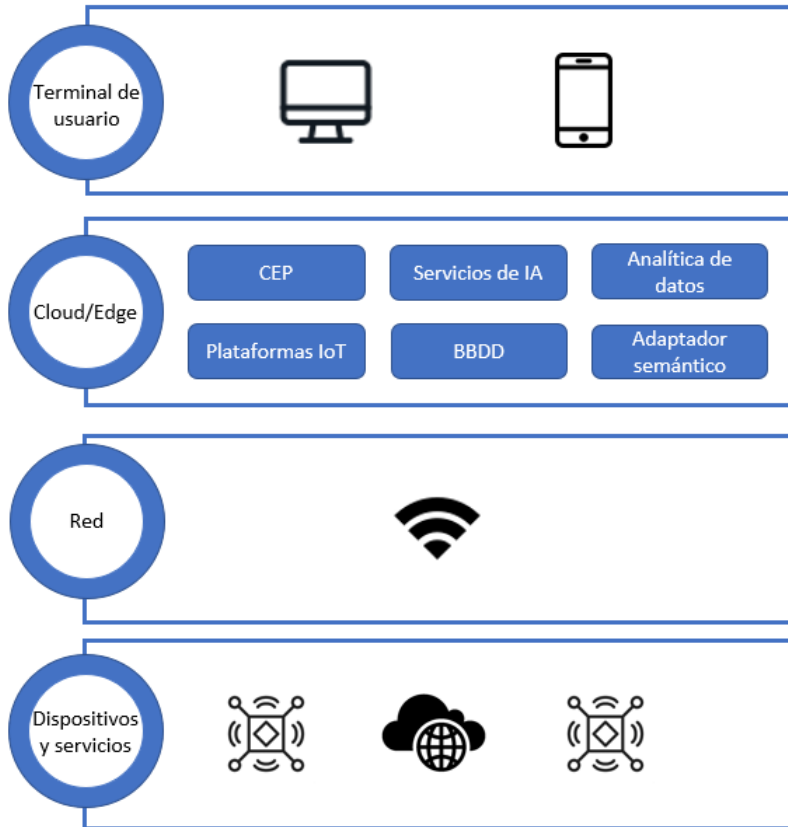


Figura 3.3: Vista de implementación.

CAPÍTULO 3. DEFINICIÓN DE LA ARQUITECTURA DE REFERENCIA MULTIDOMINIO PARA AIOT

Por último, se encuentra el nivel de terminales de usuario formado por todos aquellos dispositivos físicos junto con sus interfaces que permiten a los usuarios interactuar con el sistema.

3.4. Conclusiones

En este Capítulo se ha presentado una arquitectura de referencia multidominio AIoT que habilita la creación de servicios cognitivos en sistemas IoT. Se trata de una arquitectura por niveles, flexible y modular. Ha sido diseñada de manera genérica para servir como una plantilla adaptable que puede ser empleada para desarrollar arquitecturas específicas en distintos dominios. Gracias a la arquitectura de referencia, las arquitecturas derivadas pueden incorporar con facilidad los requisitos y la lógica de negocio particulares del sistema objetivo.

En el momento en que se inicia este trabajo, la convergencia entre IoT e IA se encontraba en sus primeras etapas y, según el conocimiento del autor, no se habían propuesto todavía arquitecturas AIoT con un enfoque integral. En su lugar, existían arquitecturas donde coexistían ambas tecnologías de manera independiente, adaptándose la una a la otra. Durante el transcurso de la tesis, la comunidad científica ha propuesto diversas arquitecturas AIoT con un enfoque integral, pero ninguna ha sido reconocida como estándar o está ampliamente aceptada.

En contraste con otras arquitecturas AIoT, la arquitectura presentada se basa en estándares reconocidos, como UIT-T Y.2066, IIRA o RAMI 4.0, con el objetivo de aumentar su nivel de aceptación y confianza. Por lo tanto, esta arquitectura contribuye en la estandarización y la consolidación de un enfoque más coherente y robusto para la integración de IoT e IA. Su definición formal se ha realizado a través de diversas vistas: vista funcional y vista de implementación.

Desde el punto de vista funcional, la combinación del entorno de IoT con servicios cognitivos de alto nivel, tanto a nivel arquitectónico como lógico, permite el desarrollo e integración de habilitadores y servicios cognitivos para apoyar esta interpretación avanzada de datos y su aplicación a diferentes escenarios. La arquitectura presenta módulos de razonamiento específicos que permiten su autoorganización con el fin de proporcionar capacidades de autorregulación continua en sus sistemas y servicios, mecanismos de predicción, fiabilidad para ofrecer oportunamente la calidad de servicio requerida durante todo el ciclo de vida del sistema y autoadaptación para mejorar el rendimiento del sistema durante los cambios de contexto. Además, permite la creación de servicios dedicados de apoyo a la toma de decisiones, que son capaces de evaluar el con-

texto actual y sugerir posibles decisiones a través servicios cognitivos que se comunican con los usuarios.

Desde el punto de vista de implementación, la arquitectura presenta las posibles topologías del sistema, ofreciendo tanto una topología basada en *cloud computing*, *edge computing* o mixta. La topología final dependerá de las características y requerimientos del sistema objetivo.

CAPÍTULO 3. DEFINICIÓN DE LA ARQUITECTURA DE REFERENCIA MULTIDOMINIO PARA AIOT

Capítulo 4

Arquitectura cognitiva para la creación de servicios comunes en entornos de datos IoT heterogéneos

4.1. Introducción

El objetivo del Capítulo es la instanciación y validación de la arquitectura de referencia para permitir la creación de servicios comunes en entornos de datos IoT heterogéneos.

Esta primera instanciación se centra en los niveles inferiores de la arquitectura de referencia y pone especial énfasis en la incorporación de mecanismos de interoperabilidad para maximizar el valor y la explotación de los datos recopilados por los sensores IoT. En concreto, la arquitectura permite la extracción y recopilación de información proveniente de diferentes plataformas IoT, habilitando la creación de un ecosistema integral donde los datos generados por una variedad de dispositivos IoT pueden ser recopilados, procesados y compartidos de manera eficiente. Además, la unificación de los datos posibilita su explotación conjunta, a través del nivel de servicios, independientemente de su fuente de origen.

Se presenta una primera transformación general de la arquitectura de referencia para habilitar la creación de servicios comunes en entornos de datos IoT heterogéneos. Esta arquitectura se instancia y valida en un caso de uso

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

perteneciente al dominio de la salud, dando lugar a AIoTES. La definición de AIoTES se realiza teniendo en cuenta tanto los requisitos generales de la arquitectura de referencia como los específicos del caso de uso al que responde. Por ese motivo, el caso de uso se presenta con anterioridad. Este se centra en la creación del *framework* AIoTES, que permite la creación de un ecosistema IoT para envejecimiento activo. A continuación se define AIoTES, se detalla su relación con la arquitectura de referencia y sus detalles de implementación. Seguidamente, se explica la metodología empleada para su verificación y validación. Posteriormente, se representan los resultados obtenidos en la fase de validación. Tras esto, se incorpora una Sección adicional que describe la arquitectura EBASI. Esta permite la detección de soledad de personas mayores y se diseña tanto para ser utilizada de forma independiente como para ser integrada dentro del ecosistema. Por último, se presentan las conclusiones del Capítulo.

4.2. Arquitectura

Como primer paso para la validación de la arquitectura de referencia, se define una primera transformación que permite la creación de servicios comunes en entornos de datos IoT heterogéneos. Esta primera transformación pone el énfasis en identificar desafíos y oportunidades en la creación de servicios comunes en entornos de datos heterogéneos. Además, permite validar por primera vez la viabilidad y efectividad de la arquitectura AIoT propuesta.

Los objetivos específicos de la arquitectura son:

- Ofrecer una solución de *big data* que permita almacenar, procesar y extraer conocimiento de los datos obtenidos por las fuentes de datos heterogéneas.
- Facilitar el desarrollo de aplicaciones sobre las fuentes de datos heterogéneas.
- Garantizar la interoperabilidad de los datos.
- Proporcionar seguridad, privacidad, protección de datos y confianza, siguiendo la normativa y políticas europeas.

La Figura 4.1 ilustra la arquitectura para la creación de servicios comunes en entornos de datos heterogéneos. Empezando por abajo, el nivel de dispositivos representa el conjunto de objetos inteligentes, como sensores y actuadores, responsables de la recogida de datos. En los niveles superiores, estos datos son analizados y utilizados para monitorizar el sistema global y facilitar la toma de decisiones. En algunos casos, estas decisiones pueden implicar decisiones de

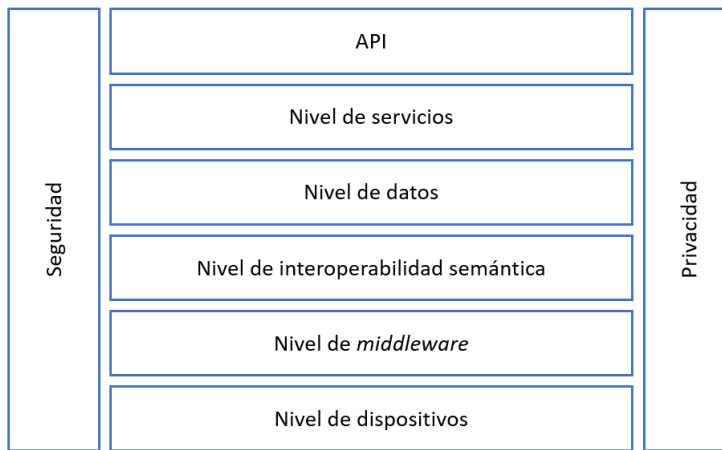


Figura 4.1: Arquitectura para la creación de servicios comunes en entornos de datos heterogéneos.

control, es decir, cambios de parámetros en los objetos conectados a través de los dispositivos actuadores.

El nivel de *middleware* contiene plataformas IoT que gestionan los datos de los dispositivos. Estas plataformas facilitan el intercambio técnico de datos, protocolos y configuraciones, lo que facilita el procesamiento de los datos IoT, ya sea en el entorno *edge* o *cloud*.

El nivel de interoperabilidad semántica desempeña un papel importante en la arquitectura. Permite y orquesta la interconexión de dispositivos IoT heterogéneos, plataformas IoT y servicios dentro de un sistema común de soluciones. Este nivel actúa como una capa de abstracción para las capas superiores. Permite que los servicios puedan explotar datos procedentes de múltiples fuentes de datos sin necesidad de lidiar con las particularidades de cada plataforma IoT. Además, facilita la creación de conjuntos de datos homogéneos provenientes de múltiples plataformas IoT. Esto permite extraer patrones y tendencias.

El nivel de datos almacena la información proveniente de las distintas plataformas IoT usando un formato de datos común. Esto garantiza la consistencia y la facilidad en el manejo de los datos en los niveles superiores.

El nivel de servicios se centra en ofrecer servicios y funcionalidades comunes. Los servicios desarrollados sobre el nivel de datos son compatibles con todas las plataformas IoT del ecosistema. Estos servicios pueden desplegarse, ampliarse y reproducirse en otros sistemas que sigan las especificaciones de la arquitectura.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

La API proporciona a los desarrolladores de aplicaciones, integradores y proveedores de servicios un marco común para crear aplicaciones y servicios. Facilita el acceso a los datos heterogéneos y la creación y mejora de servicios.

La seguridad y la protección de la privacidad son componentes cruciales de la arquitectura. Abarcan todos los niveles anteriores y garantizan tanto la protección de la información sensible de los usuarios como el cumplimiento de los requisitos éticos y legales de privacidad y confidencialidad. Además, el nivel de seguridad es responsable de la protección de la propiedad intelectual de los desarrolladores de aplicaciones.

4.3. Caso de uso: Creación de un ecosistema IoT para envejecimiento activo

En los últimos años, la esperanza de vida en la mayoría de los países del mundo ha aumentado de forma espectacular. Esta mejora se ha logrado gracias a los significativos avances en la ciencia médica y la tecnología de diagnóstico, así como a la creciente conciencia sobre la higiene personal y ambiental, la salud, la nutrición y la educación. Sin embargo, se espera que el aumento de la esperanza de vida junto con la caída de las tasas de natalidad de lugar a un gran envejecimiento de la población. De hecho, según la Organización Mundial de la Salud (OMS), la población anciana mayor de 65 años superará en número a los niños menores de 14 años en 2050 [141]. Además, alrededor del 15% de la población mundial sufre diversas discapacidades y entre 110 y 190 millones de adultos tienen dificultades funcionales significativas.

Un gran número de personas mayores requieren asistencia regular para su vida diaria y su atención médica. Los servicios formales de cuidado de ancianos ofrecidos por cuidadores o centros especializados son costosos y, por lo tanto, están todavía fuera del alcance de una gran parte de la población de edad avanzada que vive con un presupuesto limitado. En muchos casos, esta asistencia es proporcionada por cuidadores informales (familia, amigos o voluntarios [142]). Los cuidadores informales son proclives a padecer el síndrome del cuidador quemado [143], un estado de agotamiento físico, emocional y mental que, además de afectar personalmente al cuidador, puede llegar a comprometer la calidad de la atención ofrecida.

La monitorización remota de la salud de los pacientes mediante el uso de plataformas de hogar inteligente permite que las personas permanezcan en su casa en lugar de en costosas y limitadas residencias de ancianos u hospitales. Esto les proporciona independencia y descarga sensiblemente a sus cuidadores. Los hogares inteligentes están equipados con sensores y actuadores ambientales y fisiológicos discretos y no invasivos que facilitan la monitorización remota del

4.3 Caso de uso: Creación de un ecosistema IoT para envejecimiento activo

entorno del hogar (como temperatura, humedad y humo), así como importantes signos fisiológicos (como frecuencia cardíaca, temperatura corporal, la presión arterial y el nivel de oxígeno en la sangre) y las actividades de los ancianos. También puede comunicarse con las instalaciones de atención médica remotas, lo que permite que el personal de atención médica pueda realizar un seguimiento del estado fisiológico general de los ancianos y actuar, si es necesario, desde la distancia.

La comunidad de envejecimiento activo es amplia y heterogénea en términos de necesidades, demandas y entornos de vida. Los servicios de envejecimiento activo basados en IoT prometen ser un componente estratégico para apoyar la creación de un ecosistema capaz de responder y prevenir de forma dinámica los retos a los que se enfrentan los sistemas de asistencia sanitaria y social. El paradigma “siempre conectado” se está convirtiendo en una forma de vida, y esto podría dar lugar a una transformación positiva para encontrar nuevas formas de reorientar la prestación de asistencia y mantener a las personas mayores activas e independientes durante más tiempo.

El proyecto H2020 *ACTivating InnoVative IoT smart living environments for AGEing well* (ACTIVAGE) [144] tiene como misión principal la creación del primer ecosistema IoT europeo para envejecimiento activo. Busca promocionar el envejecimiento activo y saludable, apoyando y prolongando la vida independiente de las personas mayores en su entorno y sus casas. Del objetivo principal se deriva el siguiente subobjetivo, que es abordado utilizando la arquitectura de referencia propuesta en el presente trabajo:

- Diseñar *ACTIVAGE IoT Ecosystem Suite* (AIoTES), un conjunto de técnicas, herramientas y metodologías para interoperabilidad en diferentes niveles entre plataformas IoT heterogéneas existentes y un *framework* abierto para proporcionar servicios para envejecimiento activo, abordando la fiabilidad, la privacidad, la protección de datos y la seguridad.

El ecosistema está formado por múltiples pilotos, donde cada cual implementa diversos casos de uso. Los pilotos son: Comunidad de Galicia (España), ciudad de Valencia (España), Comunidad de Madrid (España), región de Emilia Romagna (Italia), múltiples ciudades de Grecia (regiones de Trikala, Atenas y Salónica), región de Isere (Francia), la residencia de ancianos WoQuaZ de la ciudad de Weiterstadt (Alemania), la ciudad de Leeds (Reino Unido) y varios municipios de Finlandia. Posteriormente el ecosistema se amplía con tres pilotos adicionales: Barcelona (España), Sofía (Bulgaria) y Lisboa (Portugal).

A continuación, se describen brevemente sus casos de uso.

- Seguimiento de la actividad diaria: Consiste en la monitorización de las actividades de la vida diaria de los mayores. Para ello se utilizan diferentes

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

sensores para analizar el entorno y las rutinas del anciano (presencia de otras personas en la casa, patrones de comportamiento, etc.).

- **Atención integrada:** Garantiza una estrecha colaboración entre los diferentes actores, como los servicios de salud y de atención domiciliaria, entre otros.
- **Control de los parámetros de salud:** Seguimiento de parámetros de salud (por ejemplo, la presión arterial, la frecuencia cardíaca y el peso). Cuando se producen anomalías en los valores normales de los ancianos, se alerta a los cuidadores.
- **Pulsador de alarma:** Permite a las personas asistidas comunicar su estado en caso de emergencia.
- **Promoción de ejercicio:** Este caso de uso se centra en la mejora de la condición física de los ancianos. También en la prevención de situaciones como las caídas o el deterioro de la salud causado por enfermedades asociadas o derivadas de la falta de ejercicio y el sedentarismo.
- **Estimulación cognitiva:** Promociona el conocimiento y la comprensión del entorno mediante la estimulación a través de diversos juegos para fomentar la actividad mental.
- **Prevención de aislamiento social:** Fomenta la participación social, monitorizando la actividad social de las personas asistidas y sus comunicaciones con diferentes personas. Esto se hace aprovechando un sencillo sistema IoT que permite la comunicación entre las personas asistidas y los cuidadores. Además, el sistema también se centra en la propuesta de diferentes actividades de ocio para fomentar las actividades al aire libre.
- **Tranquilidad, comodidad y seguridad en el hogar:** La incorporación de sensores de humo o dióxido de carbono permiten detectar la presencia de humo o gases peligrosos y alertar a los cuidadores.
- **Apoyo al desplazamiento y a la movilidad:** Monitorización sobre los patrones de movilidad de los ancianos, prestando atención no solo al transporte en sí (rutas, líneas, medios), sino también a su seguridad y comodidad, analizando sus patrones de movilidad habituales y detectando las desviaciones de su comportamiento.
- **Notificación de situaciones anómalas:** Este caso de uso se centra en analizar las pautas de comportamiento rutinarias de las personas asistidas (a través de sensores) y alertar a los cuidadores en caso de que se produzca una situación inusual.

- Soporte para cuidadores: Facilita la comunicación entre los cuidadores y los ancianos. Los cuidadores reciben resúmenes y estadísticas del comportamiento de los ancianos.

4.4. Implementación

A partir de la arquitectura definida para la creación de servicios comunes en entornos de datos IoT heterogéneos, se define e implementa AIO TES. Este *framework* se diseña con el objetivo de permitir y promover la creación, el intercambio y la adopción de servicios y aplicaciones multiplataforma para envejecimiento activo.

4.4.1. Relación con la arquitectura de referencia

En la presente Sección se describe la relación de AIO TES con la arquitectura global. La Figura 4.2 muestra los componentes específicos de la arquitectura global que han sido seleccionados.

Empezando por abajo, se dispone del nivel de adquisición de datos. Este nivel es responsable de proporcionar conectividad entre los dispositivos físicos y los niveles superiores de la arquitectura. En esta primera instanciación de la arquitectura no se incluye la adquisición de datos a través de servicios web. Su objetivo es exclusivamente recopilar información del entorno y no es necesario enriquecer estos datos con otros servicios. Por ejemplo, ningún caso de uso tiene dependencia con el tiempo atmosférico o el estado del tráfico.

El nivel de *middleware* se encarga de proporcionar conectividad y procesamiento de datos entre los dispositivos y los niveles superiores de la arquitectura. También proporciona tareas de integración y almacenamiento temporal de datos. En esta implementación concreta, este nivel está formado por todas las plataformas IoT que componen el ecosistema. Dependiendo de las necesidades de los pilotos, algunos pueden incorporar *gateways*, pero en última instancia todos utilizan plataformas IoT para gestionar los datos de los sensores.

A continuación, el nivel de datos semántico de la arquitectura global se ha desdoblado en dos niveles: nivel de interoperabilidad semántica y nivel de datos. El nivel de interoperabilidad semántica se encarga de garantizar que los diferentes componentes de la arquitectura puedan comunicarse y trabajar juntos de manera eficiente utilizando protocolos de comunicación y estándares comunes. Debido a la particularidad de que esta arquitectura involucra a múltiples plataformas IoT, el adaptador semántico a implementar debe permitir la interoperabilidad entre todas ellas.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

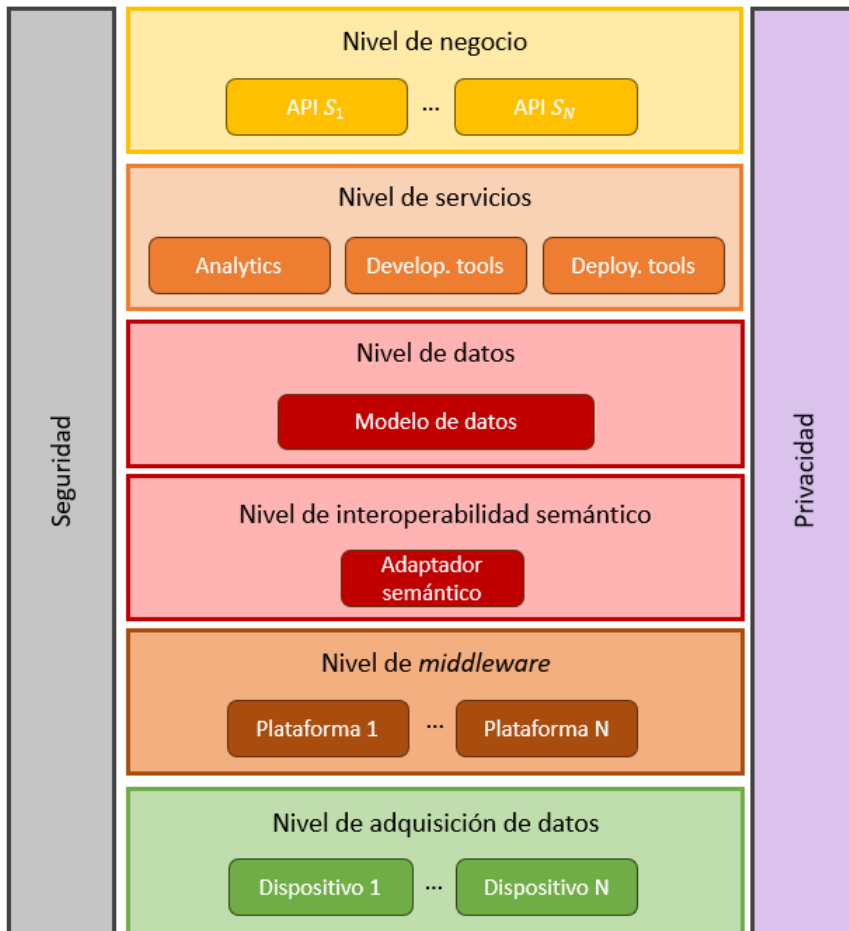


Figura 4.2: AIoTES - Comparativa con arquitectura de referencia.

Por otro lado, los modelos de datos tienen un gran peso dentro de la arquitectura y por eso se le asigna un nivel propio: el nivel de datos. Este nivel está principalmente formado por el modelo de datos común del ecosistema: una ontología central compatible con los modelos de datos de todas las plataformas. La ontología central incluye, a su vez, el resto de los modelos de datos de las plataformas del ecosistema.

Seguidamente se sitúa el nivel de servicios. Tiene una capacidad cognitiva débil, basada en *big data*. Es por esta razón que no se le atribuye el adjetivo “cognitivo”. Estos servicios proporcionan la capacidad de realizar análisis y obtener información a partir de los datos recopilados por las plataformas. Incluye funcionalidades como el análisis de tendencias, la detección de patrones y la predicción de futuros eventos. Además de esto, en este nivel se engloban herramientas destinadas al desarrollo, despliegue y gestión del ecosistema.

El nivel de negocio está formado únicamente por una API global de la arquitectura. Esta API reúne las APIs de todos los servicios de la arquitectura y facilita el uso e integración de estos servicios. Proporciona un punto de acceso único para interactuar con los servicios, lo que simplifica la integración y el acceso a las funcionalidades de estos. Así, la API simplifica la adopción de los servicios en los pilotos.

Los niveles de seguridad y privacidad deben garantizar que la arquitectura respete plenamente el derecho a la intimidad de los usuarios y actúe de acuerdo con las regulaciones del RGPD.

4.4.2. AIoTES

Como puede verse en la Figura 4.3, el *framework* AIoTES está compuesto por diferentes niveles: nivel de dispositivos, nivel de *middleware*, nivel de interoperabilidad semántica, nivel de datos, nivel de servicios, API AIoTES y nivel de seguridad y privacidad.

Nivel de dispositivos

El nivel de dispositivos proporciona información relevante sobre el comportamiento de los usuarios. Este nivel ofrece la función básica de acceso a la información y, además, se encarga de dar soporte a protocolos específicos. Por un lado, ofrece el nivel de control, que permite la capacidad de escribir, en general, actuar sobre dispositivos específicos como sistemas de ventilación y aire acondicionado, control de energía (enchufes inteligentes), iluminación, etc. Por otro lado, ofrece la actuación de los dispositivos. Tras analizar la información de su entorno, los dispositivos pueden ejecutar acciones específicas de manera autónoma o en respuesta a condiciones predefinidas.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

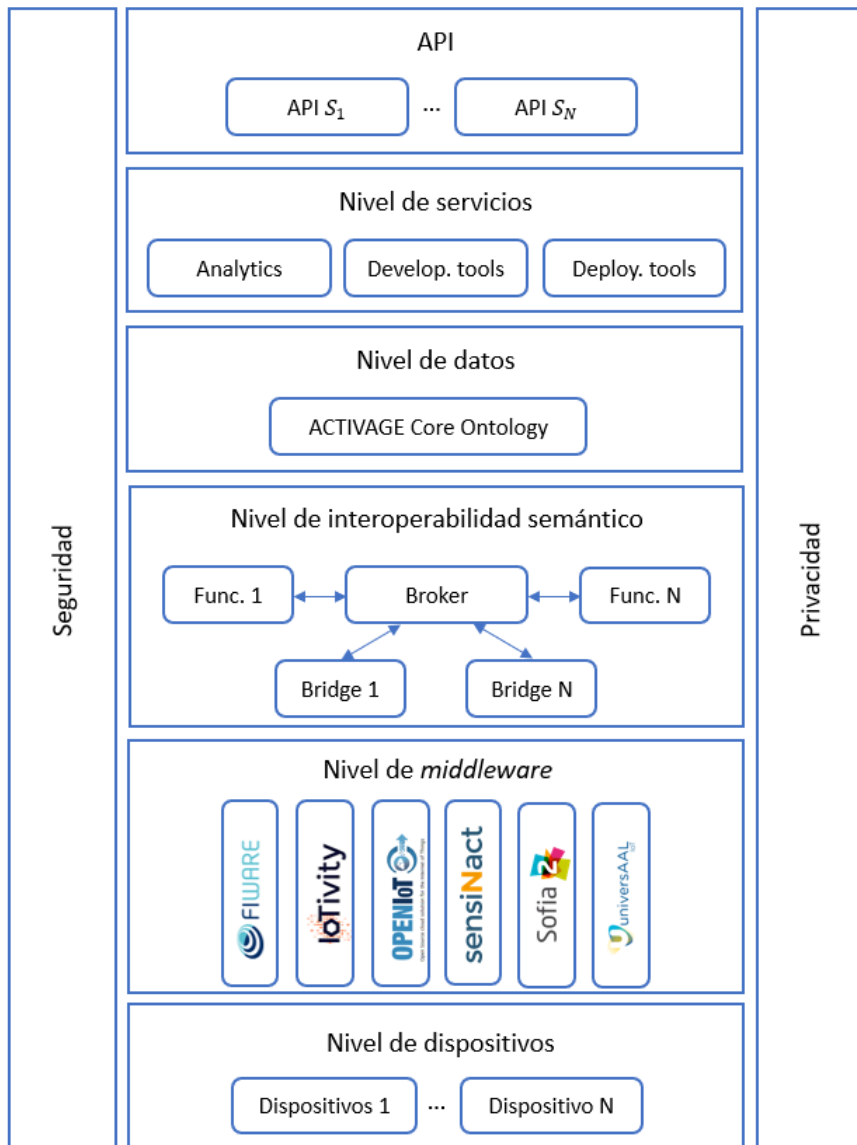


Figura 4.3: AIoTES.

En este nivel están incluidos los agentes IoT. Estos agentes son módulos *software* encargados de conectar los dispositivos con las plataformas IoT. Aseguran el acceso a la información de los dispositivos a través de unas interfaces homogéneas y seguras. Los agentes IoT mapean protocolos estándar como Bluetooth [51], ZigBee [50], Z-Wave [52], MQTT [145] u OMA LwM2M [146] con el nivel de *middleware*.

Nivel de *middleware*

El nivel de *middleware* está diseñada para incorporar nuevas plataformas IoT al ecosistema con facilidad. Está formada por el conjunto de plataformas IoT desplegadas en los pilotos: FIWARE [147], SOFIA2[148], universAAL [149], sensiNact [150], OpenIoT [151] y IoTivity [61].

Debido a la falta de estándares IoT, el acceso a los datos y dispositivos a este nivel depende de la sintaxis y el modelo semántico de datos particulares de cada plataforma. Por ese motivo, este nivel de la arquitectura es muy heterogéneo. En general, no es posible la comunicación directa entre diferentes plataformas de IoT, ya que emplean estándares, formatos de datos y semántica diferentes. Por este motivo, el intercambio y la replicación de los servicios basados en las plataformas se gestiona en el nivel inmediatamente superior.

Nivel de interoperabilidad semántica

El nivel de interoperabilidad semántica es el elemento que proporciona interoperabilidad semántica entre plataformas y aplicaciones, unificando en un formato común los datos obtenidos por el sistema. Este nivel realiza las conversiones necesarias para permitir un entendimiento común entre todos los elementos del *framework*. El uso de este nivel de abstracción simplifica sustancialmente la implementación de los mecanismos de interoperabilidad. Los niveles superiores son alimentados con datos provenientes de fuentes de datos heterogéneas con la peculiaridad de que estos datos han sido previamente unificados en un modelo de datos común. Esto permite explotar los datos del ecosistema no solo de forma local, sino también global.

Diseñar un adaptador semántico que permita la comunicación entre toda la combinatoria de modelos de datos del ecosistema no es escalable. Si se sigue este enfoque, en el momento de querer añadir una nueva plataforma es necesario diseñar nuevos mecanismos de interoperabilidad entre la nueva plataforma y las anteriores. Por ese motivo, para la implementación de este nivel se ha seleccionado una solución previa que permite transformar los datos a un formato común. Los componentes son INTER-MW [152] y *Inter Platform Semantic Mediator* (IPSM) [153]. INTER-MW es el componente encargado de gestionar

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

la comunicación entre plataformas IoT a través de un *message broker* mientras que el IPSM gestiona los mapeos semánticos entre emisores y receptores y proporciona interoperabilidad semántica mediante la traducción entre las diferentes ontologías de las plataformas IoT y el modelo de datos de ACTIVAGE.

Para añadir una nueva plataforma al ecosistema primero se debe añadir en el nivel de *middleware*. Después, se debe registrar la plataforma en INTER-MW y realizar los alineamientos sintácticos y semánticos necesarios entre el formato de datos de la plataforma y el formato de datos de ACTIVAGE.

Nivel de datos

El modelo de datos recibe el nombre de ACTIVAGE Core Ontology [154] y está diseñado para facilitar el intercambio de datos procedentes de diferentes ecosistemas locales de envejecimiento activo. Cada piloto tiene su propio modelo de datos y, por tanto, inicialmente no hay interoperabilidad entre ellos. Para pasar de un enfoque local a uno global, se crea una ontología central compatible con todo el ecosistema. Los requisitos del modelo de datos se obtuvieron teniendo en cuenta las particularidades de cada uno de los pilotos y sus casos de uso, que consideran sus plataformas, tecnologías y servicios IoT.

ACTIVAGE Core Ontology se complementa con otras ontologías como la ontología genérica de Plataformas IoT (GOIoTP [155]), que supone la terminología central para la interoperabilidad de la ontología ACTIVAGE. Esta ontología se desarrolló en el marco del proyecto H2020 INTER-IoT [156]. Existen otras ontologías de otros proyectos de investigación relacionados con IoT como BIG IoT [157], FIESTA IoT [158], VITAL IoT [159], y OpenIoT [160] cada uno de los cuales añade y amplía las especificaciones de IoT y los servicios asociados.

Nivel de servicios

El nivel de servicios es el encargado de proporcionar funcionalidades y servicios comunes en todo el ecosistema. Es un elemento clave para interconectar servicios de forma global, para lo cual este nivel se sitúa por encima del nivel de *middleware* el nivel de interoperabilidad semántica y el nivel de datos, aprovechando la abstracción establecida en los niveles inferiores. Los pilotos pueden utilizar cualquiera de los servicios, independientemente de su plataforma IoT. Este nivel se beneficia de la homogeneización de los datos procedentes del nivel anterior. El nivel de servicios incluye tres componentes principales: componente de análisis, herramientas de desarrollo y herramientas de despliegue.

El objetivo del componente de análisis es facilitar la extracción de información útil recogida de los sensores de IoT para apoyar la toma de decisiones

humanas. En los niveles inferiores de la arquitectura se recogen y almacenan datos en bruto en formatos específicos de las plataformas IoT. Aunque el nivel de interoperabilidad semántico unifica todas estas representaciones de datos diferentes en una semántica común, es necesario un procesamiento adicional para extraer conocimiento de los datos disponibles. El componente de análisis está formado por tres subcomponentes:

- *Data lake*: Es una infraestructura para almacenar y procesar un gran volumen de datos procedentes de sensores de forma distribuida.
- *Data analytics*: Permite extraer información de alto nivel mediante la aplicación de métodos de *big data*. También proporciona métodos para la extracción de la información significativa para los humanos, como los resúmenes de los datos y los patrones detectados en ellos.
- *Visual analytics*: Contiene componentes capaces de presentar los resultados de los análisis de forma intuitiva y completa.

Por otro lado, las herramientas de desarrollo son un conjunto de aplicaciones web que simplifican el uso de los componentes de AIoTES. Gracias a ello, se facilita el desarrollo de nuevas aplicaciones sobre el ecosistema.

Las herramientas de despliegue engloban una serie de aplicaciones web diseñadas para agilizar la implementación de AIoTES dentro de los pilotos. Estas herramientas simplifican la creación, modificación y mantenimiento del despliegue. También permiten supervisar las plataformas IoT y dispositivos del sistema. Cabe destacar que han sido diseñadas para integrarse con el entorno Docker [161].

API AIoTES

La API de AIoTES proporciona un punto de acceso único al (*framework*). Expone todas las funcionalidades de AIoTES a través de un conjunto unificado de operaciones RESTful. Ofrece acceso tanto a los componentes *back-end* como a las interfaces de usuario. La API trabaja conjuntamente con el nivel de seguridad y privacidad para permitir la autenticación, autorización y confidencialidad de los usuarios. Además, utiliza HTTPS y requiere la autenticación del usuario.

Nivel de seguridad y privacidad

El nivel de seguridad y privacidad garantizan protección de la información sensible, cumpliendo los requisitos éticos y legales de privacidad y confidencialidad. Los bloques funcionales que lo constituyen son los siguientes:

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

- ***Policy Administration Point (PAP)***: Se encarga de crear, actualizar, eliminar y gestionar las políticas.
- ***Policy Decision Point (PDP)***: Evalúa y emite decisiones de autorización basadas en las políticas de la Base de Datos de Políticas y del usuario o servicios que solicitan alguna acción.
- ***Policy Enforcement Point (PEP)***: Se encarga de interceptar las solicitudes de acceso de los usuarios a un recurso y hacer cumplir la decisión del PDP.
- ***Policy Information Point (PIP)***: Proporciona información externa a un PDP, como información de atributos LDAP.

Este nivel se complementa con un *Identity Management (IDM)* que desempeña el papel de proveedor de identidad para usuarios y servicios, y de PIP para el PDP. El objetivo de estos componentes (PEP, PIP, PDP, PAP e IDM) es proporcionar una visión clara y cohesionada de las funciones y flujos que deben ejecutarse en el sistema para lograr un proceso seguro de autenticación y control de acceso. El flujo de despliegue e integración se basa en OpenID Connect [162] para la identificación y autenticación, JSON Web Token, y un conjunto de mecanismos de control de acceso.

4.5. Validación

La metodología empleada para la verificación y validación de AIOOTES se define en dos fases que se detallan a continuación.

4.5.1. Fase 1: Verificación de componentes

La verificación de componentes consiste principalmente en la realización de *tests* unitarios de componentes antes de su integración con el sistema. El objetivo de los *tests* unitarios es verificar que cada componente de AIOOTES cumple sus especificaciones. En esta fase, los *tests* de integración se limitan a aquellos cuya interdependencia exige que estén integrados y, por tanto, es necesario probarlos juntos.

4.5.2. Fase 2: Validación en pilotos

Esta fase consiste en el despliegue y validación de AIOOTES en un entorno real. Durante la verificación de los componentes y la validación de la integración, es posible probar aspectos y características clave que necesitan la instalación

y configuración completas del *framework*. Sin embargo, la validación en pilotos se centra en garantizar el correcto funcionamiento, el rendimiento adecuado y la aceptación por parte del usuario de los servicios del ecosistema en lugar de probar la tecnología.

4.5.3. Escenarios de validación

Teniendo en cuenta que cada piloto tiene casos de uso diferentes, se define un conjunto de escenarios de validación que cubren todos los casos de uso y los objetivos de AIO TES. Los escenarios de validación se definen para garantizar una validación homogénea de los pilotos. Por consiguiente, cada escenario describe una configuración y unas condiciones tecnológicas genéricas, lo suficientemente abiertas como para que los pilotos puedan cumplirlas, pero lo suficientemente concretas como para garantizar que los aspectos específicos de AIO TES estén garantizados.

Los escenarios de validación se clasifican en dos categorías:

- **Interoperabilidad:** Examina los elementos básicos mínimos necesarios para el uso mínimo de AIO TES, diferenciando interoperabilidad sintáctica y semántica.
- **Funcionalidad:** Agrupa un conjunto de componentes según su uso lógico.
 - **Federación de *data lakes* y análisis de datos:** Permiten almacenar, compartir y analizar *datasets*. Los *datasets* pueden analizarse y los resultados de este análisis pueden indicar cómo perfeccionar los servicios, crear nuevos o incluso crear nuevas formas para representar servicios. El procesamiento de los datos de los pilotos permite extraer de ellos conocimientos potencialmente nuevos.
 - **Seguridad y privacidad:** Incluye la prestación de servicios integrales de seguridad interna y privacidad para el piloto. Esto incluye no solo la protección de las comunicaciones, sino también el uso de mecanismos de autenticación y autorización.
 - **Marketplace:** Permite la distribución de aplicaciones del ecosistema. El *marketplace* no es una contribución de este trabajo, pero ha sido incluido y validado durante el desarrollo de AIO TES.
 - **Herramientas de desarrollo y despliegue:** Facilitan la adopción de AIO TES a los desarrolladores para que estos creen nuevas aplicaciones, adapten las existentes o simplemente las desplieguen.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

Para definir cómo cada piloto puede implementar cada escenario de validación se crea una matriz de validación (Sección 4.6). Así, tras al menos dos meses de funcionamiento continuado con AIoTES, se recopilaban comentarios y datos. Se establecieron entrevistas individuales entre el equipo de evaluación de AIoTES y cada piloto, definiendo preguntas básicas para todos los escenarios de validación, así como la recopilación de datos específicos para cada escenario y comentarios generales de diferentes facetas del *framework*.

4.6. Resultados y discusión

En esta Sección se presentan los resultados cuantitativos y cualitativos obtenidos siguiendo la metodología de validación descrita en la Sección anterior. El *framework* AIoTES se ha validado con éxito. En primer lugar, se verifica el funcionamiento de sus componentes. Posteriormente, se despliega y valida el sistema en un entorno real con 4345 usuarios directos.

4.6.1. Verificación de componentes

Los resultados del primer nivel de validación prueban que los componentes individuales de AIoTES cumplen con la funcionalidad esperada. La Tabla 4.1 resume la validación de componentes. Con 110 módulos independientes y 441 *tests* individuales, esta fase se caracteriza por analizar sistemáticamente las principales funcionalidades de la implementación. Tras obtener un porcentaje global de aprobados del 94%, se demuestra que la mayor parte de la implementación funciona.

Objetivo	Componente	Módulo	Tests	Éxito
Interoperabilidad	Sintáctico	6	138	86 %
	Semántico	18	18	100 %
<i>Big data</i>	<i>Data Lake</i>	4	27	100 %
	<i>Analytics</i>	23	32	97 %
Ecosistema	<i>Marketplace</i>	1	16	100 %
	<i>Dev. & Depl. tools</i>	21	110	96 %
	KPI	1	1	0 %
Seguridad y privacidad	Seg. y priv.	1	8	100 %
	AIoTES API	22	108	100 %
Total		110	441	94 %

Tabla 4.1: Resultados de la evaluación de los componentes.

La distribución de los *tests* se concentra en torno a los módulos de interoperabilidad y ecosistema. En cuanto a los componentes con mejores resultados, el *marketplace* y los componentes de seguridad y privacidad superan todas sus pruebas. Esto también ocurre con el componente de interoperabilidad semántica, el *data lake* y la API. El componente KPI muestra un resultado negativo, aunque solo tiene un *tests*, no lo supera. Esto se debe a los retrasos en la implementación de este componente, que se extendió más allá de este primer paso de validación; por lo tanto, no fue posible probarlo al mismo tiempo que los otros módulos.

4.6.2. Validación en pilotos

El último paso de la validación se lleva a cabo a través de los distintos escenarios de validación (Sección 4.5). Se hace hincapié en los datos cualitativos, que se recogen mediante entrevistas al final de un periodo de al menos dos meses de ejecución continua del experimento. La Tabla 4.2 resume los perfiles de los participantes en estas entrevistas. Los participantes son 16 personas que representan a cada piloto desde las perspectivas organizativa y técnica. Los participantes tienen distinta formación. Si bien la mayoría cuenta con una formación tecnológica, también hay matemáticos y economistas; además, la mayoría tiene un máster. En cuanto a la experiencia en su sector, la media es de casi 15 años, mientras que su comprensión del proyecto se denota por la media de casi tres años de participación. Hay el doble de hombres que de mujeres, y la distribución por edades es mayoritariamente uniforme en las franjas de 26-50 años, con mayor representación en la franja de 36-40 años.

Dada la heterogeneidad de los escenarios de validación, no todos ellos implementan todas las funcionalidades de AIoTES y, por tanto, no se puede validar las funcionalidades en todos. Por este motivo, se define la matriz de validación (Tabla 4.3). Esta matriz está basada en los escenarios de validación y muestra la complejidad percibida (en una escala de 0 a 10, donde 0 es trivial y 10 es extremadamente complejo) por los pilotos para implementar los distintos escenarios de validación. Es importante destacar que la matriz de validación está estrechamente relacionada con los objetivos generales de AIoTES (Sección 4.2). Algunos implementan un escenario de validación de dos formas distintas e independientes. Los espacios en blanco de la Tabla reflejan los escenarios de validación en los que el piloto no participa, mientras que N/A representa a un piloto que participa en el escenario de validación, pero no puede proporcionar datos fiables. Es decir, un piloto se beneficia de una funcionalidad, pero puede evaluar la complejidad de su aplicación.

AIoTES demuestra la capacidad de interoperar entre diferentes plataformas IoT, proporcionando interoperabilidad sintáctica y semántica a través de

**CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA
CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE
DATOS IOT HETEROGÉNEOS**

Género	Masculino	11
	Femenino	5
Edad (años)	26-30	2
	31-35	2
	36-40	5
	41-45	3
	46-50	3
	50+	1
	Educación ([163])	6-Licenciatura
7-Máster		10
8-Doctorado		4
Área de Educación ([164])	12-Matemáticas	6
	33-Ciencias tecnológicas	9
	53-Economía	1
Experiencia (años)	Media	14.88
	Desviación estándar	9.47
Tiempo en proyecto (meses)	Media	34.19
	Desviación estándar	10.63

Tabla 4.2: Perfil de los participantes en la validación,

	Interop.		Funcionalidad					
	Sintáctica	Semántica	Data Lake federation	Seguridad	Marketplace	Data Analytics	Development tools	Deployment tools
Galicia	7.5	4		N/A			3.5	3.5
Valencia	5	8.5	5		2			
Madrid	7.5	8	7.5	3	1		3	4
RER	9	8			1			
Grecia	9	10	8		2	5	6	6
Isere	6	9		3	1			
Woquaz	5	8			N/A	N/A	N/A	N/A
Leeds	7	8.5		N/A	N/A			
Finlandia	8	9	8					
Barcelona	7	8.5						
Sofia	3	6	6					
Lisboa	7	0						

Tabla 4.3: Complejidad subjetiva percibida (de 0 a 10) de los EV (columnas) en los pilotos (filas) según sus representantes.

sus plataformas. Los participantes entrevistados lo corroboran afirmando que, entre todas los pilotos, hay un total de 2 540 usuarios conectados directamente al ecosistema. Para lograr interoperabilidad semántica, se mapean los modelos de datos individuales y genéricos al modelo de datos común de ACTIVAGE. Los doce pilotos logran alcanzar interoperabilidad sintáctica y semántica en entornos reales. Sin embargo, es importante destacar que proporcionar interoperabilidad entre plataformas no es trivial. Como puede verse en la Tabla 4.3, de forma general, la complejidad percibida por los usuarios es alta. La implementación de los mecanismos necesarios para proporcionar esta comunicación requiere cierto nivel de experiencia en el campo de la semántica. Adquirir estos conocimientos requiere de una inversión grande de tiempo.

La validación de las funcionalidades se realiza a distintos niveles.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

El *marketplace* se alimenta con un total de 17 aplicaciones. Se registran 72 usuarios, 27 de los cuales lo hacen con perfil de desarrollador. En total, hay 83 descargas de aplicaciones, donde la aplicación más popular es la herramienta de interoperabilidad semántica, con 12. Esta información está disponible en la página de inicio del *marketplace* [165]. Se puede acceder a información más detallada y a las tendencias a través de sus paneles de administrador y desarrollador.

En relación con las herramientas de desarrollo y despliegue, los pilotos señalan que han utilizado con éxito estas herramientas para reducir el tiempo de desarrollo de varias tareas. Las más destacadas son: el alineamiento semántico de ontologías, la generación de código de representación de ontologías para plataformas concretas, la composición de servicios complejos a partir de otros más sencillos y la creación de *dashboards*.

Durante el desarrollo de AIO TES se propuso un reto específico para integrar soluciones de análisis de datos nuevas y especializadas en los pilotos. La propuesta de este reto fue FILOS. Su objetivo es fomentar la autonomía de las personas mayores, apoyando sus capacidades cognitivas vinculadas a la memoria gracias a la adopción de herramientas informáticas de asistencia basadas en IoT, ML y tecnologías móviles. FILOS puede acceder a datos históricos de un usuario concreto y, a partir de ahí, construir modelos y estimar la capacidad cognitiva del usuario. El sistema se integra con éxito con el *data lake* y los servicios de datos históricos de AIO TES. Se valida así la capacidad del sistema de continuar creciendo y enriqueciéndose.

Se propuso otro reto relacionado con la seguridad y la privacidad. El reto se centraba en proporcionar nuevos mecanismos para proteger el sistema. En respuesta a este reto, se presentaron métodos de autenticación fáciles de utilizar por los usuarios en forma de dos soluciones. El sistema *Authentication Hub* ofrece la posibilidad de identificar a los usuarios mediante tarjetas RFID. *BehavAuth* permite una autenticación de comportamiento continua y multimodal mediante el uso de los sensores internos del smartphone. Ambas soluciones se integran en el *framework*. Cada solución se prueba en dos pilotos independientes, en total cuatro, integrando y ampliando el módulo de seguridad de AIO TES y permitiendo a las aplicaciones nativas autenticar a los usuarios de forma transparente. Ambos casos demuestran la capacidad del *framework* para incorporar nuevos mecanismos de autenticación, mediante el uso de estándares, e incorporarlos de forma transparente a las aplicaciones AIO TES.

4.7. EBASI

La presente Sección resume el diseño, implementación y validación de la arquitectura *iot based Emotion and Behaviour recognition Against elderly people Social Isolation* (EBASI), realizada en colaboración con el proyecto H2020 EBASI [166]. Esta arquitectura proporciona una solución adicional al ecosistema ACTIVAGE. El conocimiento adquirido durante la implementación de la arquitectura anterior se traslada para diseñar una solución compatible que pueda ser fácilmente incorporada dentro del ecosistema y que contribuya a su expansión. Así, EBASI contribuye al ecosistema mediante la detección y abordaje del aislamiento social y la soledad de las personas mayores.

Caso de uso: reconocimiento de la soledad basado en sensores multimodales

El aislamiento social es un concepto multidimensional que carece de una definición clara y consistente en la literatura [167]. Algunos expertos lo ven como directamente equivalente a la soledad y usan los términos indistintamente, mientras que otros perciben los dos conceptos como relacionados pero distintos. Por ejemplo, el aislamiento social se ha definido como la ausencia de contacto con personas que brindan apoyo social [168]. Otros lo han definido como un concepto bidimensional que contiene una ausencia objetiva de contactos o interacciones con los contactos y un sentimiento subjetivo de compañerismo o apoyo social limitado o perdido (es decir, soledad) como resultado de tener contactos o interacciones limitados.

La principal causa del aislamiento social es la ausencia o relativa escasez de relaciones sociales [169], que representan un importante factor de riesgo para la salud comparable, si no superior, a los factores de riesgo bien conocidos como el tabaquismo, el abuso del alcohol y la obesidad. En particular, las personas mayores, también se ha encontrado que el aislamiento social es una causa importante a la disminución de las capacidades cognitivas y, en general, a un peor estado de salud, tanto psicológico y físico.

Las relaciones sociales pueden influir en el estado de salud a través del intercambio de información, apoyo emocional, así como ayuda material, que a su vez promueven el comportamiento adaptativo en la presencia de fuentes agudas o crónicas de estrés [170]; pero la red de relaciones puede actuar también, directa o indirectamente, en la promoción de la adopción de conductas saludables. Finalmente, también hay evidencia de una relación directa entre el aislamiento social y la salud [171]. Para personas mayores solitarias, perder el contacto con sus seres queridos tiene un impacto negativo que puede superarse manteniéndose en contacto con antiguos compañeros de trabajo, familiares y

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

amigos. Además, involucrar a las personas mayores, aunque sea en actividades simples, puede reavivar su entusiasmo perdido.

Además, se busca fomentar un estilo de vida saludable en los mayores [172]. Esto es fundamental para retrasar el envejecimiento, prevenir las enfermedades crónicas y, en general, optimizar la salud y el bienestar. Por último, otro de los propósitos es la ayuda a las actividades de cuidado de los cuidadores profesionales, sociales y familiares. La monitorización de las personas mayores aporta una mayor confianza en el manejo de la condición del paciente tanto a este como a sus cuidadores. Esto proporciona paz mental a ambas partes, evitando que su relación se deteriore o que incluso se produzca el síndrome del cuidador quemado.

Este caso de uso consiste en el diseño y validación de un sistema para la monitorización de personas mayores jubiladas que viven solas en sus hogares. Gracias a una red de sensores IoT y un análisis cognitivo de los datos recopilados, el sistema detecta comportamientos que pueden ser síntoma de aislamiento, los analiza y en caso de detectar algún patrón asociado a la soledad interactúa con el usuario o envía una notificación a sus cuidadores.

El primer paso del diseño del caso de uso consiste en la identificación de actividades conductuales y actitudes que son síntomas de social aislamiento y soledad para, en consecuencia, definir los parámetros e indicadores que se necesita monitorizar. Identificar las causas y los indicadores relacionados con el aislamiento ayuda a definir el tipo de interacción necesaria para animar a los ancianos o cómo interactuar con la familia y/o cuidador.

El objetivo es que el sistema sea capaz de medir la soledad de los ancianos automáticamente. La soledad depende de varias cosas, como los procesos mentales y los procesos físicos. Así, se puede medir el aislamiento de las personas mayores en tres ejes (Figura 4.4):

- Aislamiento social.
- Movilidad funcional.
- Autocuidado.

El autocuidado es un claro indicador del estado de ánimo. Las personas que se sienten tristes tienden a descuidar su apariencia física y, al mismo tiempo, esto empeora aún más su estado emocional. El aislamiento social se mide por la cantidad de actividad social. La movilidad funcional ayuda a reducir los síntomas de tristeza, genera una sensación de relajación e impacta positivamente en la calidad de vida de la persona. El cálculo de la soledad de cada uno de los tres ejes se realiza a través de las acciones diarias. Cuanto mayor es la puntuación de cada eje, menor es la sensación de soledad.

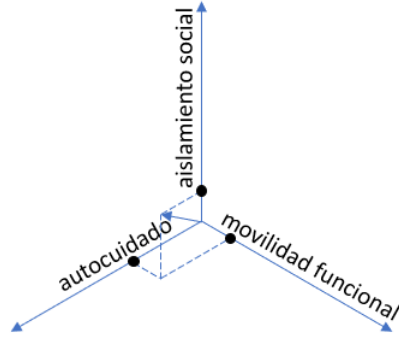


Figura 4.4: Modelo de soledad.

	Actividad	Frecuencia	Acción
Autocuidado (SADL)	Ducha	2 veces por semana	Interacción
		<2 veces por semana	Alerta
	Baño	3-5 veces por semana	Interacción
		<3 veces por semana	Alerta
Movilidad funcional (ADL)	Dormir	8-10 h por día	Interacción
		10-12h por día	Alerta
	Permanecer sentado	2-3h por día	Interacción
		3-5h por día	Alerta
	Ir fuera	< 1h cada 2 días	Interacción
		< 1h cada 4 días	Alerta
Actividades sociales (SADL)	Jugar o leer	< 1 vez por semana	Interacción
		< 1 vez por mes	Alerta
	Ver televisión	< 1h cada 3 días	Interacción
		< 1h por semana	Alerta
	Hablar por teléfono	< 3 veces por semana	Interacción
		< 1 vez cada 2 días	Alerta

Tabla 4.4: Configuración por defecto del umbral de los parámetros ADL y SADL.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

Para detectar estos tres elementos, se utiliza los términos *Social Activities of Daily Living* (SADL) [173] y *Activities of Daily Living* (ADL): ADL son actividades rutinarias que la gente realiza como comer, bañarse, vestirse, ir al baño y moverse. Por otro lado, SADL se enfocan en el autocuidado y las actividades sociales que permiten medir el sentido de comunidad de la persona.

La Tabla 4.4 refleja el mapeo entre ADL y SADL y el modelo de soledad. Muestra la configuración por defecto del umbral de parámetros a partir del cual se inicia la fase de interacción y alertas. Estos parámetros pueden ser modificados por el familiar/cuidador. Si el sistema detecta soledad, activará una interacción con los mayores o enviará una alerta a un familiar.

Arquitectura EBASI

Los objetivos de la arquitectura EBASI son los siguientes:

1. Evitar la sensación de soledad y el aislamiento social y mejorar la calidad de vida.
2. Permitir el diagnóstico precoz de los trastornos psicológicos.
3. Mantener la capacidad funcional de los mayores a largo plazo, cuya disminución es una consecuencia directa del aislamiento social.
4. Mejorar las relaciones emocionales entre las personas mayores, la familia y los cuidadores.

La Figura 4.5 muestra los diferentes niveles que conforman la arquitectura EBASI. El nivel inferior está formado por una red de sensores IoT diseñada para reconocer las actividades descritas en la metodología. Este nivel recoge datos de la actividad y la vida social de los usuarios a través de diferentes sensores distribuidos en sus viviendas. Los elementos de este nivel también permiten interactuar con los usuarios.

El nivel inmediatamente superior está formado por un asistente domótico (*gateway*). Este es el punto de conexión con los dispositivos IoT. El asistente domótico preprocesa los datos en bruto recogidos por los dispositivos, los almacena, y es responsable de ofrecer conectividad con el nivel superior. Adicionalmente, el asistente envía señales a los sensores para interactuar con el usuario a través de ellos, activando diferentes tipos de interacciones. Por ejemplo, sugerir tomar una ducha o ver contenido multimedia.

A continuación se encuentra el motor de AI. Este es el elemento cognitivo de la arquitectura que se encarga de analizar los datos de los sensores para generar diagnósticos. El motor de AI mide el aislamiento de las personas mayores utilizando el modelo de soledad descrito en la Sección anterior. Para ello analiza las acciones de la vida cotidiana y detecta valores de riesgo.

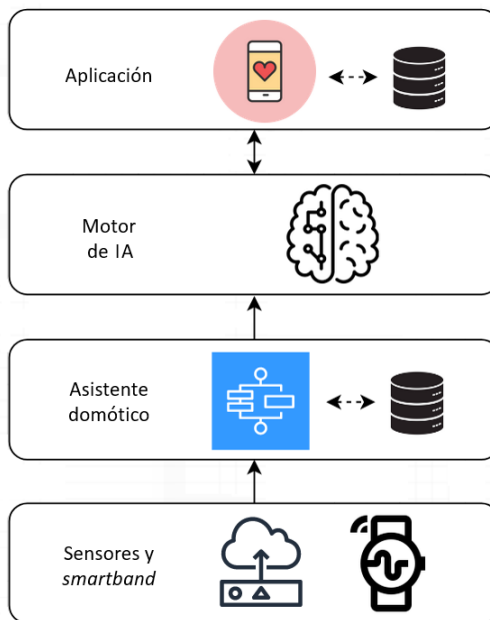


Figura 4.5: Arquitectura EBASI.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

A través de una aplicación dedicada, el nivel superior mantiene informada a los cuidadores y a la familia sobre la actividad y vida diaria de los usuarios. La aplicación genera alertas cuando se detectan situaciones de riesgo y genera estadísticas con el comportamiento de los usuarios.

La Figura 4.6 resume el flujo de trabajo del sistema. En él se distinguen tres fases principales:

1. Monitorización y detección.
2. Interacción y alerta.
3. Reporte y pronóstico.

Durante la fase de monitorización y detección, diferentes sensores IoT integrados en el sistema envían datos ADL y SADL. Los sensores IoT se instalan en cuatro espacios de la casa de la persona mayor: el baño, la cocina, la sala y el dormitorio. Un elemento cognitivo se encarga de vigilar que el valor de los sensores IoT no superan el umbral establecido.

La fase de interacción y alerta comienza tan pronto como el sistema detecta que una ADL o SADL supera el umbral configurado. Dependiendo del valor de la medida recibida, el sistema enviará una interacción positiva a la persona o alertará al cuidador a través de una notificación móvil. La interacción incluye las siguientes acciones:

- Estímulos: Según las ALD detectadas, el sistema incentiva y estimula el comportamiento del usuario. El sistema puede sugerir ducharse, bañarse, moverse, salir o dormir.
- Incentivos: El sistema estimula al usuario a mejorar su SADL. Los incentivos incluyen hacer una llamada telefónica, mirar televisión o escuchar música.
- Invitaciones: Cuando el cuidador sube contenidos multimedia a la aplicación móvil se activa una interacción emocional. Cuando ocurre, la persona mayor recibe una invitación para ver el álbum.

La fase de reporte y pronóstico está diseñada para enviar al cuidador un informe periódico sobre los parámetros ADL y SADL. Cualquier alerta es registrada en el sistema y se incluye en los reportes. Los reportes muestran tendencias teniendo en cuenta el índice de mejora del comportamiento y el estado emocional del anciano. La aplicación móvil brinda a los familiares informes interactivos en tiempo real para monitorizar las actividades de las personas mayores y brindar alertas tempranas sobre el estado de aislamiento social. Además, permite acceder a registros históricos de actividades (registrados en la nube)

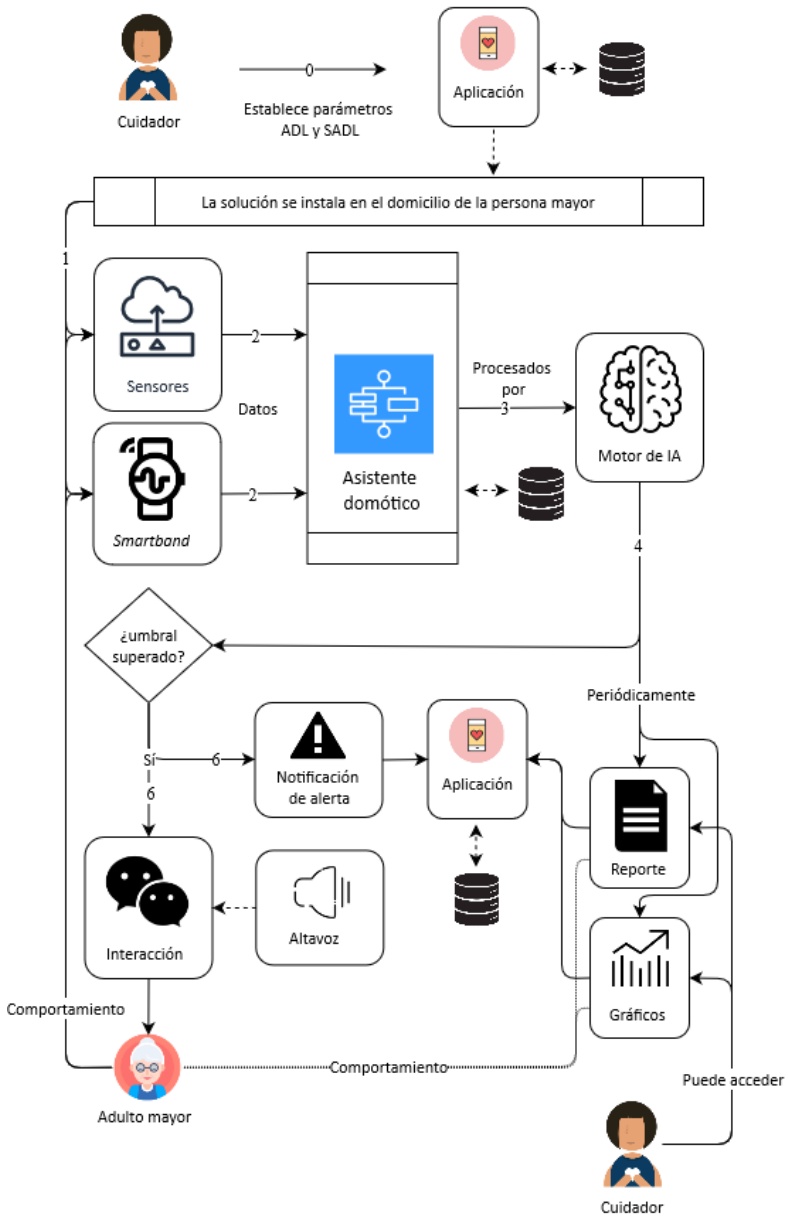


Figura 4.6: Flujo de trabajo.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

para determinar tendencias de aislamiento social y pronosticar la evolución de la situación.

Las interfaces de usuario del sistema son dos: una aplicación móvil para el cuidador y un altavoz de asistencia domótica para la persona mayor. Además de proporcionar informes periódicos, la aplicación móvil permite al cuidador establecer los umbrales de los parámetros ADL y SADL y envía notificaciones al cuidador en caso de situaciones críticas. Además, el sistema permite a los cuidadores subir contenido multimedia (fotos o videos) para que sea visualizado por la persona mayor. Cuando se inicia una fase de interacción, el sistema pregunta a la persona mayor (a través del asistente domótico) si desea ver en la televisión los videos o fotos que el familiar ha subido. Adicionalmente, el sistema puede recomendar escuchar música o llamar a un amigo o familiar.

Relación con la arquitectura de referencia

En la presente Sección se describe la relación de la arquitectura EBASI con la arquitectura global. La Figura 4.7 muestra los bloques seleccionados de la arquitectura global para la creación de la arquitectura EBASI. Esta arquitectura no implementa los niveles de seguridad y privacidad de la arquitectura de referencia.

El nivel de percepción está formado por los sensores IoT instalados en casa de los ancianos y la pulsera de actividad. Estos dispositivos permiten extraer información del mundo físico que posteriormente es analizada en las capas superiores. En esta instanciación no se necesita acceder a datos de terceros para enriquecer los datos de los sensores.

El nivel de *middleware* consiste en un asistente domótico que recopila la información de los sensores. Este realiza diferentes funciones como proporcionar gestionar la comunicación entre el sistema y el usuario a través del micrófono y altavoz. También realiza tareas de procesamiento y almacenamiento temporal de datos.

El nivel de datos semánticos engloba tanto la información generada por los sensores, los datos que los cuidadores ingresan a la aplicación y los datos históricos. Para garantizar una gestión eficiente y coherente de los datos, se emplea el formato de datos NGS-V2 [174] de FIWARE.

El nivel cognitivo de servicios permite al sistema entender, procesar y detectar los datos recopilados por los dispositivos para identificar patrones de soledad. Este nivel está formado por un motor de IA que consiste en un sistema de reglas. Este establece conjuntos de reglas personalizados para cada usuario, especificando las acciones a tomar ante diversos eventos. Se trata de un nivel cognitivo de servicios con una inteligencia débil.

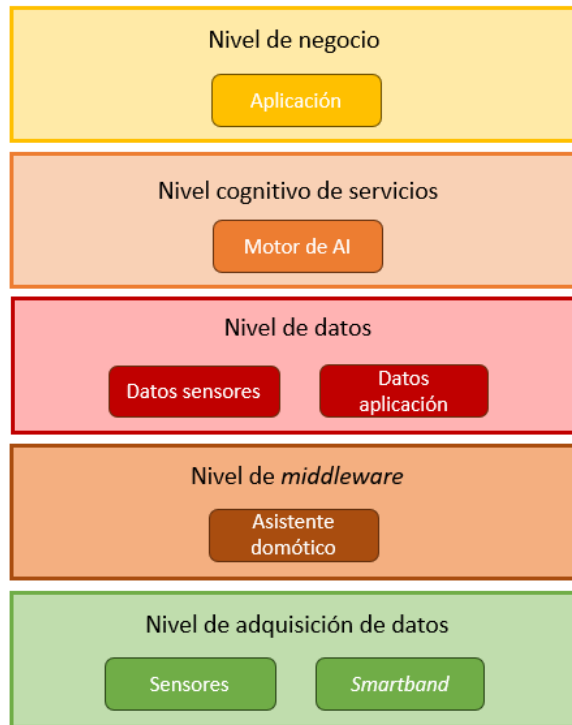


Figura 4.7: Arquitectura EBASI - Comparativa con arquitectura de referencia.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

Por último, el nivel de negocio está formado por la aplicación móvil de los cuidadores. Su objetivo es mejorar la comunicación, la supervisión y la calidad de vida de los ancianos, al proporcionar una plataforma intuitiva y eficaz para que los cuidadores se mantengan informados y conectados con los aspectos más importantes de la vida de aquellos a quienes cuidan.

Implementación

La Figura 4.8 muestra la implementación del sistema. Las fuentes de datos (parte inferior de la Figura) están compuestas por un conjunto de sensores IoT distribuidos estratégicamente en el hogar de las personas mayores y una pulsera inteligente (*smartband*) que se utiliza para detectar datos biométricos. La supervisión de las actividades de los usuarios la realizan los siguientes sensores: i) sensor de distancia ultrasónico HC-SR04 [175], ii) sensor de luz BH1750 [176], iii) sensor de presión MD30-60 [177] y iv) un detector de lluvia (AZDelivery) [178]. Para evitar problemas de funcionamiento, especialmente en el caso de los sensores IoT destinados a ser activados por el contacto con el agua, se han diseñado una carcasa que contiene los componentes electrónicos y la batería que alimenta los sensores. Además, se utiliza una *smartband* (Xiaomi Mi Band 3 [179]) para obtener los siguientes datos: frecuencia cardíaca, acelerómetro, giróscopo y pasos.

Una *Raspberry Pi 3* [180] actúa como controlador de los dispositivos y también como asistente domótico. Se utiliza el protocolo de comunicación *Wi-Fi Direct* [181] para conectar los sensores a la *Raspberry*. Para ello, cada sensor ha sido equipado con módulos *Wi-Fi* ESP8266 [182] para soportar *Wi-Fi Direct*. Además, se ha conectado un módulo a la *Raspberry*, que actúa como receptor. El asistente domótico interactúa vocalmente con el usuario mediante la API del asistente de Google, un micrófono (Anker Bluetooth Speaker [183]) y un altavoz (Movo MC 1000 [184]). La Figura 4.9 muestra los componentes que componen el asistente domótico instalado en la vivienda de los usuarios.

El *backend* es el corazón del sistema y está desarrollado en *Spring* [185] y *Java* [186]. Recibe los datos recogidos por la *Raspberry*, los analiza y, dependiendo de: i) la configuración de los parámetros ADL y SADL y ii) la gravedad de la anomalía, inicia una interacción con el usuario o avisa al cuidador mediante una notificación *push* en la aplicación móvil. Las capacidades cognitivas del sistema se han implementado siguiendo un enfoque lógico y basado en reglas. Un módulo *software* es el encargado de analizar los datos y, al detectar determinados patrones previamente establecidos, iniciar la interacción deseada. El sistema tiene también la función de recibir contenidos multimedia (fotos y vídeos), guardarlos y enviarlos para su visualización en la televisión. La configuración de los umbrales ADL y SADL y las mediciones de los sensores se

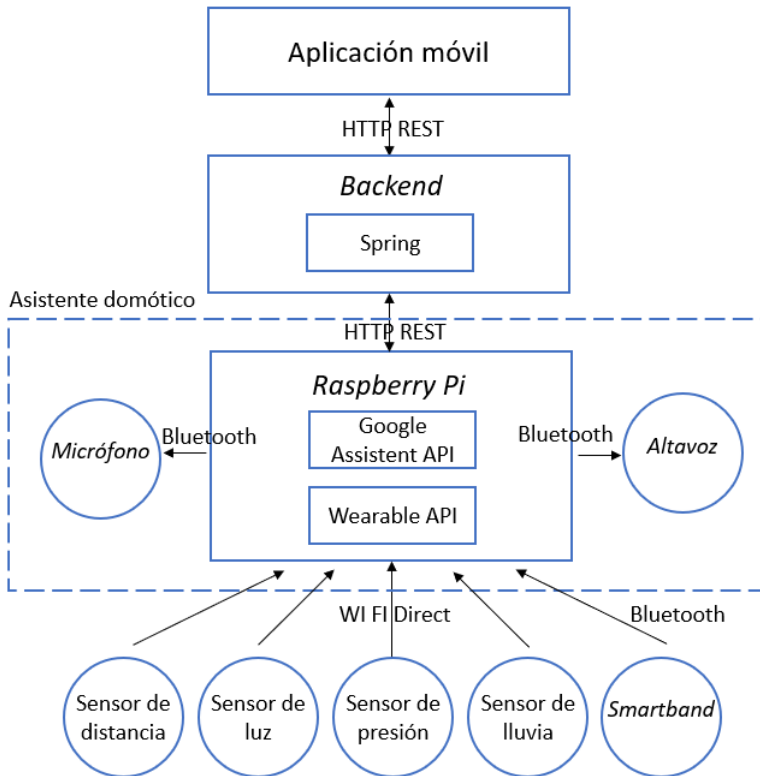


Figura 4.8: Implementación del sistema.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS



Figura 4.9: Asistente doméstico.

guardan en una base de datos MongoDB [187]. Por otro lado, el histórico de eventos, los videos y las fotos se guardan en una base de datos PostgreSQL [188] a la que accede la aplicación móvil del cuidador.

La aplicación móvil es una parte integral del sistema, donde tiene lugar la relación entre el cuidador, el asistente doméstico y el usuario. Sus funcionalidades han sido descritas en la Sección anterior. Es una aplicación móvil nativa de *iOS* desarrollada en *Swift* [189].

Todos los intercambios de datos entre la *Raspberry*, el *backend* y la aplicación del smartphone se realizan utilizando el protocolo HTTP [190] y el formato JSON [191].

Validación

Inicialmente, estaba previsto que el sistema se instalara en casa de cinco personas mayores durante 30 días. Sin embargo, la crisis derivada de la COVID-19 hizo imposible cumplir la planificación inicial, pues el confinamiento paralizó todas las actividades de instalación del sistema en casa de los usuarios finales. Sin embargo, el distanciamiento social provocado por el cierre de la COVID-19 permitió llevar a cabo el piloto dentro de los apartamentos de dos miembros del equipo de desarrollo. Es importante señalar que los usuarios reflejan el tipo de usuario final, ya que viven solos, salían de casa muy pocas veces y trabajaban en casa. Es evidente que en algunos casos fue necesario condicionar

el comportamiento real para verificar que el sistema registraba correctamente las ADL y las SADL.

El usuario A es un desarrollador *software* que trabajó desde casa durante el confinamiento. Dadas las restricciones, desde el 3 de marzo hasta el 18 de mayo de 2020, salía de casa de dos a tres veces por semana, principalmente para hacer compras. Trabajaba principalmente sentado en un escritorio o en el sofá durante una media de 8 horas al día, y a veces trabajaba por la noche. Para obtener resultados más relevantes, el usuario variaba su posición a lo largo del día. El usuario A no tiene pareja, pero tiene una buena relación con sus padres. Esto hizo que sus relaciones sociales fueran principalmente con familiares y amigos, pero a menudo escritas y no habladas (o al menos no durante largos periodos del día). A estas relaciones sociales, sin embargo, hay que añadir algunas reuniones por teleconferencia del trabajo. En su tiempo libre, el usuario jugaba principalmente a la *Playstation* y hacía rompecabezas.

El usuario B es un *project manager* que trabajó desde su casa durante el confinamiento del 3 de marzo al 18 de mayo de 2020. Solía salir a la calle una media de 3-4 veces a la semana. Su trabajo implica hacer muchas teleconferencias durante la semana. Su horario de trabajo era muy organizado y tenía un ritmo de sueño muy regular. El usuario B tiene pocos amigos cercanos y su afición favorita es la pintura. Su trabajo es menos sedentario, pues acostumbra a caminar durante las teleconferencias.

Los sensores se instalaron en sus casas para cubrir e identificar una amplia gama de actividades físicas y sociales. Para ello se tuvo en cuenta sus perfiles y la distribución de sus apartamentos. La Figura 4.10 muestra la ubicación de los sensores en los apartamentos del usuario A (izquierda) y del usuario B (derecha). Los umbrales de los parámetros se ajustaron para cada usuario, teniendo en cuenta su estilo de vida particular. Para el piloto se seleccionó un solo cuidador que gestionó a los dos usuarios a través de la aplicación móvil.

De acuerdo con los requisitos funcionales y técnicos, y gracias al uso de la aplicación móvil del cuidador, se han identificado dos niveles de alarma. En el primero, el sistema simplemente activa el asistente domótico a través de tres tipos de interacción (Sección 4.7). El segundo es un nivel de alerta en el que, además de crear la interacción con el usuario, el sistema envía una notificación de alerta al cuidador. El umbral de los parámetros de los usuarios quedan resumidos en las Tablas 4.5 y 4.6.

Para validar la predicción del sistema se proporcionó un cuestionario a los dos usuarios para verificar cuándo el sistema había generado falsos positivos en términos de:

1. Precisión en la frecuencia de detección.
2. Precisión de las horas de detección.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS



Figura 4.10: Piloto.

	Actividad	Frecuencia	Acción
Autocuidado (SADL)	Baño	≥ 2 veces por semana	Nada
		< 2 veces por semana	Interacción
		< 1 vez por semana	Alerta
	Ducha	≥ 7 veces por semana	Nada
		< 7 veces por semana	Interacción
< 3 veces por semana		Alerta	
Mobilidad funcional (ADL)	Dormir (noche)	6-10h	Nada
		> 10 h	Interacción
		< 6 h	Interacción
		< 3 h	Alerta
	Dormir (día)	2-3h	Nada
		> 3 h	Interacción
		> 6 h	Alerta
	Permanecer sentado	< 4 h por día	Nada
		> 4 h por día	Interacción
		> 6 h por día	Alarma
Ir fuera	> 2 h por semana	Nada	
	< 2 h por semana	Interacción	
	< 1 h por semana	Alerta	
Actividades sociales (SADL)	Jugar, leer, televisión o hablar	> 2 h por día	Nada
		< 2 h por día	Interacción
		< 1 h por día	Alarma

Tabla 4.5: Configuración del umbral de los parámetros ADL y SADL. Usuario A.

	Actividad	Frecuencia	Acción
Autocuidado (SADL)	Baño	≥ 4 veces por semana	Nada
		< 4 veces por semana	Interacción
		< 2 vez por semana	Alerta
	Ducha	≥ 5 veces por semana	Nada
		< 5 veces por semana	Interacción
		< 1 veces por semana	Alerta
Mobilidad funcional (ADL)	Dormir (noche)	7-10h	Nada
		> 10 h	Interacción
		< 7 h	Interacción
		< 6 h	Alerta
	Permanecer sentado	< 2 h por día	Nada
		> 2 h por día	Interacción
		> 3 h por día	Alarma
	Ir fuera	> 3 h por semana	Nada
		< 2 h por semana	Interacción
< 1 h por semana		Alerta	
Actividades sociales (SADL)	Jugar, leer, televisión o hablar	> 4 h por día	Nada
		< 4 h por día	Interacción
		< 0.5 h por día	Alarma

Tabla 4.6: Configuración del umbral de los parámetros ADL y SADL. Usuario B.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

3. Identificación de actividades ADL y SADL.
4. Adherencia de los tipos de interacción con las actividades detectadas.
5. Adherencia de las notificaciones de alerta con las actividades de seguimiento.

Resultados y discusión

La Tabla 4.7 contiene un resumen de los datos monitorizados y registrados por el sistema durante el piloto.

Usuario	Actividad	Datos reportados	Falsos positivos
A	Ducha	17	3
	Baño	16	2
	Dormir	176h	2
	Permanecer sentado	96h	N/A
	Ir fuera	8	0
	Jugar, leer, ver televisión o hablar	115h	5
B	Ducha	19	1
	Baño	13	7
	Dormir	207h	10
	Permanecer sentado	31h	4
	Ir fuera	16	1
	Jugar, leer, ver televisión o hablar	67h	3

Tabla 4.7: Datos reportados.

El sistema detectó falsos positivos para ambos usuarios. Los sensores instalados en la ducha detectaron nueve eventos erróneos. El sistema contabilizó las tareas de limpieza de la ducha como la acción de tomar una ducha o dejar el grifo abierto sin cerrarlo correctamente. Los sensores de la bañera detectaron errores causados por la colocación incorrecta de los sensores. En cuanto a las horas de sueño, debido a los movimientos y a las variaciones de los latidos del corazón durante la noche, la detección fue inexacta en algunos casos y es necesario pulir este punto. En relación con el tiempo sin hacer nada, mientras se trabajaba con el ordenador no se detectaron algunos movimientos menores

Usuario	Evento de motivación	Nº	Tipo	Alerta	Falsos positivos
A	Cuidador sube fotos o vídeos	5	Inv.	0	0
	Usuario duerme demasiado	2	Est.	2	1
	Horas de sueño insuficientes	12	Est.	0	3
	Frecuencia de ducha insuficiente	0		0	0
	Permanece inactivo demasiado tiempo	7	Inc. 2 E 5	2	0
	No sale de casa	0		0	0
	Insuficientes actividades lúdicas	1	Inc.	1	0
B	Cuidador sube fotos o vídeos	5	Inv.	0	0
	Usuario duerme demasiado	0		0	0
	Horas de sueño insuficientes	2	Est.	0	0
	Frecuencia de ducha insuficiente	1	Est.	0	0
	Permanece inactivo demasiado tiempo	2	Inc.	1	0
	No sale de casa	2	Est.	2	1
	Insuficientes actividades lúdicas	7	Inc.	0	0

Tabla 4.8: Interacciones del sistema.

que provocaron un registro de inactividad. No fue posible cuantificar el número de veces que esto ocurrió. En relación con el número de veces que los usuarios salieron de sus viviendas, en las instalaciones del usuario B no se detectó un evento porque la pulsera inteligente estaba descargada y, por tanto, no estaba conectada al sistema. Esto ocurrió porque las salidas de las viviendas se detectan como una combinación de dos eventos: el sensor de movimiento de la puerta detecta un evento y la señal *Bluetooth* de la *smartband* se pierde. Por último, se han registrado ocho falsos positivos en la detección de actividades lúdicas.

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

Esto se debe a que algunos ruidos/sonidos prolongados han sido interpretados como actividad, causando falsos positivos.

La Tabla 4.8 muestra las interacciones del sistema realizadas durante el piloto. Para el usuario A, solo se observaron falsos positivos relacionados con el número de horas de sueño. En algunos casos, el sensor de presión bajo el colchón siguió registrando la presión sin que el usuario estuviera realmente en él. En otras ocasiones, el sensor de presión dejó de registrar el evento porque el usuario se había alejado demasiado de la posición del sensor mientras dormía. Para corregir esto, también se tuvieron en cuenta los datos del acelerómetro de la *smartband* y de la frecuencia cardíaca para determinar si el usuario estaba o no dormido. En el caso del usuario B, sólo se produjo un falso positivo. El error se debió a que el usuario se olvidó de llevar la *smartband* al salir de casa y, por tanto, el sistema interpretó que el usuario estaba dentro de casa. Esto se debe a que el reconocimiento del usuario fuera de la casa fue diseñado para ser analizado tanto por un sensor de movimiento como por la pérdida de la señal de la *smartband*.

Considerando que el sistema está diseñado para la monitorización de personas mayores con un nivel de fragilidad significativo, los resultados reportan una cantidad considerable de falsos positivos. Esto puede atribuirse en parte al hecho de que el sistema actual se basa en un motor de IA basado en reglas predefinidas, lo que puede limitar su capacidad para detectar con la precisión requerida los comportamientos y situaciones específicas de los usuarios. Las reglas, al ser estáticas en su naturaleza, pueden no abarcar toda la variabilidad y matices de las situaciones reales, lo que contribuye a una tasa más alta de resultados falsos positivos.

Para mejorar la eficacia del sistema, una dirección prometedoras es la creación de un servicio cognitivo basado en técnicas de ML. Esto permitiría al sistema analizar y aprender de un conjunto amplio y diverso de datos recopilados, lo que podría llevar a una mejora significativa en la detección de patrones y comportamientos relevantes. Además, al utilizar modelos de ML, el sistema podría ajustarse dinámicamente a las características individuales de cada usuario y adaptarse a cambios sutiles en su estado de salud a lo largo del tiempo.

4.8. Conclusiones

En este Capítulo se han presentado dos instanciaciones iniciales de la arquitectura de referencia AIoT para la creación de servicios comunes en entornos de datos IoT heterogéneos.

AIoTES es una primera instanciación embrionaria de la arquitectura de referencia, diseñada para habilitar la creación de un ecosistema de soluciones para

el envejecimiento activo. Esta primera instanciación se centra en dos objetivos complementarios: interoperabilidad y creación de servicios comunes. Busca proporcionar interoperabilidad entre plataformas IoT, con el propósito último de facilitar la creación de servicios y herramientas compartidas. El *framework* ha sido satisfactoriamente validado por más de 2000 usuarios divididos en 12 pilotos.

El nivel de servicios proporciona acceso a los datos históricos de las plataformas IoT a través de un formato de datos común proporcionado por AIO-TES. Esto posibilita el desarrollo de métodos genéricos, basados en *big data*, dentro del componente *analytics*. Estos métodos pueden ser utilizados por cualquier piloto del ecosistema. Además, esta característica posibilita la creación de métricas comparativas entre los ecosistemas locales.

Por otro lado, gracias a la recopilación homogénea y el posterior análisis de los datos, el *framework* permite el futuro desarrollo y despliegue de servicios cognitivos avanzados. El uso de una ontología común permite la creación de *datasets* desvinculados de las plataformas de IoT. Estos *datasets* podrían contener información anonimizada de diferentes ecosistemas IoT locales, lo que los haría adecuados para la generación de *datasets* globales y posterior entrenamiento algoritmos de IA con estos.

EBASI es una primera aproximación de la arquitectura para la creación de servicios cognitivos basados en IA. Su incorporación al ecosistema, a través del nivel de interoperabilidad semántico de AIO-TES, añade una solución para detectar y abordar el aislamiento social y la soledad de las personas mayores. El elemento central de esta solución es su motor de IA mediante el cual se realiza el reconocimiento de patrones de soledad.

Inicialmente estaba previsto implementar y validar EBASI en la vivienda de los usuarios finales, pero debido a la crisis derivada de la COVID-19 no fue posible. En su lugar, los usuarios utilizados durante la validación del sistema fueron dos miembros del equipo de desarrollo. Debido al distanciamiento social ocasionado por el cierre de la COVID-19, estos usuarios reunían características similares a los usuarios finales. Para mejorar el sistema se sugiere crear un servicio cognitivo basado en algoritmos de ML que pueda aprender de datos variados y adaptarse a cada usuario para mejorar la detección de patrones y comportamientos relevantes y ajustarse a los cambios de salud de los usuarios con el tiempo.

De forma general, los resultados indican que la gran debilidad de AIO-TES es que su nivel de interoperabilidad resulta demasiado complejo. Pese a ser muy potente, los desarrolladores necesitan un entrenamiento intensivo, pues la curva de aprendizaje es muy grande. Este hecho dificulta su adopción. EBASI utiliza un enfoque opuesto. Emplea el modelo de datos de FIWARE para garantizar interoperabilidad. Utilizar un modelo de datos único simplifica sensiblemente

CAPÍTULO 4. ARQUITECTURA COGNITIVA PARA LA CREACIÓN DE SERVICIOS COMUNES EN ENTORNOS DE DATOS IOT HETEROGÉNEOS

el desarrollo y mantenimiento del sistema. Este hecho es clave y la motivación para utilizar FIWARE en la implementación del nivel de datos semántico de los siguientes Capítulos.

En cuanto a la creación de servicios cognitivos, ambas arquitecturas presentan un nivel de inteligencia limitada. ACTIVAGE se centra en la integración de datos para habilitar la construcción de servicios sobre estos ellos y la creación de servicios simples, mientras que EBASI se caracteriza por su enfoque en una IA basada en reglas. Los próximos Capítulos se centran a la elaboración de arquitecturas AIoT con servicios cognitivos avanzados.

Capítulo 5

Arquitectura AIoT para la creación de servicios cognitivos en el ecosistema FIWARE

5.1. Introducción

El objetivo principal de este Capítulo es la instanciación y validación de la arquitectura de referencia para la creación de servicios cognitivos en el ecosistema FIWARE.

Esta segunda instanciación define un componente cognitivo responsable de habilitar la integración de servicios cognitivos basados en IA en sistemas IoT heredados FIWARE. Con el fin de habilitar la creación de servicios cognitivos, se define e implementa un componente cognitivo que sirve como capa de abstracción entre la plataforma FIWARE [58] y los servicios de IA. Así, la arquitectura permite desacoplar los procesos cognitivos del resto del sistema y de esa forma permite la adición de capacidades cognitivas a sistemas existentes.

La arquitectura se instancia y valida en el dominio del transporte marítimo. En concreto, el caso de uso presentado busca crear un servicio cognitivo para mejorar la gestión del tráfico marítimo. Para ello se genera la arquitectura COSIBAS y se integra en una solución IT comercial del sector. Se crea un servicio cognitivo capaz de calcular la hora de llegada de los buques a puerto. Esto permite mejorar los procesos gestionados por la solución, prolongando así

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

su vida útil. Una vez realizado el esfuerzo de integrar el componente cognitivo en el sistema, este permite integrar otros servicios cognitivos con un esfuerzo mucho menor.

En primer lugar, se presenta una segunda transformación de la arquitectura de referencia. A continuación, se describen los detalles del componente cognitivo. Posteriormente, se presenta el caso de uso de gestión de tráfico marino y la implementación de la arquitectura COSIBAS. Seguidamente, se introduce la validación de la arquitectura y se presentan y discuten los resultados obtenidos. Finalmente, el capítulo finaliza presentado las principales conclusiones.

5.2. Arquitectura

En este Capítulo se utiliza la arquitectura de referencia para actualizar y expandir la arquitectura FIWARE [58], con el objetivo de habilitar la creación de servicios cognitivos NGSI. La arquitectura presentada parte de la arquitectura IoT FIWARE y se le añaden un conjunto de componentes siguiendo las directrices de la arquitectura de referencia. Para asegurar una completa interoperabilidad y portabilidad con los componentes FIWARE, los nuevos componentes se definen utilizando el modelo de datos NGSI-V2 [174] de FIWARE. De esta forma se obtiene arquitectura AIoT compatible con todo el ecosistema FIWARE.

La arquitectura se diseña con dos objetivos principales. Por un lado, debe permitir la integración y agregación de fuentes de datos IoT heterogéneas, así como su análisis mediante servicios cognitivos. Estos servicios crean o mejoran soluciones específicas que capturan la lógica empresarial de la aplicación objetivo. Por otro lado, la arquitectura tiene en cuenta los dispositivos y servicios existentes en el sistema original, creando una capa cognitiva y haciendo uso de la semántica para apoyar las comunicaciones entre los componentes, sin forzar al sistema a adaptarse a ella si no al revés. Esto permite evitar, en la medida de lo posible, inversiones de tiempo de desarrollo extraordinarias, pues se minimizan los cambios necesarios para actualizar y mejorar un sistema que funcione bien, dotándolo de capacidades de análisis cognitivas.

Siguiendo el esquema utilizado por FIWARE, la Figura 5.1 muestra los diferentes componentes que forman parte de la arquitectura y sus comunicaciones entre ellos. El diagrama agrupa los diferentes componentes en colores para diferenciar los componentes genéricos (gris) de los específicos (naranja), así como los relativos al caso de uso (blanco) y el gestor de la información de contexto (verde).

- Componentes genéricos: Componentes *software* de código abierto que pueden ser ensamblados entre sí y con otros componentes de terceros

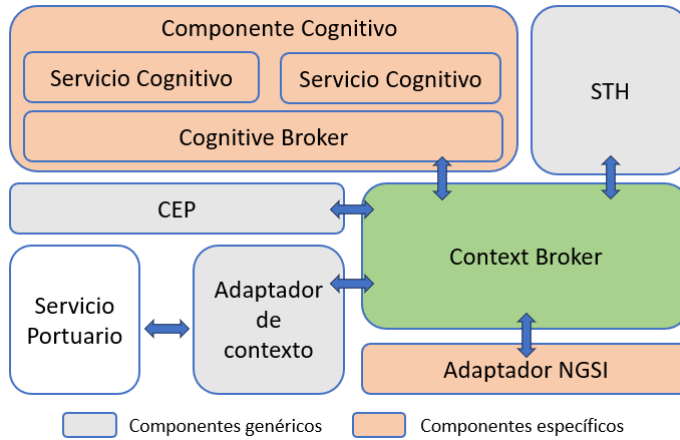


Figura 5.1: Vista global de la arquitectura.

para construir plataformas que soporten el desarrollo de soluciones inteligentes de forma más rápida, sencilla y económica.

- Componentes específicos: Componentes añadidos a la arquitectura FIWARE para habilitar la creación de servicios cognitivos nativos.
- Componente *Orion Context Broker* (OCB): Único componente necesario para que una solución sea considerada como una solución FIWARE.
- Componentes de servicios y *dashboard*: Componentes responsables de la creación de solicitudes y de la visualización de la información.

Los componentes que conforman la arquitectura van desde la solicitud y adquisición de datos y el procesamiento de estos hasta su representación en aplicaciones externas. Estos componentes se conocen como FIWARE *enablers* [192]. Todos los componentes y sus comunicaciones se describen con detalle en secciones posteriores.

La Figura 5.2 muestra los diferentes componentes que conforman la arquitectura AIoT para la creación de servicios cognitivos en el ecosistema FIWARE y las comunicaciones entre ellos.

Servicios externos

Los servicios suelen estar integrados dentro de plataformas extensibles e interoperables de forma que pueden interactuar con FIWARE a través de los

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

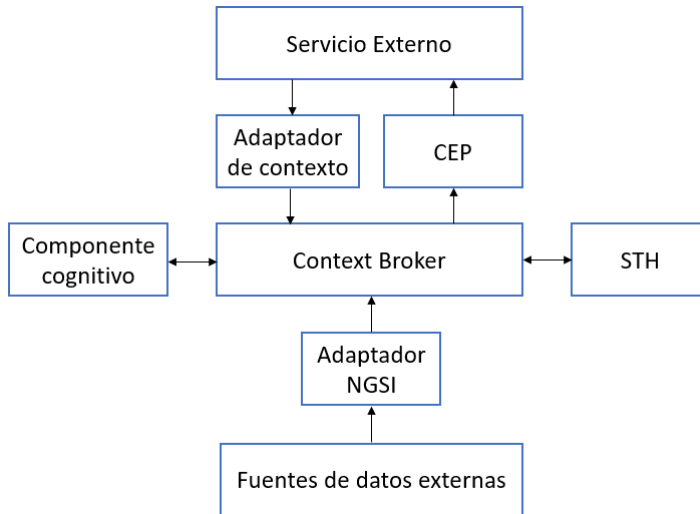


Figura 5.2: Arquitectura AIoT para la creación de servicios cognitivos en el ecosistema FIWARE.

adaptadores de contexto para enviar información a la plataforma FIWARE y con el adaptador de contexto para recibir información de esta. Algunos ejemplos de servicios son los servicios municipales ofrecidos por el gobierno a sus residentes. Estos servicios pueden ser muy diversos, como servicios de tramitación de documentación o servicios destinados a la eficiencia energética de las ciudades.

Adaptador de contexto

Los adaptadores de contexto (Figura 5.3) [193] se encargan de adaptar la comunicación entre los servicios existentes expuestos por terceros y el *context broker*. A través de los adaptadores de contexto, el *context broker* tiene acceso a la información proporcionada por los servicios externos.

Los adaptadores de contexto pueden actuar como proveedores de contexto (“poseedores” de información), productores de contexto (publicadores de información de contexto) o ambos. Un adaptador de contexto gestiona las solicitudes redirigidas (*updateContext*) y las solicitudes de notificación (*notify*) enviadas por el *context broker*, transformándolas en solicitudes al servicio externo correspondiente. Cuando se recibe una respuesta del servicio externo, el

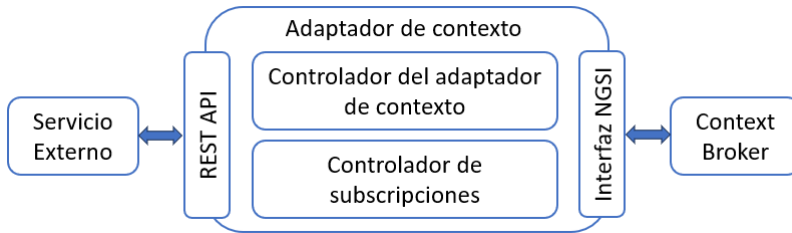


Figura 5.3: Adaptador de contexto.

adaptador de contexto actualiza la entidad correspondiente en FIWARE y el estado de la solicitud (si es necesario).

Los adaptadores de contexto se comunican con los servicios externos, ciudad, puerto, etc., a través de una API REST y con el *context broker* a través de una API REST NGSI-V2. Disponen de dos gestores, uno para gestionar la información recibida, transformándola en entidades de contexto NGSI-V2, cuando la información proviene de un servicio externo, y en entidades comprensibles por los servicios externos, cuando esta información proviene del *context broker*, y otro gestor para gestionar las suscripciones de contexto.

Adaptador NGSI

El adaptador NGSI (Figura 5.4) permite obtener información de proveedores externos a la plataforma a través de las API proporcionadas por ellos. Es un componente indispensable para el caso de uso marítimo, pues este requiere información de geolocalización e información meteorológica marina proporcionada por dos proveedores externos.

El adaptador se comunica con el *context broker* a través de una interfaz NGSI-V2. A través de esta, el adaptador obtiene y envía entidades de contexto desde/hacia el *context broker*. Por otro lado, a través de esta interfaz, el adaptador recibe notificaciones y envía suscripciones desde/hacia el *context broker*.

En su interior existe un *broker* encargado de ejecutar el agente adaptador correspondiente a la petición recibida y de gestionar las entidades de contexto gestionadas. Cada agente adaptador realiza la petición correspondiente al proveedor, y crea una entidad de contexto NGSI-V2 a partir de la información recibida.

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

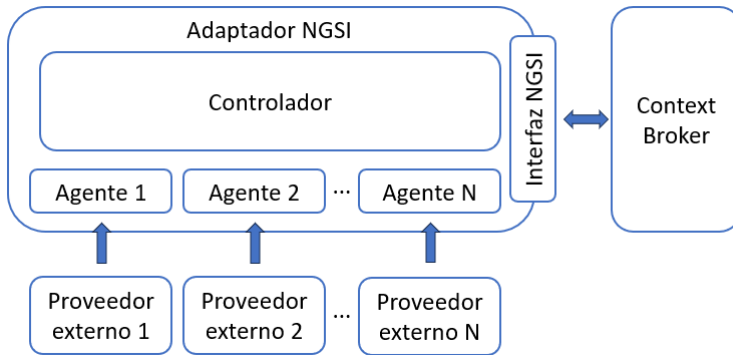


Figura 5.4: Adaptador NGSI.

Complex Event Processing

El CEP [194] tiene como misión principal apoyar el desarrollo, despliegue y mantenimiento de aplicaciones, analizando los datos de eventos en tiempo real y generando una visión inmediata. En particular, el CEP es útil para detectar patrones en los eventos (monitorización o correlación de eventos). El CEP puede recibir eventos de diferentes productores de eventos de la plataforma FIWARE. Estos productores de eventos incluyen sensores, otros habilitadores genéricos y servicios externos.

El CEP obtiene eventos utilizando dos mecanismos diferentes: los productores de eventos pueden insertar eventos en el CEP invocando operaciones estándar (modo *push*), o los productores de eventos pueden exportar operaciones estándar que el CEP puede invocar para recuperar eventos (modo *pull*). Además, los consumidores de contexto pueden conectarse al CEP para obtener los eventos de salida y reenviarlos al resto de aplicaciones suscritas o realizar acciones basadas en los eventos.

Context broker

Orion Context Broker [195] es el componente principal de FIWARE. Permite obtener información de contexto, eventos y otros datos de los productores de contexto y enviarlos a los consumidores de contexto. Este componente es como un controlador y un agregador de datos de contexto y, al mismo tiempo, es una interfaz entre los componentes de la arquitectura. Por lo tanto, los demás elementos de un sistema FIWARE pueden publicar o consumir datos sin tener un conocimiento específico del resto del sistema.

Este componente puede proporcionar datos a los consumidores de dos maneras, en modo de consulta (bajo petición) o cuando la información está disponible (bajo suscripción). El *context broker* es accesible a través de una interfaz NGSi REST.

Los productores de contexto publican elementos de contexto a través de la operación *updateContext* y los consumidores de contexto recuperan elementos de contexto a través de la operación *queryContext*, ambas operaciones sobre un *context broker*. Al mismo tiempo, por un lado, el *context broker* puede reenviar la consulta recibida de un consumidor de contexto al proveedor de contexto apropiado y devolver el resultado al consumidor de contexto solicitante y, por otro lado, el *context broker* puede reenviar una operación *updateContext* de un productor de contexto al proveedor de contexto apropiado. Otra forma de comunicarse con el *context broker* es a través de las operaciones de suscripción de contexto. Los consumidores de contexto suscritos pueden recibir elementos de contexto mediante una operación *subscribeContext* y el *context broker* puede enviar elementos de contexto a los proveedores de contexto mediante una operación *registerContext*. Además, las aplicaciones pueden suscribirse al registro de entidades y atributos que cumplen ciertas condiciones a través de una operación *subscribeContextAvailability*.

Short Time Historic Framework

Short Time Historic Framework (STH) [196] permite generar series temporales agregadas de información de contexto a partir de la evolución en el tiempo de los valores de los atributos de las entidades de contexto registradas en el *context broker*.

STH maneja cuatro conceptos principales:

- **Periodo de resolución o agregación:** Periodo de tiempo para el que se agrupa la información de la serie temporal agregada. Los valores posibles son: mes, día, hora, minuto y segundo.
- **Origen:** El origen del tiempo para el que se aplica la información de contexto de la serie temporal agregada. Por ejemplo, para una resolución de minutos, un valor de origen válido podría ser 2015-03-01T13:00:00.000Z. Nota: El origen se almacena utilizando la hora UTC para evitar problemas de configuración regional.
- **Desplazamiento:** Es el desplazamiento desde el origen al que se aplica la información de contexto de la serie temporal agregada. Por ejemplo, para una resolución de minutos y un origen 2015-03-01T13:00:00.000Z, un offset de 10 se refiere al décimo minuto de la hora concreta señalada por

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

el origen. En este ejemplo, habría un máximo de 60 desplazamientos de 0 a 59 correspondientes a cada uno de los 60 minutos de la hora concreta.

- Muestras: Es el número de muestras, valores, eventos o notificaciones disponibles para ese desplazamiento concreto desde el origen.

Componente cognitivo

El componente cognitivo ha sido diseñado siguiendo la filosofía FIWARE. Las comunicaciones internas dentro de los componentes siguen la interfaz estándar FIWARE NGSI-V2. Además, como en toda solución FIWARE, OCB gestiona la información de contexto.

Este componente es un habilitador cognitivo. Es decir, permite analizar los datos IoT obtenidos con técnicas de IA de forma transparente para la plataforma. El componente cognitivo recibe del *context broker* una petición para realizar un análisis cognitivo, reenvía la petición al servicio cognitivo solicitado e inserta el resultado del procesamiento en el *context broker*. De esta forma se sigue manteniendo toda la información de contexto en una única fuente.

El componente cognitivo consta de los siguientes elementos:

- Interfaz NGSI REST: A través de esta interfaz, el componente puede comunicarse con otros elementos de FIWARE, como el *context broker*.
- *Cognitive broker*: Recibe los datos de entrada, los reenvía a los servicios cognitivos, y envía los resultados del procesamiento al componente adecuado. El *broker* separa el procesamiento realizado por los servicios cognitivos, tanto de las fuentes de datos como del receptor, de los resultados.
- Servicios cognitivos: Son un conjunto de servicios que contienen algoritmos de ML que analizan los datos en tiempo real creando predicciones. El análisis que realizan los servicios cognitivos no es determinista, sino que se basa en hipótesis.

5.3. Componente cognitivo

Debido a que el diseño y la implementación de este componente es una de las principales contribuciones de la presente tesis, se ha creado esta Sección específica para indicar los detalles concretos de su diseño.

La Figura 5.5 muestra los elementos que forman el componente cognitivo, además de su interacción con el *context broker*. El componente cognitivo consta de dos elementos principales: el *cognitive broker* y los servicios cognitivos.

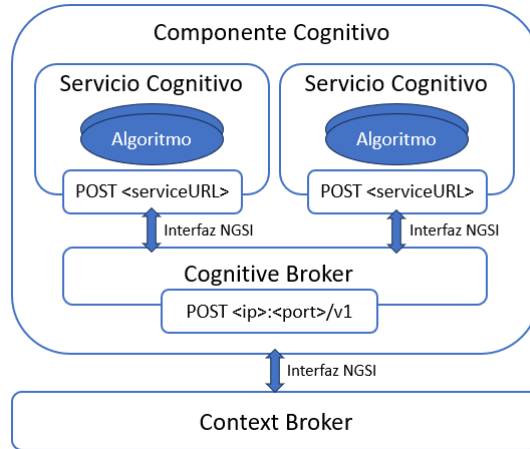


Figura 5.5: Componente cognitivo.

Mientras que los servicios cognitivos se encargan de realizar análisis cognitivos (empleando algoritmos de ML), el *cognitive broker* es responsable de las comunicaciones entre el *context broker* y los servicios cognitivos. En otras palabras, el *context broker* es un *middleware* que contiene la lógica de intercambio de información entre ambos componentes. Ambos elementos se comunican a través de interfaces NGSI-V2.

El *cognitive broker* permite:

- Gestionar las peticiones y notificaciones enviadas por el *context broker* a los servicios cognitivos.
El *cognitive broker* analiza las peticiones y notificaciones recibidas del *context broker* y envía peticiones a los servicios cognitivos solicitados.
- Registrar información de los servicios cognitivos.
Previo al uso de los servicios cognitivos, estos deben ser registrados en el *cognitive broker*. Además, permite eliminar servicios del registro o actualizar su información.
- Gestionar las suscripciones del servicio cognitivo en el *context broker*.
El *cognitive broker* proporciona métodos para consultar y crear nuevas suscripciones en el contexto por parte de los servicios cognitivos.
- Gestionar las entidades requeridas y resultantes por el servicio cognitivo.
El *cognitive broker* permite al servicio cognitivo obtener del contexto

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

aquellas entidades requeridas para el análisis cognitivo, e insertar en el contexto la entidad resultante de dicho análisis.

5.3.1. Flujo de mensajes

El intercambio de información entre los servicios cognitivos y el *context broker* comienza con el envío de una notificación por parte del *context broker* al *cognitive broker* y termina con el envío por parte del *cognitive broker* de una entidad de tipo *CognitiveResponse*, al *context broker*. Durante este proceso tienen lugar otros intercambios de información intermedios.

Antes de detallar el flujo de mensajes completo, es importante presentar las formas en que OCB puede operar. El *context broker* proporciona dos modos para cambiar el estado de la plataforma: síncrono y asíncrono. El método síncrono se produce cuando un usuario o aplicación realiza un cambio en el estado de la plataforma y la plataforma le informa del resultado. El método asíncrono consiste en que una aplicación se suscribe a los cambios en la información de contexto para que se le informe cuando se produzca dicho cambio. Este último método permite reducir el volumen de peticiones y, en consecuencia, la cantidad de datos que se transmiten entre los componentes de la plataforma, ya que éstos no necesitan sondear el estado de la plataforma cada cierto tiempo.

Cualquier suscripción de contexto está vinculada a una entidad de contexto, y también puede estar vinculada a ciertos atributos de esa entidad. Cuando el *context broker* actualiza el contexto, busca si hay una suscripción vinculada a la entidad que ha modificado el contexto. Si la hay, comprueba las condiciones establecidas en la suscripción encontrada, y si estas condiciones se cumplen, envía una notificación al componente definido en la propia suscripción encontrada. El contenido de esta notificación depende de las reglas definidas en la suscripción.

Respetando la filosofía de FIWARE, las comunicaciones con el *cognitive broker* pueden iniciarse de forma síncrona o asíncrona. En el caso de comunicación asíncrona, la operación empieza cuando el *context broker* recibe una entidad que tiene una suscripción activa cuya URL de notificación apunta al *endpoint* principal del *cognitive broker*. Cuando esto ocurre, el *context broker* busca en el contexto la suscripción vinculada a la entidad y envía una notificación al *cognitive broker*. El *cognitive broker* extrae la entidad *CognitiveRequest* del atributo de datos de la notificación, y la reenvía al servicio cognitivo correspondiente (previamente registrado).

El servicio cognitivo realiza sus predicciones utilizando un modelo de ML previamente entrenado con ciertas variables independientes. El valor de estas variables se registran en tiempo real como entidades en el *context broker* utilizando los mecanismos propios de la arquitectura FIWARE (ver Sección 5.2).

Así, para que el servicio cognitivo pueda realizar el análisis cognitivo, necesita recuperar el valor actual de estas entidades. Los identificadores de las entidades donde está almacenada la información vienen especificados en la entidad de tipo *CognitiveRequest*. Para obtener esta información, el servicio cognitivo realiza una petición de lectura al *cognitive broker* por cada identificador y este la redirige al *context broker*. El resultado del *context broker* se devuelve al *cognitive broker* que, a su vez, lo reenvía al servicio cognitivo.

Después, el servicio cognitivo alimenta el modelo y obtiene una predicción. El resultado se envía al *cognitive broker* con la creación y envío de una entidad de tipo *CognitiveResponse*. Antes de enviar la entidad, el servicio cognitivo comprueba si existe una suscripción asociada en el contexto. Si no existe, el servicio cognitivo crea una nueva suscripción para este tipo de entidad a través del *cognitive broker*. Finalmente, el servicio cognitivo envía la entidad de tipo *CognitiveResponse* al *context broker*.

La Figura 5.6 muestra el flujo operativo del *cognitive broker* y del resto de componentes. En la Figura se pueden identificar los siguientes procesos:

1. Proceso de notificación.

El proceso se inicia cuando el *context broker* envía una notificación al *cognitive broker*. Este procesa la notificación recibida y crea una entidad de tipo *CognitiveRequest* a partir de la información de los atributos de la entidad que ha sido actualizada.

2. Enviar *CognitiveRequest* a un servicio cognitivo.

El *cognitive broker* reenvía la entidad *CognitiveRequest* al servicio cognitivo indicado en el atributo *CognitiveService* de esta entidad.

3. Obtención de entidades del contexto.

Para realizar el análisis cognitivo, el servicio cognitivo necesita la información almacenada en el contexto. Esta información se le proporciona a través del atributo *entities* de la entidad *CognitiveRequest* recibida. El atributo *entities* contiene una lista de identificadores de entidades, por lo que, para cada identificador, el servicio cognitivo realiza una petición al *cognitive broker* y este, a su vez, la redirige al *context broker*. El resultado es devuelto por el *cognitive broker* al servicio cognitivo para su análisis.

4. Análisis cognitivo.

El servicio cognitivo realiza la inferencia del modelo almacenado con los datos obtenidos previamente y almacena el resultado en una entidad de tipo *CognitiveResponse*.

5. Obtener la suscripción.

Antes de que el servicio cognitivo envíe una entidad de tipo *CognitiveRes-*

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

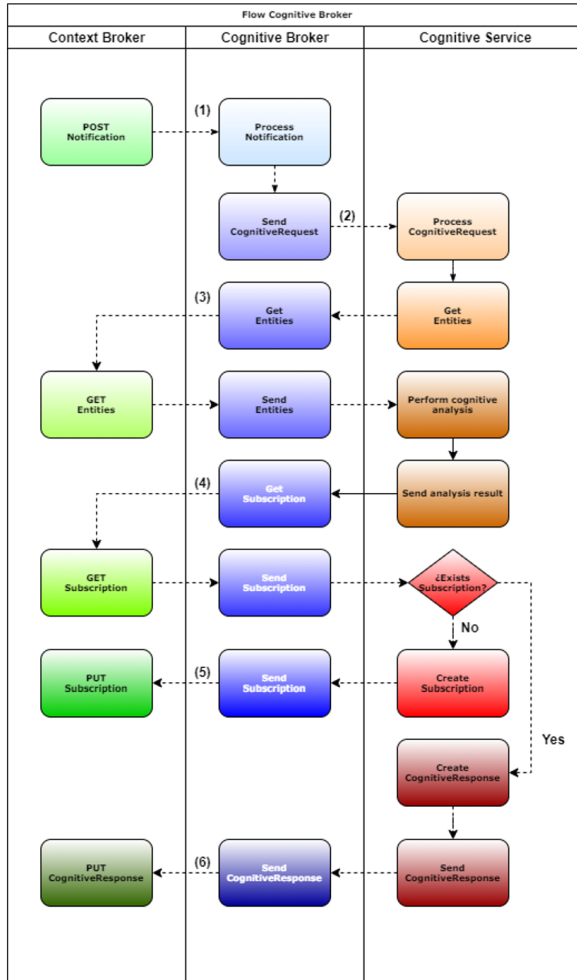


Figura 5.6: Flujo del componente cognitivo.

ponse al *context broker*, el servicio comprueba si existe una suscripción para este tipo de entidad en el contexto.

6. Crear una suscripción.

Si no se ha encontrado una suscripción en el contexto vinculado a la creación del tipo de entidad *CognitiveResponse*, el servicio cognitivo solicita la creación de una nueva suscripción para la entidad mencionada.

7. Enviar una *CognitiveResponse*.

Una vez gestionadas las suscripciones, el servicio cognitivo envía la entidad *CognitiveResponse* al *context broker*.

5.3.2. Interfaces NGSI REST

El *cognitive broker* proporciona una API que permite el intercambio de información entre el *context broker* y los servicios cognitivos, desacoplando estos componentes en una arquitectura distribuida.

La API proporciona métodos para gestionar las entidades de contexto, las suscripciones y las notificaciones:

- Gestión de entidades.

Los métodos responsables de la gestión de entidades de contexto permiten reenviar al *cognitive broker* las solicitudes de inserción y lectura de los servicios cognitivos para crear y obtener entidades de contexto.

- Gestión de suscripciones.

Los métodos responsables de la gestión de las suscripciones permiten reenviar al *cognitive broker* las solicitudes de los servicios cognitivos para crear y listar suscripciones de contexto.

- Gestión de notificaciones.

Los métodos responsables de la gestión de las notificaciones permiten al *cognitive broker* recibir notificaciones y extraer y reenviar la entidad de tipo *CognitiveRequest* contenida en ellas.

Las entidades de contexto específicas del componente cognitivo son el *CognitiveRequest* y el *CognitiveResponse*. Las entidades de tipo *CognitiveRequest* siguen la siguiente estructura:

```
{
  "id": "<CognitiveRequestId>",
  "type": "CognitiveRequest",
  "entities": {
    "type": "StructuredValue",
```

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

```
"value": [<entity_id1>, ... <entity_idn>],
"metadata": {
  "description": {
    "type": "text",
    "value": "CognitiveRequest entities"
  }
},
},
"cognitiveService": {
  "type": "text",
  "value": "<CognitiveServiceName>",
  "metadata": {
    "description": {
      "type": "text",
      "value": "<CognitiveService description>"
    }
  }
}
}
```

En esta entidad se observan los dos atributos esenciales que se utilizan para el análisis cognitivo:

- *cognitiveService*: Contiene la clave del servicio cognitivo a ejecutar. A partir de esta clave, obtiene de una base de datos, la URL correspondiente al servicio cognitivo.
- *entities*: Contiene una lista de identificadores de entidades necesarios para el análisis cognitivo.

Los servicios cognitivos requieren entidades almacenadas en el contexto de la plataforma IoT. En este punto es donde se produce la confluencia entre IoT y la IA. El atributo *entities* incluye los identificadores de las entidades necesarias para realizar las predicciones.

Por otro lado, las entidades de tipo *CognitiveResponse* siguen la siguiente estructura:

```
{
  "id": "<CognitiveResponseId>",
  "type": "CognitiveRequest",
  "correlationId": {
    "type": "text",
    "value": "<CognitiveRequestId>",
    "metadata": {
      "description": {
```



```
    "type": "text",
    "value": "Correlation id"
  }
},
"result": {
  "type": "StructuredValue",
  "value": [
    {
      "key": "<predicted attribute>",
      "value": "<CognitiveService prediction>",
      "metadata": {
        "description": {
          "type": "text",
          "value": "<attribute units>"
        }
      }
    }
  ],
  "metadata": {
    "description": {
      "type": "text",
      "value": "<CognitiveService result description>"
    }
  }
}
```

Estas entidades tienen un atributo *correlationId* que contiene el *id* de la *CognitiveRequest* con la que se corresponde. El atributo *result* es un vector que contiene las N predicciones proporcionadas por el servicio cognitivo.

5.4. Caso de uso: gestión del tráfico marítimo

La IA está permitiendo a la industria marítima no solo adelantarse al mercado, sino acelerar su transformación digital y afrontar los retos de la próxima transición energética. Está revolucionando las operaciones marítimas, creando importantes ventajas competitivas para las empresas que la emplean. No obstante, la adopción de la IA en el sector marítimo está aún en una fase incipiente, y su velocidad de desarrollo dependerá de la infraestructura informática y las soluciones de conectividad disponibles. Cuando la integración de la IA en los sistemas marítimos se implemente de forma adecuada y eficiente, proporcionará un gran valor. La industria marítima está construida sobre siglos de experiencia

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

humana. La combinación de esta experiencia y la IA permite procesar con gran velocidad un mayor volumen de información y, en última instancia, alcanzar nuevos niveles de optimización. Se estima que los análisis de datos tradicionales y heredados solo tienen en cuenta el 10 % de los datos disponibles, mientras que modelos de IA pueden tener en cuenta cerca del 100 % de los datos y procesarlos instantáneamente para crear una visión extremadamente precisa del problema bajo estudio.

Dentro de las actividades portuarias, la incertidumbre sobre la hora de llegada de un buque a puerto es una problemática destacada. En la actualidad, un operador marítimo calcula la hora estimada de llegada (en inglés *Estimated Time of Arrival* (ETA)) de los buques utilizando fuentes de información limitadas y una técnica de estimación muy básica [197]. Además, la ETA se calcula cuando se crea la escala y no se actualiza con frecuencia. Este dato suele diferir considerablemente de hora actual de llegada (en inglés *Actual Time of Arrival* (ATA)), sobre todo para horizontes temporales largos, convirtiéndose así en una estimación poco fiable. Las desviaciones son debidas principalmente a los contratiempos que pueden surgir durante la travesía, como el cambio en la velocidad del buque, las condiciones meteorológicas, las malas estimaciones de las autoridades portuarias, el retraso del puerto de origen, etc., que no se tienen en cuenta en el cálculo. Además, la técnica de estimación se basa en una aritmética muy básica y simple, y no puede proporcionar una gran precisión, ni incluir otros factores relevantes más allá del tiempo de viaje inicial y la distancia al puerto. Estas desviaciones con respecto a la hora de llegada real dificultan las actividades de planificación de las partes implicadas en el transporte de contenedores, ya que la llegada del buque es el punto de partida del transporte interior de mercancías. Por lo tanto, la desviación entre la ETA y la ATA implica un elevado coste económico en la medida en que provoca tener recursos en espera o no disponer de los recursos necesarios en el momento requerido. Una mejor estimación proporcionaría inmensos beneficios en términos de costes para los agentes portuarios, una importante optimización de tiempos y una mejor eficiencia de las operaciones portuarias.

El objetivo que persigue este caso de uso es la integración un servicio cognitivo que permita calcular la ETA de buques portacontenedores en el sistema de gestión portuario Posidonia PortCDM [198] de la empresa Prodevelop. Así, este caso de uso tiene dos objetivos principales: i) añadir en el sistema los mecanismos necesarios que permitan una sencilla integración de servicios cognitivos y ii) integrar un primer servicio cognitivo que permita predecir de forma precisa la ETA de los buques. Para ello, se busca agregar nuevas fuentes de datos y analizarlas en tiempo real mediante algoritmos de IA y ML. El resultado de este demostrador permite optimizar el tiempo, los recursos y el coste de las operaciones marítimas.

5.4 Caso de uso: gestión del tráfico marítimo

Los buques se registran en el sistema cuando estos están a 10-15 millas del puerto. Por lo tanto, el tipo de ETA a estimar es de corta distancia. La algoritmia aplicada se basa en datos heterogéneos recogidos que miden factores críticos que determinan o tienen influencia relevante en el tiempo de llegada de los buques y, en consecuencia, en el cálculo y predicción de la ETA. Los trabajos que estudian la problemática del cálculo de las ETA de los buques [199], [200], [201], [202], [203], [204] utilizan datos heterogéneos que miden factores críticos que determinan o tienen influencia relevante en el tiempo de llegada de los buques. Estos señalan que la posición del buque, su velocidad o la meteorología (entre otros) son factores críticos a la hora de realizar estas predicciones. Tras un análisis detallado de los trabajos mencionados se seleccionan los atributos a emplear durante el entrenamiento de los algoritmos: datos del Sistema de Identificación Automática (*Automatic Identification System* (AIS)) de las embarcaciones e información oceanográfica y meteorológica. Se busca integrar y analizar estas fuentes de datos en tiempo real.

De acuerdo con las disposiciones del *Safety Of Life At Sea* (SOLAS) y a los convenios aprobados por la Organización Marítima Internacional (OMI), AIS es un sistema internacional obligatorio de comunicaciones para la seguridad de la navegación. Las señales AIS pueden ser detectadas por un satélite en una órbita terrestre baja y proporcionan una capacidad global para supervisar todos los buques equipados con antenas y sistemas AIS utilizando una constelación de satélites y una amplia red de estaciones terrestres. Además, AIS proporciona información relevante sobre la embarcación, como el *Maritime Mobile Service Identity* (MMSI), el nombre de la embarcación, la posición actual y el destino, el tiempo estimado de llegada, el rumbo, la velocidad, la OMI, etc. La recopilación y gestión de los datos generados por AIS la realiza AISHub [205], una plataforma de intercambio de datos que comprende una red de receptores AIS (en su mayoría proporcionados por aficionados). Los mensajes AIS se transmiten en el protocolo *National Marine Electronics Association* (NMEA) 0183 [206]. AISHub permite recuperar estos mensajes a través de su API. Además, esta plataforma también permite obtener información en línea sobre buques y estaciones AIS a través de una interfaz gráfica y una API.

Otra fuente de información a utilizar en este caso de uso es la proporcionada por fuentes abiertas de información meteorológica. Se proporciona información oceanográfica y meteorológica en tiempo real sobre la altura, el periodo y la dirección de las olas, la velocidad y la dirección del viento, la temperatura del aire, los perfiles de las corrientes a diferentes profundidades, etc.

Estas fuentes de datos son gestionadas por la arquitectura *Cognitive services for IoT-based scenarios* (COSIBAS) (véase la Sección 5.2) para alimentar los servicios cognitivos y entrenar los algoritmos de predicción presentes en ellos. Con el fin de validar la arquitectura propuesta en un entorno real, este caso

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

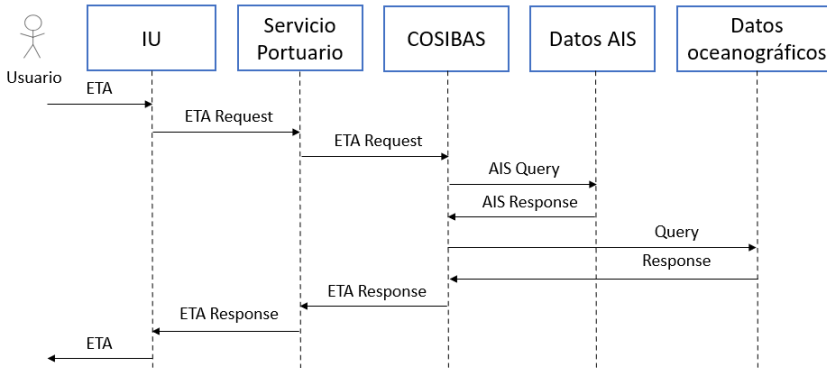


Figura 5.7: Flujo de datos del demostrador.

de uso se implementa y valida en los puertos de Valencia y Algeciras, donde la empresa ya tiene el sistema desplegado. El caso de uso ha sido desarrollado en el contexto del proyecto CDTI COSIBAS [207].

El flujo de datos del demostrador queda reflejado en la Figura 5.7. Los pasos son los siguientes:

1. El usuario selecciona el buque del que desea conocer la ETA.
2. La aplicación cliente solicita la ETA a los servicios portuarios.
3. Los servicios portuarios solicitan la ETA a la arquitectura COSIBAS.
4. COSIBAS solicita los datos de entrada necesarios a las fuentes de datos. Los modelos predictivos son proporcionados por el *cognitive broker* y la comunicación con las fuentes de datos es gestionada por el *context broker*. El *context broker* obtiene los datos de las diferentes fuentes y los envía al *cognitive broker*.
5. COSIBAS ejecuta el modelo para calcular la ETA y devuelve el resultado a los servicios portuarios. El resultado se devuelve a través del CEP y este lo redirige al servicio portuario adecuado.
6. Los servicios portuarios envían el resultado a la aplicación cliente.
7. El personal del puerto puede ver en tiempo real cómo evoluciona la escala durante la estancia en el puerto gracias a la interfaz de usuario.

5.5. Implementación

A partir de la arquitectura presentada en la Sección 5.2 se define e implementa la arquitectura COSIBAS.

5.5.1. Relación con la arquitectura de referencia

En la presente Sección se describe la relación de la arquitectura COSIBAS con la arquitectura global. La Figura 5.8 muestra los bloques seleccionados de la arquitectura de referencia para la creación de la arquitectura COSIBAS. Esta arquitectura no implementa los niveles de seguridad y privacidad de la arquitectura de referencia.

Empezando por abajo, y a diferencia de las arquitecturas previas, el nivel de percepción está formada por fuentes de datos externas en lugar de sensores IoT. Este nivel incluye la integración con sistemas de información externos a través de API REST o la integración con bases de datos externas, entre otras. En esta arquitectura resulta preciso monitorizar atributos tales como la ubicación de los buques, sus características o el estado del mar. Con ello se busca obtener una visión precisa del estado del buque y realizar predicciones basadas en esa información. En lugar de incluir nuevos sensores, se integra la información necesaria a través de fuentes de datos ya existentes. Esto permite ahorrar tiempo y costes de desarrollo.

El nivel de *middleware* está formado por un *context broker*. Este se encarga de recopilar y almacenar el contexto del sistema IoT, y luego proporcionar acceso a ese contexto a otros componentes de la arquitectura. Esto permite a estos componentes tomar decisiones y realizar acciones basadas en el contexto del sistema.

A continuación, se encuentra el nivel cognitivo de servicios, compuesto por el componente cognitivo, el CEP y el adaptador de contexto. El componente cognitivo proporciona funcionalidades de procesamiento cognitivo y análisis de datos. Utiliza técnicas de ML para detectar patrones y tendencias en los datos recopilados por la red IoT y tomar decisiones basadas en esa información. El CEP permite procesar y analizar los datos recopilados por la red IoT en tiempo real para analizar y procesar los datos recopilados, detectando patrones y relaciones complejas entre ellos. El adaptador de contexto facilita la integración entre el sistema y la interfaz de usuario, ya que se utilizan distintos formatos y estándares para representar y compartir el contexto. El adaptador de contexto actúa como un puente entre el sistema heredado y la arquitectura COSIBAS, permitiendo que compartan y utilicen el contexto de manera uniforme.

El nivel de datos semánticos se sitúa de manera transversal entre el nivel de *middleware* y el nivel cognitivo de servicios. El adaptador semántico propuesto

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

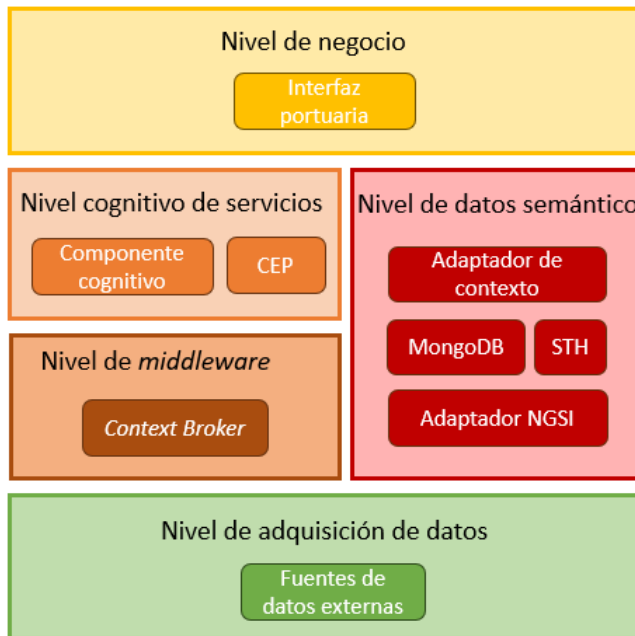


Figura 5.8: Arquitectura COSIBAS - Relación con la arquitectura de referencia.

en la arquitectura de referencia se ha implementado en forma de dos adaptadores en COSIBAS. Cada adaptador es responsable de adaptar la información compartida con las capas inmediatamente superior e inferior. El adaptador NGSi actúa como un puente entre el sistema IoT y el *context broker*, permitiendo que el sistema IoT envíe y reciba contexto del *context broker* a través de la API NGSi-V2. Esto permite que el sistema IoT se integre con otros sistemas y aplicaciones que utilizan el estándar NGSi-V2 para el intercambio de contexto. Por otro lado, el adaptador de contexto adapta la comunicación entre el servicio portuario y el *context broker*. Finalmente, los datos de contexto del *context broker* se almacenan en la base de datos MongoDB, mientras que los datos históricos se guardan en STH.

Por último, el nivel de negocio incluye la interfaz de usuario del servicio portuario. Esta permite a los trabajadores del puerto calcular la hora de llegada a puerto de un buque portacontenedores gracias al servicio cognitivo asociado.

5.5.2. Arquitectura COSIBAS

La integración del servicio cognitivo en Posidonia Port CDM se realiza a través de la arquitectura COSIBAS (Figura 5.9). Esta arquitectura se ha diseñado e implementado a partir de la arquitectura anteriormente presentada para dar respuesta al caso de uso de la gestión del tráfico marítimo.

El usuario selecciona un buque en la interfaz gráfica y solicita su ETA. El servicio portuario realiza esta petición al servicio cognitivo al *context broker* a través del adaptador de contexto. A continuación, a través del adaptador NGSi el *context broker* se añade en el contexto tanto información AIS del buque seleccionado como la información meteorológica de las inmediaciones del buque. Una vez esta información está disponible, el *context broker* realiza la petición cognitiva al componente cognitivo. Este recupera la información AIS y meteorológica del contexto, realiza la predicción e inserta el resultado en el contexto. Por último, a través del CEP se envía una notificación al servicio portuario con la ETA del buque.

La Figura 5.10 resume el intercambio completo de mensajes del sistema. Cuando un usuario solicita la ETA de un determinado buque, el adaptador de contexto transforma la petición en una entidad NGSi-V2 de tipo *VesselETARequest* (véase el Apéndice A). Antes de enviar la entidad al *context broker*, revisa si hay una suscripción asociada a ese tipo de entidad. Si no existe la crea. La suscripción debe estar configurada para notificar al adaptador de contexto cuando se crea una entidad de este tipo.

Cuando el *context broker* recibe la notificación de tipo *VesselETARequest*, la reenvía al *broker* del adaptador NGSi y este notifica a los agentes NGSi. El agente NGSi AIS usa el identificador del buque, MMSI, para hacer una peti-

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

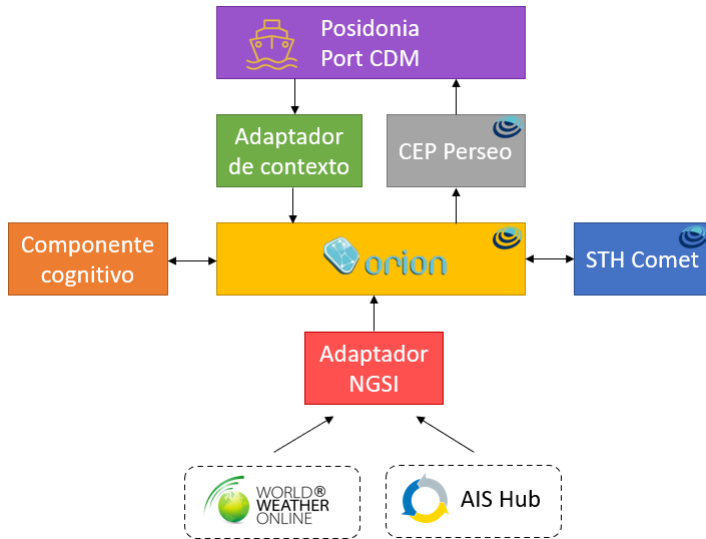


Figura 5.9: Arquitectura COSIBAS.

ción a la API de AISHub y obtener su información AIS. La información estática del buque (MMSI, nombre, bandera, tipo de buque, entre otros) se incluye en una entidad NGSI-V2 de tipo *Vessel* (véase el Apéndice A), mientras que la información dinámica del buque (ubicación, rumbo, velocidad, ETA, etc.) se incluye en una entidad *VesselLocation* (véase el Apéndice A). Si el buque ya estaba registrado como entidad NGSI-V2 en el contexto, se actualiza en lugar de crear una nueva. En paralelo, el agente NGSI meteorológico obtiene la información meteorológica marina correspondiente a la geolocalización del buque y registra esta información en el contexto con una entidad de tipo *Weather* (véase el Apéndice A). Los agentes NGSI crean las entidades, pero es el *broker* del adaptador NGSI quien las registra en el contexto. Seguidamente, el *broker* del adaptador NGSI crea una entidad adicional de tipo *CognitiveRequest* (véase el Apéndice A). Esta entidad incluye un campo con el nombre del servicio cognitivo requerido, además de proporcionar los identificadores de las entidades *Vessel*, *VesselLocation* y *Weather* que el servicio recuperará para hacer la predicción. Antes de insertar la entidad *CognitiveRequest*, se comprueba si existe una suscripción para este tipo de entidad. Si no existe, se crea. El objetivo de esta suscripción es notificar al *cognitive broker* cada vez que se crea una entidad de este tipo.

5.5 Implementación

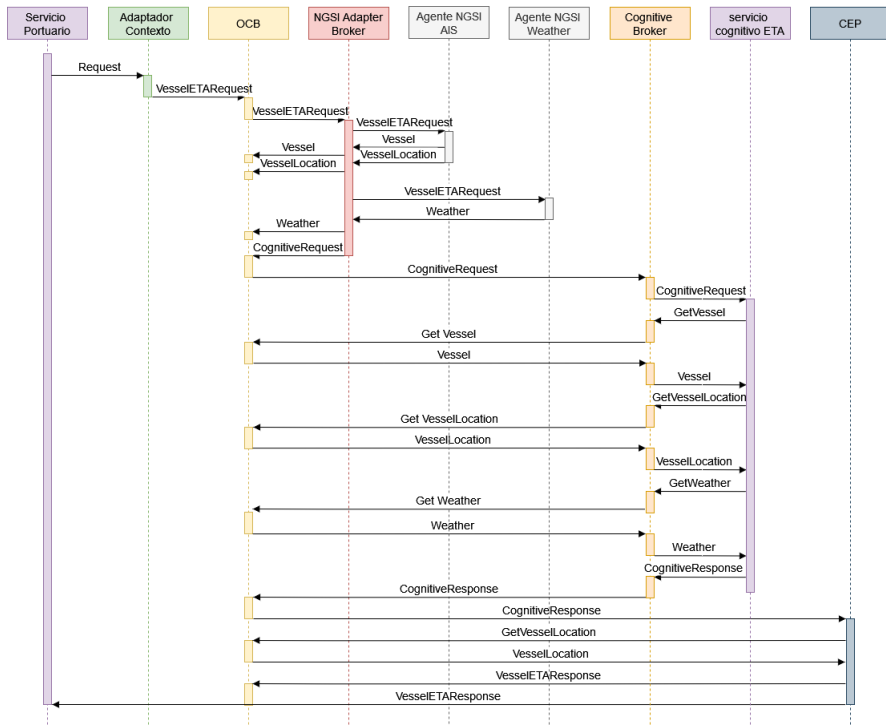


Figura 5.10: Flujo de mensajes.

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

Cuando el *cognitive broker* recibe una notificación, la examina, extrae la entidad que contiene y la envía al servicio cognitivo seleccionado a través de su API. Dado que la información asociada a las entidades *Vessel*, *VesselLocation* y *Weather* está almacenada en el contexto, el servicio realiza una petición a través del *cognitive broker*, para obtener la información de las entidades. Posteriormente, el servicio cognitivo extrae de las entidades los atributos que el modelo necesita y realiza la predicción. El último paso consiste en registrar la ETA predicha en el contexto. El servicio inserta el resultado en una entidad NGS-V2, la *CognitiveResponse* (véase el Apéndice A), enviándola al *cognitive broker*. Antes de insertar la entidad en OCB, el *cognitive broker* comprueba que existe una suscripción de tipo *CognitiveResponse* que notifica al CEP. Si no existe, crea esta suscripción.

Cuando el CEP recibe una notificación, compara la ETA predicha por el servicio cognitivo con la ETA planificada proporcionada en los datos AIS. Primero realiza una petición al *context broker* para obtener la entidad *VesselLocation* del buque asociado. En esta entidad está contenida la ETA planificada. La ETA predicha está contenida en la notificación recibida, por lo que en ese momento ya puede comparar ambos valores. En función de la diferencia, se crea una respuesta, cuyo tipo puede ser informativo o de advertencia con una descripción. Esta respuesta se empaqueta en una entidad de tipo *VesselETAResponse* (véase el Apéndice A) que, por un lado, se registra en el contexto y, por otro, se envía al servicio portuario.

Servicios cognitivos

El caso de uso marítimo busca agregar un servicio cognitivo capaz de predecir con exactitud la ETA de los buques portacontenedores a puerto destino. El caso de uso se aplica en dos puertos distintos: el puerto de Valencia y el puerto de Algeciras.

Los servicios cognitivos incorporan modelos entrenados de ML que proporcionan la estimación de la ETA. Para obtener una buena precisión, se necesita entrenar un modelo para cada puerto, pues cada uno tiene unas características únicas. De esta forma, cada algoritmo aprende únicamente los patrones de las rutas marítimas que rodean al puerto destino. Los servicios cognitivos se implementan como microservicios accesibles a través de una API REST. Antes de realizar una solicitud cognitiva, los servicios deben ser registrados en el componente cognitivo. Los servicios se han implementado utilizando Python 3 [208] y los modelos predictivos se han integrado dentro de ellos mediante su serialización utilizando la librería Pickle [209].

Los algoritmos se entrenan utilizando dos fuentes de datos: datos AIS y meteorológicos. La información AIS puede dividirse en dos categorías. La in-

Tabla 5.1: Datos AIS.

Información dinámica	Información estática
Estado de navegación	MMSI
Velocidad de giro	IMO
Speed Over Ground	Call sign
Longitud	Nombre del buque
Latitud	Tipo de buque y carga
Rumbo	Dimensiones
Timestamp	Puerto destino
Course Over Ground	ETA

Tabla 5.2: Datos meteorológicos.

Información oceanográfica
Temperatura del agua
Velocidad del viento
Dirección del viento
Precipitaciones
Visibilidad
Presión atmosférica
Altura del oleaje
Dirección del oleaje
Periodo del oleaje
Altura de la marea
Tipo de marea

formación dinámica, como la velocidad y la información de posición, captada a partir de los sensores internos instalados a bordo del buque y que se transmite cada 2-10 segundos mientras está en marcha y cada tres minutos mientras está fondeado, y la información estática, proporcionada a partir de la instalación de la unidad AIS (por ejemplo, MMSI, nombre del buque, tamaño del buque) o introducida manualmente por la tripulación para cada viaje (por ejemplo, puerto de destino, hora prevista de llegada, calado), que se transmite cada seis minutos. Algunos de los campos más informativos transmitidos en los mensajes AIS se presentan en la siguiente Tabla 5.1. Por otro lado, la Tabla 5.2 incluye los atributos del histórico de datos meteorológicos.

Como paso previo al entrenamiento de los modelos, los datos fueron pre-procesados. El procesado de los datos, su entrenamiento y validación se hizo

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

utilizando el paquete de aprendizaje automático *Scikit-learn* [118] y las librerías *Pandas* [210], *Numpy* [211] y *Matplotlib* [212].

Durante el análisis exploratorio se eliminaron todos los registros que contenían datos no disponibles o incorrectos. Además, fue necesario normalizar el valor de algunos atributos AIS. Los datos AIS no proporcionan el ATA del buque (variable dependiente), valor indispensable para entrenar el algoritmo. La ATA de los buques se ha calculado almacenando la marca de tiempo de los buques cuando cruzan una determinada zona del puerto destino. Cuando llegan a este punto se considera que han llegado al puerto. Es importante tener en cuenta que las características irrelevantes y redundantes pueden confundir a un algoritmo de aprendizaje al dificultar la distribución del pequeño conjunto de características verdaderamente relevantes [213]. Por lo tanto, se ha analizado qué características proporcionan mayor información para la predicción y solo estas se han incluido en el modelo.

Los datos se dividieron en un *dataset* de entrenamiento y otro de validación. Este paso no se hizo al azar, sino que cada *dataset* incluía rutas completas de buques. Entrenar y validar los algoritmos con una misma ruta pondría en entredicho la integridad de la validación, ya que los algoritmos podrían aprender específicamente los patrones de la ruta utilizada en lugar de generalizar de manera efectiva para nuevos datos.

Por último, para predecir los minutos que le quedan al buque para llegar a su destino, se crearon modelos de predicción mediante el entrenamiento de algoritmos de regresión *Decision Tree*, *Support Vector Machine*, *Random Forest* y *K-Nearest-Neighbor*.

El *dataset* de Valencia está formado por dos *datasets* (datos AIS y datos meteorológicos). El primer *dataset* consta de 700 614 registros de datos AIS procedentes de un receptor AIS [214] ubicado cerca del puerto de Valencia. Incluye información de las rutas seguidas por 271 buques para llegar al puerto en un rango de tiempo de un mes. El segundo *dataset* incluye información meteorológica marina correspondiente a la geolocalización de los buques durante este periodo. El *dataset* se construye utilizando el proveedor *World Weather Online* [215] e incluye 1 548 registros.

Para obtener el *dataset* de Algeciras se utilizó otro receptor AIS para construir un histórico de las tramas transmitidas por todos los buques portacontenedores que estuvieron dentro del alcance del receptor en un período temporal de seis meses. El *dataset* resultante incluye 3 761 994 mensajes AIS correspondientes a las rutas realizadas por 6 618 buques portacontenedores. El *dataset* anterior fue complementado con información meteorológica correspondiente a la geolocalización de los buques durante este periodo.

5.6. Validación

La validación del *framework* se hace en dos pasos. En primer lugar, se valida el componente cognitivo, y, en segundo lugar, los modelos integrados en los servicios cognitivos.

El desarrollo y validación del componente cognitivo se realiza utilizando la metodología *Test-Driven Development* (TDD) [216]. Esto garantiza que el componente se ajuste a los requisitos específicos y que cumpla con las expectativas previamente definidas. Durante el desarrollo del componente se llevaron a cabo tanto *tests* unitarios como de aceptación. Los *tests* unitarios se definen para asegurar la funcionalidad correcta de unidades individuales del código. Por otro lado, los *tests* de aceptación se enfocan en verificar que el componente completo cumpla con los criterios de aceptación y se integrara correctamente en el sistema.

Por otro lado, las métricas de validación utilizadas para validar los modelos son las siguientes: error absoluto medio (del inglés *Mean Absolute Error* (MAE)) Ecuación (5.1), error cuadrático medio (del inglés *Root Mean Squared Error* (RMSE)) Ecuación (5.2) y coeficiente de determinación (R^2) Ecuaciones (5.3) y (5.4).

$$MAE = \sum_{i=1}^n \frac{|\hat{y}_i - y_i|}{n} \quad (5.1)$$

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}} \quad (5.2)$$

donde \hat{y}_i son los valores reales e y_i los predichos de las n observaciones consideradas.

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (\hat{y}_i - \hat{y})^2} \quad (5.3)$$

donde \hat{y}_i representa el valor predicho de y_i e \hat{y} es la media de los datos observados que se calcula como

$$\hat{y} = \sum_{i=1}^n y_i \quad (5.4)$$

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

El MAE indica la diferencia absoluta media entre los valores previstos y los valores reales. Se trata de una métrica comúnmente utilizada para evaluar modelos de regresión. A diferencia de otras métricas, el MAE no eleva al cuadrado las diferencias entre predicciones y valores reales. Esto significa que no otorga un peso excesivo a los valores atípicos, lo que lo hace más robusto en escenarios donde los datos pueden contener observaciones inusuales. Puesto que esta métrica no penaliza los errores más grandes de manera proporcional, puede no ser sensible a pequeñas mejoras en el rendimiento del modelo. Para poder evaluar los errores grandes se utiliza la métrica RMSE.

RMSE proporciona la raíz cuadrada del promedio de diferencias cuadradas entre la predicción y la observación real. Es el criterio más importante para ajustar el modelo cuando los errores grandes son más problemáticos o costosos en el contexto específico del problema.

En cuando al coeficiente de determinación R^2 , este mide la cantidad de variación que puede ser explicada por los modelos. El coeficiente de determinación varía entre 0 y 1. Un valor de R^2 igual a 1 indica que el modelo explica toda la variabilidad en los datos, mientras que un valor de 0 indica que el modelo no explica nada. Los valores intermedios reflejan la proporción de variabilidad explicada.

5.7. Resultados y discusión

Esta Sección presenta los resultados obtenidos al incorporar en Posidonia Port CDM un servicio cognitivo capaz de calcular la ETA de buques portacontenedores utilizando servicios cognitivos basados en FIWARE. La incorporación de este nuevo servicio se ha realizado mediante la integración de la arquitectura COSIBAS en dos puertos diferentes. En primer lugar, en el puerto de Valencia y posteriormente en el puerto de Algeciras.

Tras la presentación y discusión de estos resultados se analizan los beneficios derivados del uso del componente cognitivo, y en general de la arquitectura COSIBAS, para la integración de servicios cognitivos en el sistema heredado.

5.7.1. Componente cognitivo

El caso de uso abordado tiene dos objetivos principales. Por un lado, agregar a un sistema heredado un servicio cognitivo capaz de predecir la ETA de los buques con precisión y, por otro lado, realizar esta integración de forma fluida y con los mínimos cambios posibles para el sistema. El componente cognitivo ha sido el elemento clave para lograr dichos objetivos.

La consecución del primer objetivo ha sido validada en el apartado anterior. Como se ha mostrado, los algoritmos desarrollados e integrados en el sistema son capaces de realizar estimaciones precisas del tiempo de llegada de los buques. Por otro lado, en el presente apartado se discute la consecución de la integración fluida y sencilla del servicio.

El esfuerzo inicial durante la primera integración (puerto de Valencia) del sistema Posidonia Port CDM con el resto de los componentes del sistema y, en especial, con el componente y el servicio cognitivo de la ETA fue significativo. Fue necesario realizar gran cantidad de *tests* de aceptación para asegurar una correcta integración entre componentes. Sin embargo, este esfuerzo se vio recompensado durante la segunda integración (puerto de Algeciras). Una vez se disponía de un sistema ya correctamente integrado, se necesitó poco tiempo para incorporar el segundo servicio cognitivo. De esa forma, el componente cognitivo permite reducir sensiblemente el esfuerzo técnico necesario para incorporar servicios cognitivos una vez se ha configurado correctamente la conexión entre los elementos del sistema. En este caso, el mayor esfuerzo estuvo dedicado al desarrollo de un nuevo modelo de ML (elemento clave del servicio cognitivo). Al tratarse de un puerto distinto, sus características cambian y no es posible reutilizar el modelo obtenido en el primer puerto.

Hasta donde el autor sabe, este trabajo representa el primer enfoque de código abierto que permite la creación de soluciones basadas en IA, garantizando el cumplimiento de las directrices y estándares de FIWARE.

El componente cognitivo permite una integración sencilla de futuros servicios en el sistema Posidonia Port CDM (ya sean servicios de ETA en otros puertos u otro tipo de servicios). Además, es importante destacar que, aunque durante este trabajo solo se ha desarrollado e integrado servicios cognitivos en una solución de gestión portuaria, el componente cognitivo está diseñado para incorporar servicios de cualquier dominio (salud, transporte, ciudades inteligentes, etc.) dentro del ecosistema FIWARE. Esto es posible gracias a que el componente cognitivo almacena la información de contexto en OCB empleando el modelo de datos NGSI-V2, el cual es compatible con múltiples verticales. Futuros trabajos podrían validar el componente propuesto en otros verticales. Para ello, los pasos principales serían: i) la incorporación de las fuentes de datos en el sistema (mediante el adaptador de contexto), ii) el desarrollo de los servicios cognitivos y iii) su correcta integración a través de las entidades NGSI-V2 descritas en el Apéndice A. Además, este componente facilita que los servicios creados evolucionen fácilmente, añadiendo nuevas fuentes de datos según sea necesario y con todo ello ahorrando un tiempo de desarrollo significativo.

Por otro lado, el componente cognitivo tiene un punto débil. Como ha sido desarrollado para utilizar el modelo de datos NGSI-V2, está fuertemente acoplado a FIWARE. Por lo tanto, solo sistemas basados en FIWARE pueden

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

Tabla 5.3: Importancia de los atributos del modelo ETA Valencia,

Atributo	Importancia
Distancia	0.188072
Longitud	0.183474
Latitud	0.167127
SOG	0.142248
COG	0.128106
Rumbo	0.103996
Calado	0.086977

integrarse directamente con el componente cognitivo, y en general con la arquitectura COSIBAS. En caso contrario, sería necesario hacer una adaptación *software*.

5.7.2. Servicios cognitivos

Puerto de Valencia

La selección de los atributos más relevantes para la construcción del modelo predictivo (cálculo de la ETA de los buques) se ha realizado usando el método *feature importance* en un clasificador *Random Forest*.

La Tabla 5.3 contiene los atributos finalmente seleccionados y su importancia. Los atributos seleccionados para el entrenamiento del modelo son: distancia, longitud, latitud, *Speed Over Ground* (SOG), *Course Over Ground* (COG), rumbo y calado. El resto de los atributos son descartados, ya que su contribución es despreciable. De esto se desprende que la información relativa a las condiciones meteorológicas marinas no mejora el rendimiento predictivo del modelo. Esto se debe a que la meteorología ya se reflejan en el atributo velocidad (SOG). Es el capitán del buque quien determina la navegación en función de las condiciones meteorológicas. Si el tiempo es desfavorable, el buque reduce su velocidad. Así que las variables oceanográficas solo añaden redundancia y complejidad innecesaria al modelo, lo que empeora su comportamiento [213].

El MAE obtenido de los datos AIS es de 1 066 minutos (se obtiene a partir del atributo ETA de los datos AIS y la ATA). Tras el desarrollo e integración del servicio cognitivo en el sistema, la predicción mejora sensiblemente. Los resultados de la validación del algoritmo se resumen en la Tabla 5.4. De todos los algoritmos de regresión utilizados durante el entrenamiento, el algoritmo KNN presenta los mejores resultados, con un MAE de 12.23 minutos, un RMSE bajo (21.09 minutos) y un R^2 aceptable (0.61). Por otro lado, aunque estas

Tabla 5.4: Resultados de las métricas de evaluación del modelo ETA Valencia,

Algoritmo	MAE	RMSE	R ²
DT	12.62	21.95	0.58
RF	13.71	24.12	0.49
SVR	13.00	22.76	0.54
KNN	12.23	21.09	0.61

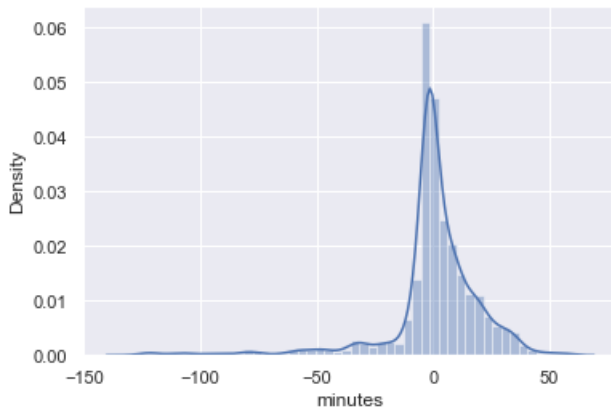


Figura 5.11: Distribución de error en modelo ETA Valencia.

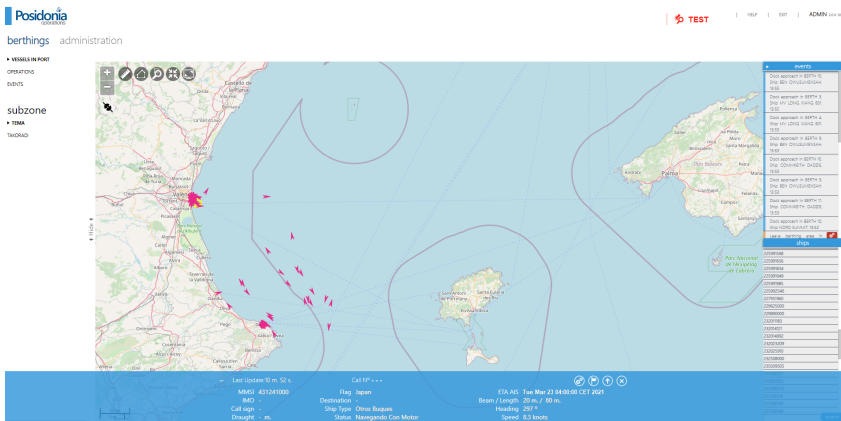
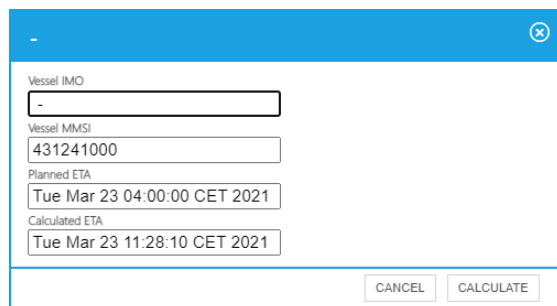


Figura 5.12: Interfaz de usuario - puerto de Valencia.

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE



Vessel IMO	-
Vessel MMSI	431241000
Planned ETA	Tue Mar 23 04:00:00 CET 2021
Calculated ETA	Tue Mar 23 11:28:10 CET 2021

CANCEL CALCULATE

Figura 5.13: Predicción de ETA.

métricas permiten evaluar el rendimiento del modelo, se pueden obtener más detalles trazando el error entre el valor predicho y el valor real en un gráfico de dispersión (Figura 5.11). El gráfico muestra el comportamiento deseado. La campana es estrecha y está centrada en cero. Esto indica que la mayoría de los errores que se producen están cerca de cero. Las peores predicciones, muy residuales, estimaron la hora de llegada del buque con un error de 50 minutos.

La Figura 5.12 muestra la interfaz gráfica de Port CDM en el puerto de Valencia. No es la interfaz definitiva del sistema, sino una interfaz de *test* utilizada durante la primera integración completa. Cuando el usuario selecciona un buque, aparece una ventana emergente. Esta nueva ventana se ha desarrollado para permitir al usuario solicitar y mostrar la predicción de la ETA de un buque específico. Al hacer clic en el botón de cálculo, el usuario inicia el flujo de mensajes descrito en la Sección 5.5.2. El adaptador de contexto se encarga de transformar la solicitud del usuario en una entidad NGS-V2 que inicia el flujo que desencadena el cálculo de la hora de llegada del buque a su destino mediante el servicio cognitivo de ETA del buque. Cuando el flujo de mensajes termina, la interfaz recibe una notificación del CEP y muestra la predicción (Figura 5.13).

Puerto de Algeciras

El método utilizado para la Sección de atributos es el mismo que en el caso anterior, *feature importance*. En esta ocasión se ha experimentado con la creación de nuevos atributos generados a partir de los ya disponibles: tiempo empleado para calcular la velocidad media (*Time Av.Speed*), velocidad media y aceleración. La Tabla 5.5 incluye la importancia de los atributos utilizados para la creación del modelo.

Tabla 5.5: Importancia de los atributos del modelo ETA Algeciras,

Atributo	Importancia
Time AvSpeed	0.198273
Distancia	0.156509
Velocidad media	0.134935
Latitud	0.130961
Longitud	0.120629
Velocidad	0.107028
Aceleración	0.073097
Eslora	0.044712
Manga	0.033856

Tabla 5.6: Resultados de las métricas de evaluación del modelo ETA Algeciras,

Algoritmo	MAE	RMSE	R ²
LR	16.13	25.81	0.85
DT	13.07	47.20	0.88
RF	11.79	19.61	0.91
GB	13.39	21.75	0.89
KNN	11.31	19.56	0.91

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

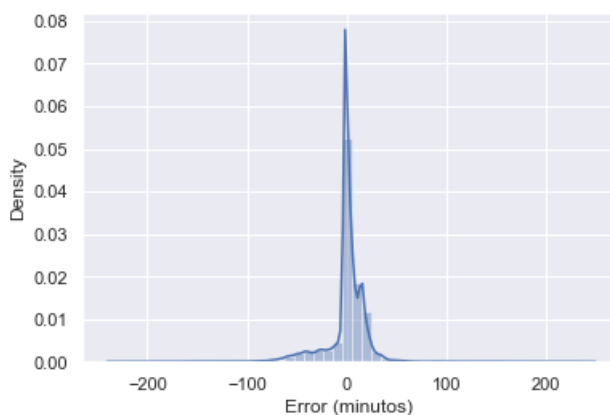


Figura 5.14: Distribución de error en modelo ETA Algeciras.

La Tabla 5.6 resume los resultados de las métricas de validación del modelo ETA Algeciras. Se observa que los resultados de los algoritmos RF y KNN son los mejores y los ajustes de todos los modelos son buenos (cerca de la unidad). El algoritmo LR consigue una MAE de 16.13 minutos con un RMSE bajo, 25.81 minutos y un R^2 alto (0.85). Se trata de un resultado bueno que muestra que en este caso el uso de los atributos seleccionados permite la resolución de problemas complejos con modelos simples. Este modelo es un buen candidato para ser utilizado en entornos con capacidad computacional ajustada. Si se desea obtener la mayor precisión posible y el coste computacional no es problema, el mejor modelo es el proporcionado por el algoritmo KNN, que ofrece una MAE de 11.31 minutos, RMSE de 19.56 minutos y R^2 de 0.91. La Figura 5.14 muestra la distribución de error con el algoritmo KNN. Como se observa en la Figura, los errores están centrados en cero y la función de distribución del error es más estrecha que en el caso del puerto de Valencia, lo cual indica que se producen menor cantidad de errores grandes.

La comparación de las métricas obtenidas en ambos puertos muestra que la predicción lograda en el puerto de Algeciras es mayor. Se consigue un R^2 cercano a 1 (0.91) y un MAE de 11.31 minutos, mientras que en el caso de Valencia el R^2 es 0.61 y el MAE 12.23 minutos. Esto se debe a que el *dataset* de Algeciras es mayor y los algoritmos pueden extraer mayor conocimiento de él.

La Figura 5.15 muestra la interfaz definitiva de Posidonia Port CDM. En el menú izquierdo se listan los buques detectados por la antena AIS en el rango mostrado en la interfaz. La ubicación y la dirección de los buques se representan

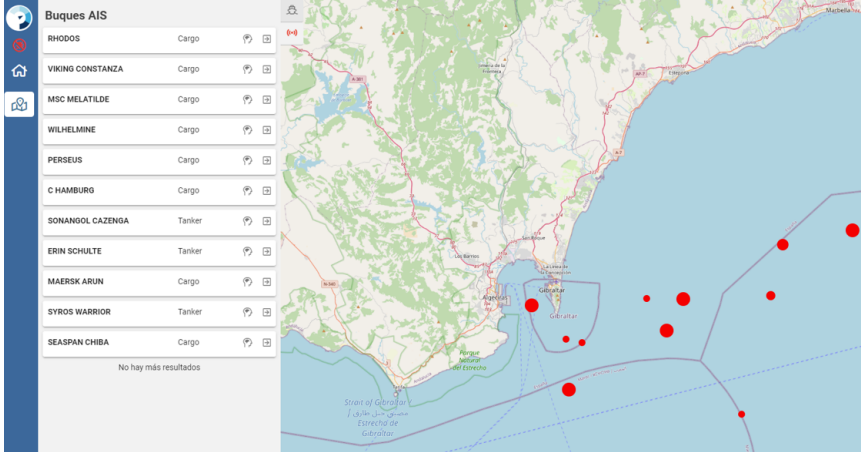


Figura 5.15: Interfaz de usuario - puerto de Algeciras.

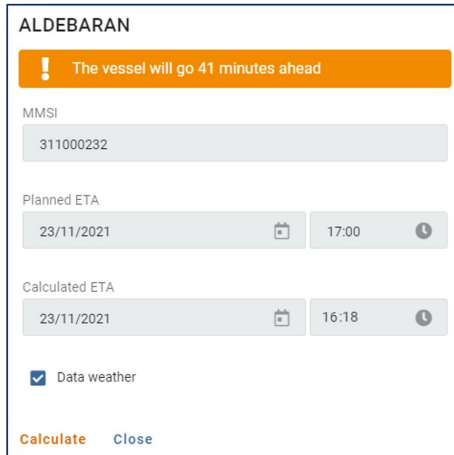


Figura 5.16: Resultado de cálculo de ETA.

CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE

mediante una flecha en la parte derecha de la pantalla. Al seleccionar un buque se despliega un menú que permite lanzar la solicitud al servicio cognitivo. Tras esto, el sistema inicia el flujo de mensajes con la arquitectura COSIBAS, la cual devuelve a través del CEP el resultado de la predicción y este es mostrado en la ventana (Figura 5.16).

5.8. Conclusiones

En este Capítulo se ha presentado el diseño e implementación de una arquitectura AIoT para la creación de servicios cognitivos en el ecosistema FIWARE. Según el conocimiento del autor, este trabajo presenta la primera arquitectura AIoT afín al ecosistema que permite la creación de servicios cognitivos conformes con las directrices y estándares de FIWARE.

Para ello, el componente cognitivo es un elemento clave. Ha sido específicamente diseñado para facilitar la incorporación de servicios cognitivos en el ecosistema. El componente cognitivo almacena la información de contexto en OCB y se comunica con el resto de los elementos del sistema haciendo uso de los mecanismos de notificación asíncrona de FIWARE. La incorporación de este componente en el ecosistema permite incorporar servicios cognitivos basados en IA en sistemas existentes con cambios mínimos para el sistema original.

La arquitectura propuesta ha sido implementada y validada para un sistema real IoT marítimo, Posidonia PortCDM, a través de la arquitectura COSIBAS. En particular, se han incorporado al sistema original capacidades cognitivas para mejorar la estimación de la hora estimada de llegada de los buques (ETA).

La validación del sistema se ha llevado a cabo en dos pilotos: el puerto de Valencia y el puerto de Algeciras. Para ambos pilotos se ha desarrollado un servicio cognitivo específico capaz de modelizar las características de cada puerto y predecir la llegada de los buques basándose en su información AIS y oceanográfica. Estos servicios están basados en algoritmos de regresión de ML que calculan los minutos que le faltan a un buque para llegar a su destino. Los resultados son satisfactorios, ya que muestran una mejora significativa en la estimación de los tiempos de llegada de los buques. El análisis revela que la información oceanográfica no aporta información al modelo.

Tras la integración inicial del componente cognitivo en el sistema heredado, la incorporación de nuevos servicios cognitivos se vuelve mucho más sencilla. Así, el esfuerzo en incorporar nuevos servicios se concentraría únicamente en el desarrollo de los mismos, ya que el componente que los integra en el sistema ya se ha incorporado. Algunos ejemplos a servicios a incorporar podrían ser la predicción de la demanda de carga, la gestión de inventario o el seguimiento de carga.

De forma general, el enfoque propuesto permite ampliar la vida útil efectiva de soluciones IoT heredadas, a través de la adición de capacidades cognitivas, sin incurrir en un aumento significativo de los costes operativos. Además, es importante destacar que, aunque la transformación propuesta solo se ha integrado y validado en una solución de gestión portuaria, está diseñada para incorporar servicios cognitivos de cualquier dominio (salud, transporte, ciudades inteligentes, etc.) dentro del ecosistema FIWARE. Por otro lado, la transformación propuesta no solo es aplicable a soluciones existentes, sino que también puede ser implementada como base para desarrollar sistemas completamente nuevos en el contexto de FIWARE.

La mayor desventaja que presenta la arquitectura es su estrecha dependencia con FIWARE. No obstante, la definición de sus bloques es lo suficientemente genérica como para servir de punto de partida para la generación de una arquitectura AIoT genérica. El siguiente Capítulo explora este punto.

**CAPÍTULO 5. ARQUITECTURA AIOT PARA LA CREACIÓN
DE SERVICIOS COGNITIVOS EN EL ECOSISTEMA FIWARE**

Capítulo 6

Arquitectura AIoT para la creación de servicios cognitivos avanzados

6.1. Introducción

En los Capítulos previos se sientan las bases de una arquitectura AIoT con un enfoque integral y flexible, diseñada para permitir la reutilización de componentes. La tercera y última instanciación combina el conocimiento adquirido en los Capítulos anteriores para dar lugar a una arquitectura AIoT de carácter genérico y plenamente independiente tanto del dominio como de la plataforma IoT.

En esta arquitectura, el componente cognitivo ocupa un lugar central. Un aspecto clave de este componente es su capacidad para funcionar de manera independiente del dominio específico, lo que permite su aplicación en una amplia variedad de contextos. Además, se diseña para ser un módulo fácilmente adaptable a diversas aplicaciones y plataformas IoT. Su versatilidad no solo simplifica el proceso de desarrollo, sino que también facilita la escalabilidad de las soluciones asociadas.

En este Capítulo la arquitectura de referencia se instancia y valida en el dominio de la industria 4.0. El caso de uso asociado consiste en el desarrollo e integración, en una línea de producción robotizada de antenas, de un servicio cognitivo capaz de detectar variaciones anómalas en la tasa de recepción de los datos de producción. Las variaciones en la tasa de recepción de datos pueden

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

indicar anomalías en la línea de producción y su detección automática permite un mantenimiento proactivo sin necesidad de intervención humana.

En primer lugar, se presenta la tercera transformación de la arquitectura de referencia, que habilita la creación de servicios cognitivos avanzados en entornos AIIoT. Seguidamente, se presenta el caso de uso para la detección de anomalías en una línea de producción real y la implementación de la arquitectura OPTIMAI. A continuación, se lleva a cabo la validación de la arquitectura, seguida de la exposición y análisis de los resultados obtenidos. Para finalizar, el Capítulo cierra con las conclusiones más destacadas.

6.2. Arquitectura

A modo de última validación de la arquitectura de referencia AIIoT, se define una tercera transformación. El objetivo de esta transformación es la creación de un sistema AIIoT, no limitado por las restricciones de sistemas o entornos preexistentes, que permita gestionar servicios cognitivos genéricos sin fisuras.

Esta última transformación utiliza el conocimiento extraído de los Capítulos anteriores para obtener y procesar datos a través de una integración fluida entre IIoT e IA. La primera transformación de la arquitectura es clave para consolidar aquellos aspectos fundamentales para asegurar la creación de *datasets* homogéneos, mientras que la segunda transformación permite establecer los requerimientos necesarios para la integración sin fisuras de servicios cognitivos en sistemas IIoT.

Los objetivos específicos de la arquitectura son:

- Adquisición y almacenamiento de datos de diferentes fuentes en tiempo real.
- Recogida, gestión y provisión de acceso a los datos.
- Proporcionar mecanismos inteligentes de gestión y distribución de datos.
- Despliegue de servicios de IA.

La Figura 6.1 muestra los diferentes componentes que conforman la arquitectura y las comunicaciones entre ellos. Empezando por abajo, se desea que los datos se recolecten en tiempo real mediante un agente IIoT. Este traduce los datos a un formato común y los inserta en el *context broker*. Tras esto, los datos son procesados por los modelos de IA ubicados en el componente cognitivo. A continuación se describen los componentes de la arquitectura.

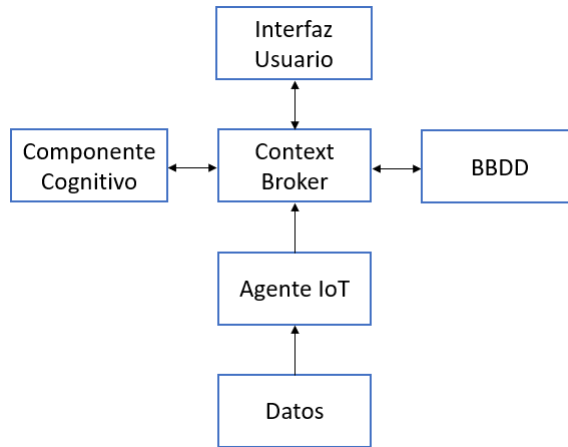


Figura 6.1: Arquitectura AIoT para la creación de servicios cognitivos.

6.2.1. Agente IoT

Los agentes IoT permiten la comunicación de los dispositivos IoT con un *middleware* (en este caso el *context broker*). Actúan como un enrutador de red, dirigiendo los datos desde los sensores hacia el *middleware*. Pueden gestionar tanto el tráfico entrante como el saliente. Los flujos de tráfico saliente se utilizan para enviar datos de sensores al *middleware*, mientras que el tráfico entrante se utiliza para recibir comandos de actuación.

Este componente gestiona la creación de entidades para cada dispositivo IoT, produce un contexto para los atributos vinculados a las capacidades de detección y observación y proporcionan una API de administración/configuración.

Las principales funcionalidades de un agente IoT son:

- Comunicación con dispositivos.
- Gestión de datos no estructurados.
- Conversión de los datos de los dispositivos a un modelo de datos común.
- Conversión de peticiones al formato sintáctico de los dispositivos.
- Generación de información de contexto (metadatos).
- Gestión de entidades de contexto relacionadas con los dispositivos.

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

Existen agentes IoT para muchos protocolos de comunicación y modelos de datos. Algunos ejemplos son IoTAgent-JSON, IoTAgent-LWM2M o IoTAgent-UL [217].

6.2.2. Context broker

Como ya se ha descrito en el Capítulo anterior, OCB es una plataforma IoT desarrollada por la fundación FIWARE. Su arquitectura distribuida y escalable, junto con su soporte para estándares abiertos y protocolos, permite gestión y procesamiento de datos de sensores y dispositivos en tiempo real.

OCB se basa en estándares abiertos y utiliza el protocolo *Next Generation Service Interface* (NGSI) para la gestión de datos. El protocolo NGSI define un conjunto de interfaces y servicios que permiten la integración de datos y la comunicación entre diferentes sistemas y dispositivos IoT. Además, OCB proporciona una API REST que permite a los desarrolladores acceder y utilizar los datos en tiempo real para construir aplicaciones y servicios IoT.

Los datos recogidos por el agente son almacenados en OCB. A través de su mecanismo de suscripción, OCB notifica al componente cognitivo cuando llegan nuevos datos de la línea de producción. Además, este componente almacena los resultados obtenidos por el algoritmo de detección de anomalías.

6.2.3. Componente cognitivo

A diferencia del Capítulo anterior, esta implementación de la arquitectura apuesta por un componente cognitivo independiente de la plataforma IoT. De esta forma no se limita la reutilización del componente entre diferentes ecosistemas IoT. En lugar de desarrollar un componente desde cero, el componente cognitivo se implementa reutilizando elementos existentes. Esta estrategia no solo ahorra tiempo y recursos, sino que también permite una mayor flexibilidad y adaptabilidad en el ecosistema AIoT.

Uno de los desafíos más destacados su diseño y desarrollo es la necesidad de mantener unos tiempos de respuesta extremadamente rápidos. Si bien es cierto que las inferencias y cálculos realizados por los algoritmos de IA pueden ser intrínsecamente complejos y, en algunos casos, llevar tiempo, es necesario que este componente no agregue retrasos adicionales. En muchas aplicaciones y dominios, especialmente aquellos en los que se requiere una toma de decisiones en tiempo real o en los que la eficiencia es crucial, los tiempos de respuesta ágiles son esenciales para cumplir con las expectativas de los usuarios y lograr un funcionamiento eficaz.

El componente cognitivo (Figura 6.2) se diseña teniendo en cuenta dos factores: la modularidad y la simplicidad. El primero se refiere a facilitar la

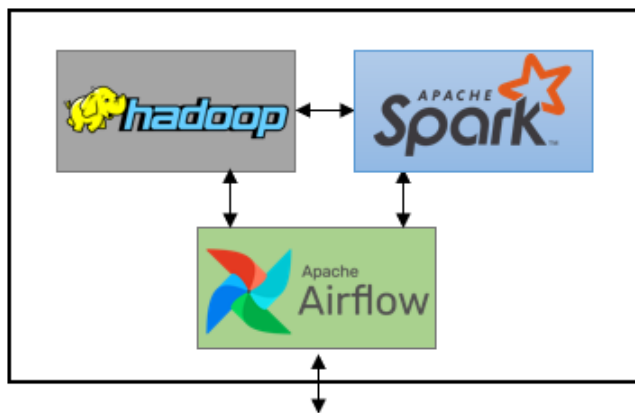


Figura 6.2: Componente cognitivo.

actualización e implementación de módulos adicionales. El segundo se centra en la provisión de servicios *cloud*, ofreciendo un mecanismo de implementación sencillo para soluciones de inteligencia artificial con la posibilidad de que sea distribuido. Además, ofrece capacidades para gestionar y agregar datos provenientes de diferentes fuentes. A continuación se describen los módulos que lo forman.

Apache Airflow

Apache Airflow [218] es una plataforma de orquestación de procesos de datos que proporciona una forma fácil de crear, programar y supervisar flujos de trabajo complejos de procesamiento de datos. Permite manejar grandes cantidades de datos y se requiere una gestión eficiente de procesos complejos. Además, Airflow es altamente escalable y extensible, lo que permite la integración con múltiples herramientas y sistemas, incluyendo sistemas de almacenamiento de datos, bases de datos, sistemas de colas y servicios en la nube.

En el caso de uso, Apache Airflow es el controlador del flujo de datos del componente cognitivo. Recibe datos de OCB, define el flujo de procesamiento adecuado y, por último, almacena el resultado en OCB.

Hadoop Distributed File System (HDFS)

El objetivo principal de HDFS [219] es proporcionar un sistema de almacenamiento escalable y tolerante a fallos para *datasets* grandes. HDFS divide los datos en bloques y los distribuye en múltiples nodos en un clúster de

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

computadoras, lo que permite el acceso y procesamiento de los datos de manera distribuida y paralela.

HDFS también proporciona una serie de características adicionales, incluyendo replicación de datos, tolerancia a fallos y recuperación de datos automática. Los datos se copian en múltiples nodos en el clúster para garantizar que los datos estén disponibles incluso en caso de fallo de un nodo. La tolerancia a fallos permite que el sistema siga funcionando incluso si un nodo en el clúster falla, mientras que la recuperación de datos automática permite que el sistema se recupere de manera automática en caso de fallo.

Dentro del caso de uso, HDFS se encarga de almacenar el modelo de IA para la detección de anomalías. También almacena de forma temporal el resultado de la inferencia del algoritmo.

Apache Spark

Apache Spark [220] es un motor de procesamiento de datos distribuido de código abierto diseñado para trabajar con *datasets* grandes en paralelo en clústeres de ordenadores. Puede gestionar cargas de trabajo de análisis y procesamiento de datos tanto por lotes como en tiempo real. Una de las características clave de Spark es su capacidad para procesar datos en memoria, lo que significa que los datos se almacenan en la memoria RAM en lugar de en el disco, acelerando significativamente el tiempo de procesamiento. Spark también proporciona una API de programación que permite a los desarrolladores crear aplicaciones y servicios complejos de análisis de datos.

En el caso de uso, Apache Spark es el elemento encargado de ejecutar el modelo de detección de anomalías almacenado en HDFS.

6.2.4. Base de datos de series temporales

Las bases de datos de series temporales permiten realizar análisis retrospectivos y predictivos, ya que se puede acceder y consultar el estado de los datos en diferentes momentos. Esto resulta especialmente útil para aplicaciones que necesitan rastrear cambios y tendencias a lo largo del tiempo, como el monitoreo de sensores, el seguimiento de registros de eventos o la gestión de datos geoespaciales. Además, las bases de datos temporales permiten realizar consultas específicas basadas en intervalos de tiempo, lo que facilita la obtención de información relevante para un período determinado. Esto es esencial para el análisis histórico, la generación de informes y la toma de decisiones basadas en datos.

La base de datos escogida en este caso de uso es QuantumLeap, un servicio REST para almacenar, consultar y recuperar datos espacio-temporales en

6.3 Caso de uso: detección de anomalías en la tasa de recepción de datos OEE

formato NGS-V2 y NGS-LD. QuantumLeap convierte datos semiestructurados de NGS en un formato tabular y los almacena en una base de datos de series temporales, asociando cada registro de la base de datos con un índice de tiempo y a una ubicación. Los clientes REST pueden recuperar entidades NGS filtrando conjuntos de entidades mediante rangos de tiempo y operadores espaciales.

6.2.5. Interfaz de usuario

La arquitectura proporciona una interfaz de usuario que permite, entre otros, definir y personalizar la visualización de datos, configurar reglas de alerta y acceso a servicios cognitivos, y tomar decisiones basadas en la información contextual proporcionada por el sistema AIoT.

En cualquier caso, los requisitos de la interfaz de usuario dependerán de las reglas de negocio y las necesidades específicas de cada aplicación AIoT.

6.3. Caso de uso: detección de anomalías en la tasa de recepción de datos OEE

La Industria 4.0 [221], también llamada la cuarta revolución industrial, se refiere a la automatización y la integración de datos en los procesos de fabricación. Esta tendencia está originando un gran impacto en todas las áreas de la producción, produciendo una mejora significativa en la eficiencia y productividad, así como en la creación de modelos de negocio innovadores y en las cadenas de valor. Esta revolución se caracteriza por la aparición de tecnologías innovadoras tales como la robótica, la IA, ML o IoT. Estas tecnologías permiten la integración de sistemas físicos y digitales, lo que facilita la creación de fábricas inteligentes capaces de tomar decisiones autónomas y optimizarse a sí mismas. Los fabricantes están adoptando estas tecnologías avanzadas en sus plantas de producción y en todas sus actividades.

Sin embargo, a pesar de esta visión, la Industria 4.0 sigue siendo un concepto difícil de materializar para las industrias manufactureras. Una de las mayores dificultades radica en la calidad y la cantidad de datos necesarios para entrenar los modelos de IA y hacer que funcionen de manera efectiva. Las empresas pueden tener dificultades para recopilar y analizar grandes cantidades de datos de manera eficiente, especialmente si no tienen una infraestructura tecnológica diseñada para tal fin. Además, es frecuente encontrar resistencia por parte de los empleados a la adopción de tecnologías de IA. A menudo se sientan intimidados por la tecnología, temerosos de que puedan reemplazarlos en el futuro. Además, algunos pueden no sentirse cómodos al aprender nuevas

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

habilidades y adaptarse a un nuevo entorno de trabajo. Por lo tanto, es esencial que las empresas aborden esta resistencia de manera efectiva para garantizar una transición fluida y exitosa.

La línea de fabricación de antenas de la empresa TELEVÉS S.A.U. (TVES) produce más de un millón de antenas al año con muchas configuraciones diferentes, lo que implica el uso de muchas piezas distintas. Los factores clave del negocio de antenas son un alto rendimiento a un coste ajustado. Por esta razón, es obligatoria una línea altamente automatizada y especializada, donde una detención por fallo tardío puede provocar una cantidad de desperdicios importante.

Esta línea de fabricación de antenas se ha diseñado siguiendo un enfoque de gran volumen y alta combinación de diseños. Por lo tanto, es fundamental realizar cambios de configuración rápidos y fiables y detener los fallos de las piezas lo antes posible en el proceso de fabricación. Como resultado, TVES es capaz de producir antenas con un alto rendimiento y un coste ajustado para que la empresa pueda seguir siendo competitiva frente a competidores que producen en países de bajo coste. Para lograrlo, esta línea se reutiliza constantemente y funciona con distintas configuraciones de productos. Esta operación requiere el desmontaje y montaje regular de los equipos por lotes, lo que conlleva un consumo de tiempo significativo y aumenta los costes de producción. Además, las configuraciones periódicas son más propensas a causar defectos y cada vez es más difícil detectar su causa cuando la operación cambia a menudo.

La línea consta de un conjunto de células robóticas y máquinas que ejecutan secuencialmente cada uno de los procesos de producción necesarios para la fabricación de antenas. Por motivos de confidencialidad no es posible incluir la disposición de la línea de fabricación. Es un escenario complejo en el que cada célula robótica y/o máquina, además del robot o automatismo que constituye su elemento central, cuenta con periféricos auxiliares y múltiples sensores (sensores de presencia inductivos, sensores ópticos, cámaras, sensores 3D, etc.).

Con el objetivo de medir la eficiencia de la línea de producción, TVES ha desarrollado un sistema para calcular el indicador *Overall equipment effectiveness* (OEE). La explotación de estos datos ha permitido a la empresa identificar nuevas áreas de mejora en la línea de producción. Para facilitar el acceso a los datos de la línea de producción necesarios para calcular este indicador, TVES ha desarrollado una API REST que permite recuperar los datos de las sesiones de fabricación de cada una de las células robóticas de la línea. Para su desarrollo, ha sido necesario obtener ciertas métricas clave para cada una de estas células/máquinas robóticas que componen la línea de fabricación. Los principales campos de los datos capturados se recogen en la Tabla 6.1. Se debe tener en cuenta que solo se obtienen datos de las células robóticas que tienen un identificador en la columna “ID de API” de la Tabla anterior.

6.3 Caso de uso: detección de anomalías en la tasa de recepción de datos OEE

Datos que identifican una sesión de fabricación en una máquina:	
grupo máquina	Agrupación de máquinas, que en este caso es una línea de producción
operador	Máquina o robot
referencia	Operador responsable de la línea
orden	Referencia de producción
selloTemporalInicio	Lote de producción
	Marca de tiempo en la que se inicia la sesión actual.
	Actúa como identificador de sesión
Métricas acumuladas por sesión:	
tiempoSesionActiva	Tiempo total durante el cual la sesión está activa
tiempoTrabajo	Tiempo total durante el cual una sesión está activa
tiempoEsperaEntradaLarga	Tiempo total en el que la máquina espera más de 1 minuto sin carga de trabajo
tiempoEsperaEntradaCorta	Tiempo total en el que la máquina espera menos de 1 minuto sin carga de trabajo
tiempoEsperaSalidaLarga	Tiempo total en el que la máquina espera más de 1 minuto por línea saturada
tiempoEsperaSalidaCorta	Tiempo total en el que la máquina espera menos de 1 minuto por línea saturada
tiempoErrorLargo	Tiempo total de error de la máquina superior a 1 minuto
tiempoErrorCorto	Tiempo total de error de la máquina inferior a 1 minuto
piezasBuenas	Suma de las piezas producidas correctamente
piezasMalas	Suma de las piezas producidas incorrectamente
piezasAusentes	Suma de las piezas ausentes
Otros datos:	
selloTemporal	Fecha y hora de los datos de fabricación
fechaRecepcion	Fecha de recepción de los datos en el servidor de la planta
recepcionServidorCentral	Fecha de recepción de los datos en el servidor central
ciclo	Duración teórica del ciclo de fabricación de la máquina

Tabla 6.1: Datos de una sesión de fabricación.

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

El caso de uso presentado en este Capítulo busca continuar explotando estos datos para continuar mejorando la eficiencia de la línea. Para tal fin se introduce la búsqueda de anomalías en los datos anteriores.

La detección temprana de anomalías en la tasa de recepción de datos en una línea de fabricación de antenas es de vital importancia para asegurar la calidad y eficiencia de la producción. Estas anomalías pueden ser indicio de posibles problemas en la línea, tales como fallos en los sistemas de control, averías en los equipos o variaciones en los procesos de fabricación. Detectarlas de forma automática mediante la implementación de servicios cognitivos, puede permitir a los encargados de mantenimiento tomar medidas proactivas para corregir el problema antes de que se agrave, lo que se traduce en una reducción de costos y un incremento en la productividad y calidad de los productos manufacturados.

En este sentido, la detección temprana de anomalías en la tasa de recepción de datos se ha convertido en una práctica indispensable en la industria manufacturera moderna. Así pues, el objetivo de este caso de uso es el análisis, a través de un servicio cognitivo, de los datos de las sesiones de fabricación (Tabla 6.1) en búsqueda de anomalías.

6.4. Implementación

A partir de la arquitectura presentada en la Sección 6.2 se define e implementa la arquitectura OPTIMAI.

6.4.1. Relación con la arquitectura de referencia

En la presente Sección se describe la relación de la arquitectura OPTIMAI con la arquitectura global. La Figura 6.3 muestra los bloques seleccionados de la arquitectura global para su diseño. Es importante destacar que esta arquitectura se implementa para ser genérica y compatible con cualquier plataforma IoT. La arquitectura no implementa los niveles de seguridad y privacidad de la arquitectura de referencia.

Al igual que en la arquitectura presentada en el Capítulo anterior, el nivel de percepción está formado por fuentes de datos externas en lugar de sensores IoT. Este nivel proporciona información de la eficiencia de los equipos utilizados en la línea de producción de antenas.

El nivel de *middleware* y el de datos semánticos se encuentran a la misma altura. Los datos de la línea de producción se integran en el *middleware* a través de un agente IoT. El agente forma parte del nivel semántico de datos y funciona como adaptador semántico. Su función principal es actuar como un intermediario entre la línea de producción de antenas y la arquitectura OPTI-

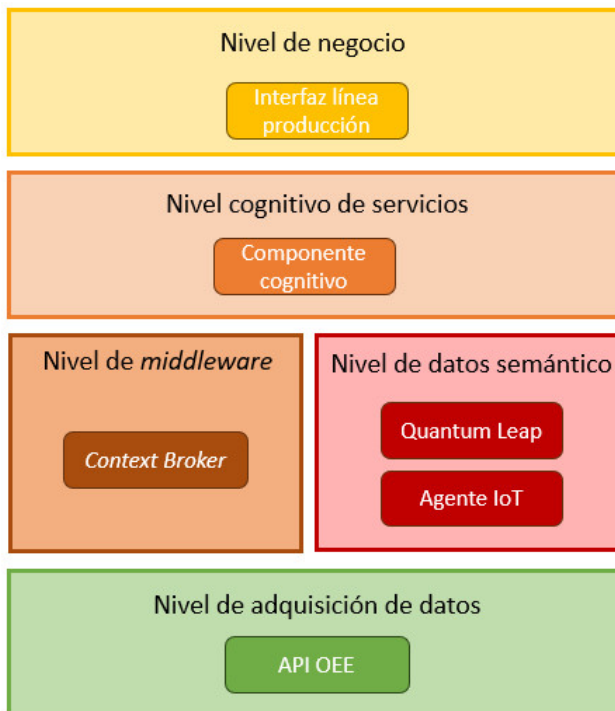


Figura 6.3: Arquitectura OPTIMAI - Relación con la arquitectura de referencia.

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

MAI, traduciendo el formato de datos utilizado en la línea en el formato de datos NGS-V2. Otros elementos de la arquitectura OPTIMAI que pertenecen al nivel semántico de datos son la base de datos QuantumLeap, encargada de almacenar información histórica, y Hadoop, que proporciona la infraestructura necesaria para administrar y almacenar los datos durante la inferencia del modelo.

El nivel cognitivo de servicios lo compone el componente cognitivo, formado a su vez por Apache Airflow y Apache Spark. Apache Airflow es el orquestador del flujo de trabajo del nivel cognitivo de servicios, mientras que Apache Spark proporciona funcionalidades de procesamiento cognitivo y análisis de datos. Cuando el *context broker* recibe nuevos datos de la línea de producción, envía una petición a Apache Airflow para detectar si se ha producido alguna anomalía en la línea. En ese momento, Airflow solicita a Apache Spark que analice los datos con el modelo guardado en Hadoop.

Por último, el nivel de negocio está formado por la interfaz de la línea de producción. En caso de detectarse una anomalía en la tasa de recepción de datos se muestra una alerta.

6.4.2. Arquitectura OPTIMAI

La Figura 6.4 ilustra la implementación de la solución para el caso de uso de la industria 4.0: la arquitectura OPTIMAI. Su objetivo principal es la simplificación de la implementación de sistemas de producción inteligente que aprovechan sensores con capacidades de inteligencia artificial para desempeñar labores de supervisión y control de calidad de manera efectiva. Permite la integración de los datos de las fábricas para posteriormente analizarlos mediante modelos de IA y detectar defectos o anomalías en una fase temprana de los procesos de producción. Además, busca identificar deficiencias en las primeras etapas del proceso y reconfigurar los parámetros de fabricación con el fin de optimizar la producción y mejorar la eficiencia del sistema en su conjunto.

Datos de la línea de producción

Para proporcionar acceso a los datos de las células robotizadas, se ha desarrollado una API REST. Esta API tiene los siguientes *endpoints* que se muestran en la Figura 6.5. Se describen a continuación:

- **/api/oe/data/group:** Es el *endpoint* principal con el que se obtienen todas las métricas disponibles. Dispone de múltiples filtros que permiten realizar consultas muy específicas.

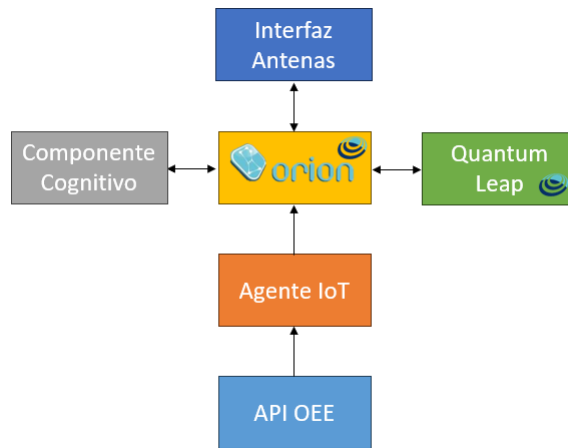


Figura 6.4: Arquitectura OPTIMAL.

- **/api/oee/groups:** Obtiene los grupos sobre los que podemos realizar consultas. Se entiende por grupo un conjunto de células robotizadas y/o máquinas, que en este caso es la línea robotizada para la fabricación de antenas.
- **/api/oee/machines/group:** Obtiene las células robóticas y/o máquinas que forman parte del grupo pasado como parámetro y que tienen capacidad para registrar datos operativos.
- **/api/oee/referencias/group:** Obtiene las referencias producidas en el último grupo como parámetro.
- **/api/users/token:** Si se invoca con las credenciales adecuadas, se obtiene un token (y su fecha de caducidad) para poder interactuar con el resto de la API.

Agente IoT

El agente IoT es el componente encargado de capturar, traducir e insertar en OCB los datos OEE en tiempo real. En lugar de emplear los agentes IoT proporcionados por la FIWARE Foundation, con un diseño poco reusable y extensible, se ha decidido desarrollar un agente IoT propio genérico que sigue los principios SOLID [222].

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

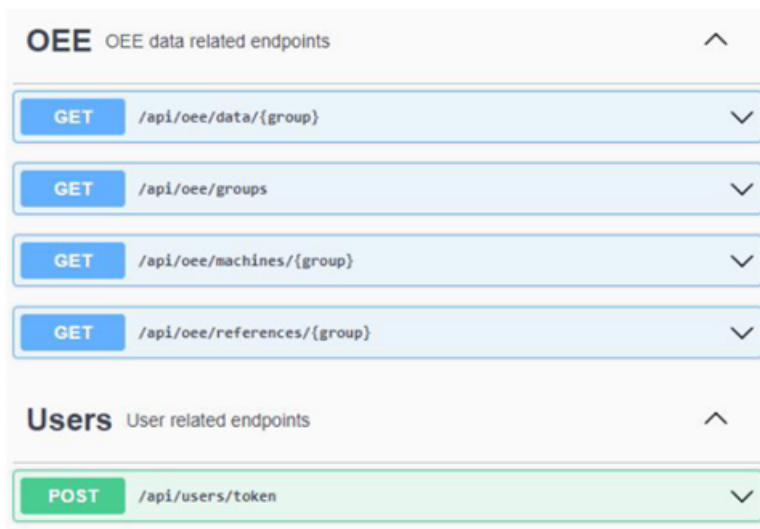


Figura 6.5: TVES OEE API swagger.

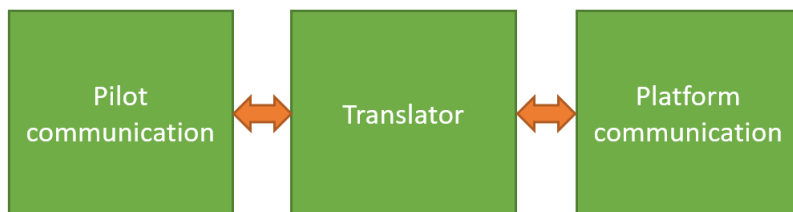


Figura 6.6: Diseño e implementación del agente IoT.

Con ello se pretende implementar un agente IoT que sea robusto y estable, además de flexible ante los cambios. Es decir, que se pueda modificar fácilmente según necesidad, y que por lo tanto, sea fácilmente reutilizable y mantenible. Por último, el agente IoT se diseña para permitir su escalabilidad. Su diseño debe permitir ser ampliado con nuevas funcionalidades de manera ágil.

El agente IoT está compuesto de tres módulos (Figura 6.6). El primer módulo, *pilot communication*, (izquierda) se encarga de las comunicaciones con el piloto. El segundo módulo, *translator*, (centro) realiza la traducción del formato de datos origen al formato de datos destino. Por último, el tercer módulo (derecha) se comunica con la plataforma IoT escogida. Los tres módulos están

desacoplados entre sí, de forma que si es necesario aplicar cambios en alguno de ellos los otros no se ven afectados. Esto también permite sustituir las implementaciones concretas de los módulos por otras, por ejemplo, para comunicarse con un nuevo servicio, dejando inalterados los otros módulos.

El agente IoT utilizado en este caso de uso está implementado en Python 3.9 y emplea la librería *pyngsi* [223] que proporciona soporte para el manejo de entidades NGSI-V2. Específicamente, cada 30 segundos el agente realiza una petición a la API OEE para recuperar los últimos datos disponibles. A continuación, los traduce a NGSI-V2 y, por último, ingresa los datos en FIWARE.

Modelo de IA para detección de anomalías

El caso de uso de la línea de producción de antenas busca desarrollar e integrar un modelo de IA capaz de detectar anomalías en la tasa de recepción de los datos OEE de la línea de producción. El modelo se ubica en el componente cognitivo e incorpora un modelo entrenado utilizando técnicas de ML. El modelo se ha entrenado utilizando la librería *Scikit-learn* de Python y serializado para su integración en el servicio con la librería *Pickle*.

Para el desarrollo del modelo de detección de anomalías en la tasa de recepción de los datos OEE se han realizado dos experimentos. El primer experimento se basó en el uso de algoritmos no supervisados de detección de anomalías, en el cual se buscó identificar patrones atípicos en los datos mediante la exploración y análisis de los registros. En el segundo experimento, se utilizaron algoritmos supervisados de clasificación binaria (anomalía/no anomalía), con el objetivo de que el modelo pueda aprender a distinguir entre registros normales y anómalos a partir de un *dataset* etiquetado. Ambos experimentos se llevaron a cabo con el fin de seleccionar el enfoque más adecuado para el desarrollo de un modelo de IA que permita detectar de manera eficaz las anomalías en la tasa de recepción de los datos OEE.

El *dataset* de entrenamiento fue recopilado utilizando la API REST OEE. Contiene 431 923 registros, obtenidos durante el período comprendido entre el 1 de septiembre de 2022 y el 15 de enero de 2023. Incluye atributos como la marca de tiempo, el identificador de la máquina correspondiente a los valores OEE, la cantidad de piezas buenas, malas o faltantes, la eficiencia y el tiempo total de funcionamiento de la máquina. El detalle de todos los atributos del *dataset* están recogidas en la Tabla 6.1.

Durante la selección de atributos y el preprocesamiento, se eliminaron los registros que contenían campos incompletos, reduciendo el *dataset* a 373 436 registros. A continuación, se seleccionaron los atributos a utilizar durante el entrenamiento de los algoritmos. En primer lugar, para detectar variaciones en la tasa de recepción de datos OEE, se añadió un nuevo atributo mediante

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

técnicas de fusión de datos. El nuevo atributo indica, para cada máquina de la línea de fabricación de antenas, los segundos transcurridos desde la última medición recibida de esta máquina. Por otra parte, se utiliza la codificación *one-hot* para codificar cada día de la semana. Con ello se pretende que los algoritmos puedan detectar patrones en los eventos que se producen en determinados días de la semana. Además, la hora se extrae del campo *timestamp* para crear un atributo adicional.

Tabla 6.2: Atributos seleccionados.

Atributo
Lunes
Martes
Miércoles
Jueves
Viernes
Hora
Tiempo delta
Anomalía

El *dataset* de entrenamiento no contiene etiquetas. Después de analizarlo, se detectaron patrones que siguen a los datos anómalos. Gran parte de los registros se reciben con una diferencia de tiempo (delta) de entre 60 y 70 segundos. La mayoría de los registros restantes se consideran anómalos, con dos excepciones. En primer lugar, temprano los lunes después del cierre del fin de semana, el primer registro recibido tiene un tiempo delta alto (entre 190 000 y 240 000 segundos). En segundo lugar, el resto de los días de la semana se aprecia un comportamiento similar. Cuando la línea de producción comienza a operar, cada máquina comienza a funcionar y se reciben registros con tiempos delta altos. Estos registros no deben considerarse anómalos, ya que siguen el comportamiento esperado de la línea de producción.

La Tabla 6.2 muestra los atributos utilizados durante los experimentos.

6.5. Validación

La validación de la arquitectura se realiza en dos pasos. En primer lugar, se valida el correcto funcionamiento del agente IoT mientras que en el segundo paso se evalúan los modelos incorporados en los servicios cognitivos.

Al igual que durante el desarrollo del componente cognitivo del Capítulo anterior, la creación y validación del agente IoT se lleva a cabo mediante la

metodología TDD. La validación incluye tanto *tests* unitarios como de aceptación.

Los *tests* unitarios son pruebas diseñadas para validar el funcionamiento de una unidad de código individual. En el caso del agente IoT, los *tests* unitarios se han utilizado para detectar errores en la fase de desarrollo temprana y evitar que se escalen a problemas mayores. Estos *tests* se centran en verificar el comportamiento y el funcionamiento del agente en una variedad de situaciones y escenarios.

Por otro lado, los *tests* de aceptación se utilizan para validar que el agente cumpla con los requerimientos y especificaciones del sistema. Estos *tests* incluyen los principales escenarios de uso del agente y se diseñan para simular situaciones del mundo real en las que el agente debe operar. Los resultados de los *tests* de aceptación son cruciales para garantizar que el agente funcione correctamente y cumpla con los requisitos del sistema.

A través de una evaluación experimental, se ha verificado el rendimiento de los modelos obtenidos durante los experimentos. Las métricas utilizadas para la validación son: matriz de confusión, *accuracy*, *precision* y *recall*.

La matriz de confusión (Tabla 6.3) es una herramienta utilizada en modelos de clasificación para evaluar su rendimiento. La matriz proporciona un resumen del número de predicciones correctas e incorrectas realizadas por el clasificador, y organiza los resultados por cada clase en el *dataset*. La matriz de confusión se basa en los términos:

- Los verdaderos positivos (en inglés, TP) son instancias donde el algoritmo indica correctamente la presencia de una condición o atributo.
- Los verdaderos negativos (en inglés, TN) son instancias donde el algoritmo indica correctamente la ausencia de una condición o atributo.
- Los falsos positivos (en inglés, FP) son instancias donde el algoritmo indica erróneamente la presencia de una condición o atributo.
- Los falsos negativos (en inglés, FN) son instancias donde el algoritmo indica erróneamente que una determinada condición o atributo está ausente.

Accuracy es una medida de rendimiento que define la proporción de predicciones correctas realizadas por un modelo en comparación con el número total de predicciones. Su definición viene dada por:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.1)$$

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

Tabla 6.3: Matriz de confusión para un problema de clasificación de dos clases.

		Predicción	
		Negativo	Positivo
Realidad	Negativo	TN	FP
	Positivo	FN	TP

Precision es una métrica de rendimiento que evalúa la calidad de las predicciones realizadas por un clasificador. Su definición viene dada por:

$$Precision = \frac{TP}{TP + FP} \quad (6.2)$$

Recall es una métrica que evalúa la capacidad de un clasificador para identificar todas las instancias positivas relevantes. Su definición viene dada por:

$$Recall = \frac{TP}{TP + FN} \quad (6.3)$$

6.6. Resultados y discusión

6.6.1. Agente IoT

El objetivo del agente IoT es monitorizar y recopilar datos de fabricación de la línea de producción de antenas en la plataforma IoT FIWARE. Para ello, el agente interactúa con la API OEE, traduce los datos a formato NGSI-V2 y los ingresa en la plataforma.

La validación del agente IoT es una parte crítica del proceso de desarrollo del sistema. En total se han llevado a cabo 32 *tests*: ocho unitarios del módulo *pilot communication*, cuatro unitarios del módulo *translator*, 15 unitarios del módulo *platform communication* y cinco de aceptación. Todos los *tests* tienen un resultado positivo.

El agente IoT cumple la función de recopilar los datos de la línea de producción en tiempo real y enviarlos a la plataforma IoT para su posterior preprocesado a través del modelo de IA. Operar en tiempo real o casi real es clave para disminuir el tiempo en el que la línea de producción está parada o produce productos defectuosos, reduciendo así los costes de producción. El agente IoT opera en el *edge* para asegurar tanto el procesado en tiempo real como la privacidad y la seguridad de los datos recopilados de la línea de producción. Al

mantener los datos en el *edge*, se reduce el riesgo de violaciones de seguridad o pérdida de datos.

El diseño genérico del agente ha sido utilizado por otros casos de uso del proyecto OPTIMAI para conectar sus dispositivos IoT con la plataforma de la arquitectura. De forma general, el diseño propuesto permite una integración eficiente de nuevos dispositivos y casos de uso sin necesidad de rediseñar por completo el agente. Para ello, únicamente se han aplicado cambios en los módulos *pilot communication* y *translator*.

6.6.2. Servicio cognitivo

El rendimiento de los experimentos propuestos se evalúa mediante las métricas presentadas en la Sección anterior: matriz de confusión, *accuracy*, *precision* y *recall*.

Experimento 1: Aprendizaje no supervisado

En el primer experimento se utilizó el algoritmo *Isolation Forest* (IF) [224] para detectar anomalías en la tasa de recepción de los datos OEE. Al principio, se estima la contaminación en 0.0054 y luego se ajusta para mejorar la precisión del modelo.

Los resultados experimentales muestran que el modelo obtenido no proporciona una precisión adecuada para identificar las anomalías. La Tabla 6.4 indica que, aunque el modelo logra identificar 631 anomalías correctamente, también identifica erróneamente 1386 registros como anómalos y pasa por alto 1398 anomalías. Aunque la precisión del 99.2% puede parecer buena (Tabla 6.7), debido al desequilibrio en el *dataset*, en realidad no lo es (hay pocos registros anómalos). Por lo tanto, se utilizan dos métricas adicionales para analizar el comportamiento del modelo: *precision* y *recall*. Ambas métricas indican un resultado de 0.65, lo que indica que el modelo acierta el 65% de las veces cuando predice una anomalía y que identifica correctamente el 65% de las anomalías.

La Tabla 6.5 y la Tabla 6.6 presentan las matrices de confusión obtenidas de los modelos con un factor de contaminación de 0.005 y 0.0058, respectivamente. El modelo con el factor de contaminación de 0.0058 obtiene mejores resultados, ya que es capaz de detectar más anomalías, 661 en total. En el problema que nos ocupa, los falsos negativos tienen un coste más elevado que los falsos positivos. Por lo tanto, es deseable aumentar la recuperación a expensas de la precisión. Como se ve en la Tabla 6.7, el modelo con un factor de contaminación de 0.0058 también muestra una mejor recuperación (0.66). Aunque es posible mejorar ligeramente los resultados ajustando el factor de contaminación, el algoritmo IF tiene dificultades para identificar anomalías con buena precisión. IF no es

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

capaz de detectar anomalías con una recuperación superior a 0.7. A menudo interpreta erróneamente los acontecimientos periódicos. A menudo interpreta erróneamente como anomalías los acontecimientos periódicos, como la puesta en servicio de las instalaciones después del fin de semana o el inicio de la jornada laboral (FP), o clasifica las anomalías como registros normales (FN).

Tabla 6.4: IF - Matriz de confusión (contamination parameter = 0,0054).

		Predicción	
		Negativo	Positivo
Realidad	Negativo	370 021	1 386
	Positivo	1 398	631

Tabla 6.5: IF - Matriz de confusión (contamination parameter = 0,005).

		Predicción	
		Negativo	Positivo
Realidad	Negativo	370 109	1 298
	Positivo	1 466	563

Tabla 6.6: IF - Matriz de confusión (contamination parameter = 0,0058).

		Predicción	
		Negativo	Positivo
Realidad	Negativo	369 908	1 499
	Positivo	1 368	661

Tabla 6.7: Experimento 1 - Resultados de las métricas de evaluación,

Contamination parameter	Accuracy	Precision	Recall
0.005	0.92	0.65	0.64
0.0054	0.92	0.65	0.65
0.0058	0.92	0.65	0.66

Experimento 2: Aprendizaje supervisado

El segundo experimento se realizó aplicando algoritmos de aprendizaje supervisado: DT, RF, KNN y *Naïve Bayes* (NB) [225]. Durante este experimento,

el *dataset* se dividió en datos de entrenamiento (60%) y datos de prueba (40%). Por este motivo, las matrices de confusión de estos experimentos no contienen el mismo número de registros que en el experimento anterior.

Como se observa en la Tabla 6.8, el rendimiento de NB es peor en comparación con los resultados obtenidos en el experimento anterior, identificando correctamente 141 anomalías. Además, aunque su precisión es mejor (0.74), el *recall* solo alcanza un valor de 0.57 (Tabla 6.12). Por tanto, se descarta el modelo. La Tabla 6.9 contiene los resultados de la matriz de confusión del modelo resultante del entrenamiento del algoritmo DT. Se puede observar que este modelo consigue clasificar correctamente 955 de las 965 anomalías de los datos. Además, no comete ningún error falso positivo (*precision*=1), fallando únicamente al interpretar erróneamente 10 anomalías como datos regulares (*recall*=0.99). Por otra parte, el modelo RF obtiene resultados ligeramente peores que DT. La Tabla 6.10 indica que este modelo tiene 955 verdaderos positivos, 0 falsos positivos y 30 falsos negativos. Esto implica que su *precision* es de 1 y su *recall* de 0.98. Por último, el modelo KNN obtiene casi el mismo resultado que el modelo DT. El modelo KNN identifica satisfactoriamente 954 de 965 anomalías (Tabla 6.11), teniendo una *precision* de 0.99 y un *recall* de 0.99.

Tabla 6.8: NB - Matriz de confusión.

		Predicción	
		Negativo	Positivo
Realidad	Negativo	148 259	150
	Positivo	824	141

Tabla 6.9: DT - Matriz de confusión.

		Predicción	
		Negativo	Positivo
Realidad	Negativo	148 409	0
	Positivo	10	955

Tabla 6.10: RF - Matriz de confusión.

		Predicción	
		Negativo	Positivo
Realidad	Negativo	148 409	0
	Positivo	30	935

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

Tabla 6.11: KNN - Matriz de confusión.

		Predicción	
		Negativo	Positivo
Realidad	Negativo	148 386	23
	Positivo	11	954

Tabla 6.12: Experimento 2 - Resultados de las métricas de evaluación,

Modelo	Accuracy	Precision	Recall
NB	0.9934	0.74	0.57
DT	0.9999	1	0.99
RF	0.9997	1	0.98
KNN	0.997	0.99	0.99

6.7. Conclusiones

En este Capítulo se ha descrito el diseño y desarrollo de una arquitectura AIoT con servicios cognitivos basados en IA. Es importante destacar que esta arquitectura es genérica y no está acoplada a ecosistemas preexistentes. Esta característica garantiza que pueda integrarse de manera versátil en una amplia variedad de entornos y aplicaciones sin imponer restricciones relacionadas con la plataforma IoT. Los usuarios pueden aprovechar su flexibilidad para diseñar sistemas AIoT a medida que se ajusten a sus necesidades específicas, permitiendo la generación de predicciones y el procesamiento cognitivo de datos de manera eficiente y sin fisuras. Esta última transformación de la arquitectura de referencia representa la última contribución a la estandarización y la consolidación de la arquitectura de referencia propuesta en el Capítulo 3.

A diferencia del Capítulo anterior, el componente cognitivo se implementa utilizando componentes *open source* ya existentes. Una de las ventajas es que, puesto que han sido probados y mejorados por la comunidad de desarrolladores durante muchos años, son confiables, estables y escalables. La gran cantidad de información disponible permite ahorrar tiempo y recursos en el desarrollo del componente. Adicionalmente, los componentes seleccionados para la implementación del componente cognitivo permiten la recopilación distribuida de datos, así como la posibilidad de entrenar los modelos de IA de manera distribuida. Sin embargo, una de las principales desventajas es que, al tratarse de componentes ya diseñados, la configuración necesaria para que estos trabajen juntos de forma cohesiva puede consumir mucho tiempo. Puede ser necesario invertir

una cantidad de tiempo y esfuerzo considerable en integrar y sincronizar los componentes para que funcionen juntos de manera efectiva.

La arquitectura ha sido implementada y validada para detectar anomalías en una línea de producción real de industria 4.0 dando lugar a la arquitectura OPTIMAL.

Se ha diseñado y validado un agente IoT genérico, sólido y flexible basado en los principios SOLID. El diseño del agente sirve como plantilla para la implementación de agentes IoT genéricos. Su diseño permite la adición de nuevas funcionalidades de manera ágil y además lo hace flexible ante los cambios. A partir de esta plantilla se ha implementado el agente OEE. Este recopila datos de la línea de producción en tiempo real, los procesa y los envía al *context broker*. A medida que se identifican nuevas necesidades, el agente puede adaptarse fácilmente sin requerir cambios masivos en su estructura. Por ejemplo, gracias a su diseño desacoplado, cambios en la comunicación con los servicios del piloto no afectan al resto de subcomponentes. La adaptabilidad del agente IoT genérico permite crear agentes específicos para estas aplicaciones y muchas otras con relativa facilidad.

Por último, se ha diseñado y validado un servicio cognitivo para detectar de manera automatizada anomalías en la tasa de recepción de los datos. Este servicio permite corregir errores rápidamente, evitar la pérdida de material o la generación de productos defectuosos. Los resultados del entrenamiento del modelo indican que, para la detección de anomalías en el problema de estudio, los métodos de clasificación supervisados de aprendizaje automático son más efectivos que los métodos de aprendizaje no supervisado. Los métodos no supervisados tienen dificultades para distinguir entre anomalías reales y patrones resultantes de la puesta en marcha de la línea de producción tras periodos de inactividad. Entre los modelos obtenidos durante el entrenamiento, el mejor modelo corresponde al algoritmo DT.

CAPÍTULO 6. ARQUITECTURA AIOT PARA LA CREACIÓN DE SERVICIOS COGNITIVOS AVANZADOS

Capítulo 7

Conclusiones

7.1. Conclusiones finales

En esta tesis se han identificado y analizado el estado del arte de las tecnologías IoT e IA. Posteriormente, se han identificado las limitaciones presentes en las arquitecturas AIoT. Tras esto, se ha presentado una arquitectura de referencia multidominio AIoT que combina ambas tecnologías con un enfoque coherente, robusto y basado en estándares ampliamente aceptados. La arquitectura de referencia se ha instanciado y validado en tres casos de uso siguiendo una metodología de validación *use case driven*.

El trabajo presentado en la tesis pone de manifiesto que, a pesar de la abundancia de investigaciones previas que emplean técnicas de IA, ML o DL en el ámbito de IoT, hasta la fecha ha habido escaso esfuerzo en la formulación de un concepto integral y una arquitectura para AIoT. Esto es sumamente importante porque la falta de un marco sólido y una estructura bien definida limita la capacidad para diseñar sistemas más avanzados, eficientes y adaptados a las necesidades cambiantes de la sociedad y la industria. Así, este trabajo contribuye al desarrollo de un marco conceptual y una arquitectura de referencia AIoT. Esta arquitectura constituye una base sólida para abordar los desafíos y las oportunidades que presenta la convergencia de ambas tecnologías.

Del trabajo realizado se extraen las siguientes conclusiones:

- La rápida expansión de IoT está dando lugar a un aumento significativo de los datos, caracterizado por su creciente volumen, variedad y velocidad. Esta explosión de datos presenta tanto retos como oportunidades. La gestión y procesado eficaz de grandes cantidades de datos es un proceso complejo, pero que resulta un recurso muy valioso para el análisis y la

CAPÍTULO 7. CONCLUSIONES

toma de decisiones. Esta capacidad de extraer ideas y tomar decisiones informadas basadas en datos pone de relieve el potencial transformador de aprovechar los datos para obtener información valiosa e impulsar la innovación en diversos ámbitos.

- La integración de capacidades cognitivas en IoT abre la puerta a la adquisición y análisis de información en un nivel nuevo. A pesar de estos avances, la correcta interpretación de los datos y la toma de decisiones basada en ellos siguen siendo un desafío, ya que requiere un equilibrio entre la automatización y la intervención humana para garantizar resultados precisos y confiables. Además, la gestión y procesamiento gran cantidad de datos generados por esta convergencia AIoT plantea un desafío significativo en términos de almacenamiento, análisis y extracción de información de valor. Este aspecto crítico subraya la importancia de desarrollar infraestructuras robustas y estrategias de análisis de datos sofisticadas para maximizar el potencial de esta tecnología.
- En la revisión bibliográfica se han analizado las metodologías, plataformas y tecnologías habilitadoras de IoT y la IA para asentar las bases de la arquitectura propuesta. Primero se proporciona una visión general y se define el concepto de IoT. A continuación, se exploran sus tecnologías de comunicación, arquitecturas, así como los temas de investigación abiertos en la actualidad. Seguidamente, se introduce el concepto de IA. Se exploran los diferentes tipos de aprendizaje, así como los algoritmos y las plataformas que facilitan el desarrollo y la implementación de modelos de IA.
- No se ha encontrado ningún trabajo que haya establecido una arquitectura AIoT de referencia que esté emergiendo como un estándar dominante o que haya sido ampliamente adoptada en los ámbitos académicos e industriales. Aunque existen algunas propuestas de arquitecturas AIoT generales, estas no están alineadas con las recomendaciones de los organismos de estandarización pertinentes.
- En esta tesis doctoral se propone una arquitectura de referencia multidominio AIoT que habilita el uso de servicios cognitivos sobre IoT. El diseño de la arquitectura considera las recomendaciones de la ITU propuestas en UIT-T Y.2066. Además, la arquitectura se alinea a su vez con otras dos arquitecturas de referencia: IIRA y RAMI 4.0. Ambas arquitecturas proporcionan una estructura modular y escalable que permite la interoperabilidad, la seguridad y la integración de dispositivos y servicios. Como consecuencia, la arquitectura presenta una naturaleza versátil y adap-

table, lo que le permite ajustarse a las necesidades y requerimientos de diferentes contextos y dominios.

- Desde la perspectiva de la vista funcional de la arquitectura, esta detalla los elementos funcionales necesarios para habilitar la conectividad y comunicación eficiente entre dispositivos, así como para gestionar, analizar y tomar decisiones basándose en los datos obtenidos. Los niveles de la arquitectura son: nivel de percepción, nivel de *middleware*, nivel de datos semántica, nivel cognitivo de servicios y nivel de negocio. Dentro de estos niveles, se definen diversos componentes que permiten el desarrollo e integración de servicios cognitivos para apoyar esta interpretación avanzada de datos.
- Desde el punto de vista de implementación, la arquitectura ofrece diversas opciones de topología del sistema, que incluyen las basadas en *cloud computing*, *edge computing* o una combinación de ambas. La elección de la topología final dependerá de las características y requisitos específicos del sistema objetivo. La arquitectura proporciona diferentes configuraciones posibles para la distribución de sus componentes según las necesidades y particularidades del sistema objetivo.
- La arquitectura se ha definido, implementado y validado usando una metodología *use case driven* en tres dominios de aplicación: salud, transporte e industria 4.0. Las transformaciones propuestas en la arquitectura contribuyen a su evolución y completa definición mediante un proceso iterativo.
- La primera transformación de la arquitectura permite la creación de servicios comunes en entornos de datos IoT heterogéneos. La implementación y validación de esta transformación se aplica en un caso de uso de salud. AIoTES habilita el desarrollo de un ecosistema de soluciones y servicios de asistencia destinados a cuidar a las personas mayores en su vida diaria. Como colorario se presenta EBASI, una arquitectura que amplía el ecosistema de AIoTES y que está diseñada para identificar patrones de soledad en personas mayores. AIoTES presenta capacidades cognitivas basadas en *big data*, mientras que EBASI utiliza un enfoque de IA débil basado en reglas. Ambas arquitecturas han sido satisfactoriamente validadas con usuarios.
- La validación de la primera transformación contribuye al establecimiento de las bases de la gestión y procesamiento de datos de la arquitectura. Esto es esencial para garantizar la creación de conjuntos de datos homogéneos y

CAPÍTULO 7. CONCLUSIONES

de alta calidad. Los resultados muestran que, pese a que el nivel de interoperabilidad semántica propuesto en AIO TES tiene muchos beneficios, no es ampliamente aceptado por los usuarios debido a su complejidad. Por otro lado, EBASI utiliza exitosamente el modelo de datos de FIWARE, lo que motiva su implementación en el nivel de datos semántica en las siguientes transformaciones.

- La segunda transformación consiste en una arquitectura AIO T para la creación de servicios cognitivos en el ecosistema FIWARE. Esta arquitectura se integra y valida en un caso de uso de transporte. COSIBAS incorpora a un sistema IoT heredado capacidades cognitivas para mejorar la estimación de la hora estimada de llegada (ETA) de los buques. La arquitectura permite la integración de diversas fuentes de datos IoT heterogéneas y utiliza servicios cognitivos de ML para analizar estos datos. La implementación del componente cognitivo ha sido clave para facilitar la incorporación de servicios cognitivos siguiendo la filosofía de FIWARE. La validación del sistema se ha llevado a cabo satisfactoriamente en dos pilotos: el puerto de Valencia y el puerto de Algeciras.
- La validación de la segunda transformación permite definir los requerimientos necesarios para la integración sin fisuras de servicios cognitivos en sistemas IoT. Los requisitos extraídos aseguran que los sistemas IoT sean capaces de aprovechar plenamente las capacidades de la IA. Se define la forma en que los dispositivos IoT se comunican con los servicios cognitivos, cómo se transmiten los datos, cómo se entrenan los modelos y cómo la arquitectura se beneficia de la capacidad de aprendizaje de estos servicios.
- La tercera transformación define una arquitectura AIO T independiente tanto del dominio como de la plataforma IoT. Esta arquitectura se implementa y valida en el contexto de la industria 4.0. OPTIMAI incluye un servicio cognitivo que detecta de forma automática anomalías en la velocidad de recepción de datos en una línea de producción de antenas. En contraste con el caso de uso anterior, el componente cognitivo se implementa de forma genérica utilizando componentes *open source*. El resultado es un *software* complejo, escalable y de calidad, desarrollado en un plazo de tiempo más breve.
- La validación de la tercera transformación permite la creación, desde cero, de un sistema AIO T no limitado por las restricciones de sistemas o entornos preexistentes. Para ello se aplica el conocimiento adquirido en las transformaciones previas. Esta transformación se caracteriza por su

capacidad para adaptarse a una amplia variedad de aplicaciones y entornos. Puede ser implementada en sectores tan diversos como la industria 4.0, la atención médica, la agricultura, la logística y el hogar, entre otros. Asimismo, no está limitada por una plataforma de IoT particular, lo que brinda flexibilidad en la elección de dispositivos y tecnologías de IoT compatibles. Además, garantiza que los sistemas sean escalables y puedan mantenerse actualizados a medida que evoluciona la inteligencia artificial. Los requisitos derivados de esta transformación se utilizan para enriquecer por última vez la especificación de la arquitectura de referencia presentada en el Capítulo 3.

7.2. Líneas futuras de investigación

A continuación, se exponen las potenciales líneas de investigación que se sugieren para ser consideradas y exploradas en trabajos posteriores. Estas líneas representan oportunidades para ampliar el conocimiento y abordar aspectos aún no explorados, con la intención de avanzar en la comprensión y aplicación de los conceptos abordados.

- Explorar la aplicación de la arquitectura de referencia AIIoT propuesta en otros dominios y casos de uso diferentes a los ya abordados. Al adaptar y extender esta arquitectura a nuevos contextos, se podrían definir y desarrollar sistemas innovadores que aprovechen las ventajas de AIIoT en otros ámbitos como la agricultura inteligente, la energía sostenible o las ciudades inteligentes, entre otros. Esta expansión permitiría emplear los conocimientos y experiencias adquiridas, brindando una mayor flexibilidad y aplicabilidad de la arquitectura de referencia AIIoT en un abanico más amplio de dominios de aplicación.
- Validar las capacidades de DL de la arquitectura. Al aprovechar las capacidades de procesamiento avanzadas y los algoritmos sofisticados del DL, se abre la puerta a un sistema AIIoT más inteligente y adaptable. Esta evolución conduce a la integración de áreas como el procesamiento de lenguaje natural o la visión por computadora, allanando el camino para la creación de sistemas AIIoT más sofisticados y eficientes.
- Explorar la integración de aprendizaje federado en la arquitectura. La inferencia o entrenamiento de modelos DL suele ser más costosa en comparación con el ML clásico. Esto se debe a la mayor complejidad y profundidad de estos modelos, que requieren más recursos computacionales y tiempo para procesar. Una estrategia eficaz para reducir estos tiempos

CAPÍTULO 7. CONCLUSIONES

y mejorar la eficiencia es incorporar mecanismos de aprendizaje federado. Estos permiten entrenar y/o inferir modelos de forma distribuida sin que los datos abandonen sus lugares de origen.

- Introducir mecanismos que permitan el reentrenamiento automático de los modelos de IA. Una posible dirección es introducir flujos basados en técnicas de *machine learning operations* para optimizar el ciclo de vida de desarrollo y despliegue de modelos. Esto implica la introducción de nuevos componentes que faciliten la gestión de datos, la experimentación, el monitoreo y la implementación de los modelos de IA en entornos AIoT. Además, la integración de tecnologías como *delta lakes* puede ser beneficiosa para filtrar, almacenar y administrar de manera eficiente los *datasets* necesarios para el entrenamiento de los modelos.
- Continuar trabajando en las especificaciones de los niveles de seguridad y privacidad de la arquitectura. En el presente trabajo solo se ha abordado la definición de estos niveles en la primera transformación de la arquitectura. Una especificación detallada de estos niveles es esencial para preservar la integridad de los datos y garantizar la confianza en el sistema. Sin embargo, debido a su complejidad y alcance, este trabajo podría constituir una tesis completa. Se podría abordar de manera más exhaustiva los desafíos relacionados con la protección de los datos, la autenticación de dispositivos y la gestión de la privacidad, proporcionando una mayor confianza y protección para los usuarios y sus datos. Por ejemplo, *blockchain* podría combinarse con técnicas como el aprendizaje federado para preservar la privacidad.
- Actualizar y adaptar la arquitectura propuesta para aprovechar plenamente las ventajas y capacidades del enfoque *cloud edge continuum*. Esta evolución implica explorar nuevas formas de integrar de manera eficiente y los componentes de computación *cloud* y *edge*, con el objetivo de optimizar el procesamiento de datos y la toma de decisiones en tiempo real. Esto requiere investigar y desarrollar nuevos mecanismos de gestión de datos, algoritmos de análisis en tiempo real, técnicas de distribución de carga y protocolos de comunicación adaptados a esta arquitectura actualizada, buscando maximizar la escalabilidad, la eficiencia energética y la capacidad de respuesta en sistemas AIoT. Esta futura investigación permitirá perfeccionar y adaptar la arquitectura AIoT a nuevas tecnologías emergentes como IA explicable o realidad aumentada.
- Abordar e incorporar en la arquitectura los desafíos éticos que surgen en los sistemas que incorporan IA para garantizar un desarrollo tecnológico

7.2 Líneas futuras de investigación

responsable. La integración efectiva de principios éticos garantizaría la toma de decisiones autónomas, responsables y éticas. Uno de los aspectos que se sugiere abordar es la alineación con normativas y estándares éticos. También es importante garantizar que la arquitectura sea transparente y explicable, para que los usuarios y las partes interesadas puedan comprender cómo se toman las decisiones. Por último, podría ser interesante integrar métodos de evaluación ética de los algoritmos utilizados en la arquitectura, identificando sesgos y posibles problemas éticos en la toma de decisiones.

CAPÍTULO 7. CONCLUSIONES

Apéndice A

Entidades NGSi

Entidad NGSi - Vessel

```
{
  "id": "<Vessel_id>",
  "type": "Vessel",
  "mmsi": {
    "type": "Text",
    "value": "<val_mmsi>",
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Maritime Mobile Service Identity"
      }
    }
  },
  "time": {
    "type": "DateTime",
    "value": <val_time>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Data timestamp AIS format-unix timestamp Human
          readable format-UTC"
      }
    }
  },
  "imo": {
    "type": "Text",
    "value": "<val_imo>",
```

```
"metadata": {
  "description": {
    "type": "Text",
    "value": "IMO ship identification number"
  }
},
"name": {
  "type": "Text",
  "value": "<val_name>",
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Vessel"s name (max.20 chars)"
    }
  }
},
"callSign": {
  "type": "Text",
  "value": "<val_callSign>",
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Vessel"s callsign"
    }
  }
},
"vesselType": {
  "type": "Text",
  "value": "<val_vesselType>",
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Vessel"s type"
    }
  }
},
"dimStern": {
  "type": "Number",
  "value": <val_dimStern>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Dimension to Stern (meters)"
    }
  }
}
```

```
    }
  }
},
"dimBow": {
  "type": "Number",
  "value": <val_dimBow>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Dimension to Bow (meters)"
    }
  }
},
"dimPort": {
  "type": "Number",
  "value": <val_dimPort>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Dimension to Port (meters)"
    }
  }
},
"dimStarboard": {
  "type": "Number",
  "value": <val_dimStarboard>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Dimension to Starboard (meters)"
    }
  }
}
}
```

Entidad NGSÍ - VesselLocation

```
{
  "id": "<vesselLocation_id>",
  "type": "vesselLocation",
  "imo": {
    "type": "Text",
    "value": "<val_imo>",
    "metadata": {
      "description": {
        "type": "Text",
        "value": "IMO ship identification number"
      }
    }
  },
  "location": {
    "type": "geo:json",
    "value": {
      "type": "Point",
      "coordinates": [<longitude>, <latitude>]
    }
  },
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Geographical longitude and latitude AIS format-
in 1/10000 minute i.e. degrees multiplied by 600000
Human readable format-degrees"
    }
  },
  "time": {
    "type": "DateTime",
    "value": <val_time>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Data timestamp AIS format-unix timestamp Human
readable format-UTC"
      }
    }
  },
  "cog": {
    "type": "Number",
    "value": <val_cog>,
    "metadata": {
```

```

    "description": {
      "type": "Text",
      "value": "Course Over Ground AIS format - in 1/10
        degrees i.e. degrees multiplied by 10. COG=3600
        means "not available" Human readable format -
        degrees. COG=360.0 means not available"
    }
  },
  "sog": {
    "type": "Number",
    "value": <va_sog>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Speed Over Ground AIS format - in 1/10
          knots i.e. knots multiplied by 10. SOG=1024 means "
          not available" Human readable format - knots. SOG
          =102.4 means not available"
      }
    }
  },
  "heading": {
    "type": "Number",
    "value": <val_heading>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Current heading of the AIS vessel at the time
          of the last message value in degrees, HEADING=511
          means not available"
      }
    }
  },
  "rot": {
    "type": "Number",
    "value": <val_rot>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "(AIS format only) - Rate of Turn"
      }
    }
  }
},

```

```
"navStat": {
  "type": "Number",
  "value": <val_navStat>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Navigational Status"
    }
  }
},
"draught": {
  "type": "Number",
  "value": <val_draught>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "AIS format - in 1/10 meters i.e. draught
        multiplied by 10. Human readable format - meters"
    }
  }
},
"sourcePort": {
  "type": "Text",
  "value": "<val_sourcePort>",
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Vessel source port"
    }
  }
},
"destinationPort": {
  "type": "Text",
  "value": "<val_destinationPort>",
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Vessel destination port"
    }
  }
},
"eta": {
  "type": "DateTime",
  "value": <val_eta>
```

```
"metadata": {  
  "description": {  
    "type": "Text",  
    "value": "Estimated Time of Arrival"  
  }  
}  
}
```

Entidad NGSI - Weather Marine

```
{
  "id": "<WeatherMarine_id>",
  "type": "WeatherMarine",
  "time": {
    "type": "DateTime",
    "value": <val_time>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "local forecast time"
      }
    }
  },
  "waterTempC": {
    "type": "Number",
    "value": <val_waterTempC>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Water temperature in Celcius"
      }
    }
  },
  "waterTempF": {
    "type": "Number",
    "value": <val_waterTempF>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Water temperature in Fahrenheit"
      }
    }
  },
  "windSpeedKm": {
    "type": "Number",
    "value": <val_windSpeedKm>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Wind speed in kilometers per hour"
      }
    }
  }
},
```

```
"windSpeedMI": {
  "type": "Number",
  "value": <val_windSpeedMI>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Wind speed in miles per hour"
    }
  }
},
"windDir": {
  "type": "Number",
  "value": <val_windDir>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Wind direction in degrees"
    }
  }
},
"precipMm": {
  "type": "Number",
  "value": <val_precipMm>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Precipitation in milimeters"
    }
  }
},
"visibility": {
  "type": "Number",
  "value": <val_visibility>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Visibility in kilometers"
    }
  }
},
"pressure": {
  "type": "Number",
  "value": <val_pressure>,
  "metadata": {
```

```
"description": {
  "type": "Text",
  "value": "Atmospheric pressure in millibars"
}
},
"swellHeightM": {
  "type": "Number",
  "value": <val_swellHeightM>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Swell wave height in metres"
    }
  }
},
"swellHeightFt": {
  "type": "Number",
  "value": <val_swellHeightFt>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Swell wave height in feed "
    }
  }
},
"swellDir": {
  "type": "Number",
  "value": <val_swellDir>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Swell direction in degree"
    }
  }
},
"swellPeriod": {
  "type": "Number",
  "value": <val_swellPeriod>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Swell period in seconds"
    }
  }
}
```

```

    }
  },
  "tideHeight": {
    "type": "Number",
    "value": <val_tideHeigth>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Tide height in meter"
      }
    }
  },
  "tideType": {
    "type": "Text",
    "value": <val_tideType>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Type of tide i.e. High, Low or Normal"
      }
    }
  }
}
}
}

```

Entidad NGSI - VesselETARquest

```

{
  "id": "<VesselETARquest_id>",
  "type": "VesselETARquest",
  "imo": {
    "type": "Text",
    "value": "<val_imo>",
    "metadata": {
      "description": {
        "type": "Text",
        "value": "IMO ship identification number"
      }
    }
  },
  "mmsi": {
    "type": "Text",
    "value": "<val_mmsi>",
    "metadata": {
      "description": {
        "type": "Text",

```

```

    "value": ""
  }
}
},
"time": {
  "type": "DateTime",
  "value": <val_time>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Data timestamp AIS format-unix timestamp Human
readable format-UTC"
    }
  }
},
"dest": {
  "type": "Text",
  "value": "<val_destinationPort>",
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Vessel destination port"
    }
  }
},
"destLocation": {
  "type": "geo:json",
  "value": {
    "type": "Point",
    "coordinates": [<longitude>, <latitude>]
  }
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Geographical longitude and latitud format-in
1/10000 minute i.e. degrees multiplied by 600000
Human readable format-degrees for vessel destination
port"
    }
  }
},
"eta": {
  "type": "DateTime",
  "value": <val_eta>,

```

```
"metadata": {
  "description": {
    "type": "Text",
    "value": "Estimated Time of Arrival"
  }
},
"dataWeather": {
  "type": "Boolean",
  "value": <val_dataWeather>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Establishes whether to include marine
        meteorological data in the calculation of the
        estimated time of arrival"
    }
  }
},
"urlCognitiveService": {
  "type": "String",
  "value": <val_urlCognitiveService>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Url cognitive service"
    }
  }
},
"urlResponseService": {
  "type": "String",
  "value": <val_urlResponseService>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Url response service"
    }
  }
}
}
```

Entidad NGSÍ - VesselETAResponse

```
{
  "id": "<VesselETAResponse_id>",
  "type": "VesselETAResponse",
  "imo": {
    "type": "Text",
    "value": "<val_imo>",
    "metadata": {
      "description": {
        "type": "Text",
        "value": "IMO ship identification number"
      }
    }
  },
  "time": {
    "type": "DateTime",
    "value": <val_time>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Data timestamp AIS format-unix timestamp Human
          readable format-UTC"
      }
    }
  },
  "destinationPort": {
    "type": "Text",
    "value": "<val_destinationPort>",
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Vessel destination port"
      }
    }
  },
  "eta": {
    "type": "DateTime",
    "value": <val_eta>,
    "metadata": {
      "description": {
        "type": "Text",
        "value": "Estimated Time of Arrival Calculate"
      }
    }
  }
}
```

```
},
"devETA": {
  "type": "Number",
  "value": <val_devETA>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Deviation ETA"
    }
  }
},
"longitude": {
  "type": "Number",
  "value": "<val_longitude>",
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Geographical longitude format-in 1/10000
        minute i.e. degrees multiplied by 600000 Human
        readable format-degrees for vessel"
    }
  }
},
"latitude": {
  "type": "Number",
  "value": "<val_latitude>",
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Geographical latitud format-in 1/10000 minute
        i.e. degrees multiplied by 600000 Human readable
        format-degrees for vessel"
    }
  }
},
"messageType": {
  "type": "String",
  "value": <val_messageType>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Message type: Notification, Warning or Alert"
    }
  }
}
```

```
},
"messageText": {
  "type": "String",
  "value": <val_messageText>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Message text"
    }
  }
},
"urlResponseService": {
  "type": "String",
  "value": <val_urlResponseService>,
  "metadata": {
    "description": {
      "type": "Text",
      "value": "Url response service"
    }
  }
}
}
```


Referencias

- [1] A. Papageorgiou, B. Cheng, and E. Kovacs, “Real-time data reduction at the network edge of internet-of-things systems,” in *2015 11th international conference on network and service management (CNSM)*. IEEE, 2015, pp. 284–291.
- [2] L. Wang and R. Ranjan, “Processing distributed internet of things data in clouds,” *IEEE Cloud Computing*, vol. 2, no. 1, pp. 76–80, 2015.
- [3] K. Witkowski, “Internet of things, big data, industry 4.0–innovative solutions in logistics and supply chains management,” *Procedia engineering*, vol. 182, pp. 763–769, 2017.
- [4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [5] A. Arsénio *et al.*, “Internet of intelligent things: Bringing artificial intelligence into things and communication networks,” in *Inter-cooperative collective intelligence: Techniques and applications*. Springer, 2014, pp. 1–37.
- [6] M. S. Mahdavinejad *et al.*, “Machine learning for internet of things data analysis: A survey,” *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, 2018.
- [7] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, “Big data: the management revolution,” *Harvard business review*, vol. 90, no. 10, pp. 60–68, 2012.
- [8] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, “Iot-based big data storage systems in cloud computing: perspectives and challenges,” *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75–87, 2016.

REFERENCIAS

- [9] C. Z. Janikow, “A knowledge-intensive genetic algorithm for supervised learning,” in *Genetic Algorithms for Machine Learning*. Springer, 1993, pp. 33–72.
- [10] W. W. Hsieh, *Machine learning methods in the environmental sciences: Neural networks and kernels*. Cambridge university press, 2009.
- [11] T. G. Dietterich, “Ensemble methods in machine learning,” in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.
- [12] F. Shi *et al.*, “Recent progress on the convergence of the internet of things and artificial intelligence,” *IEEE Network*, vol. 34, no. 5, pp. 8–15, 2020.
- [13] A. A. Wiktorzak, A. Sawicki, T. Kuźmierowski, and G. Rubin, “The cognitive internet of things,” *Polish Journal of Applied Sciences*, vol. 4, no. 3, pp. 85–90, 2019.
- [14] “ChatGPT,” <https://chat.openai.com/>, accessed on: Jul. 31, 2023. 2023.
- [15] “Phind,” <https://www.phind.com/>, accessed on: Jul. 31, 2023. 2022.
- [16] “Google Bard,” <https://bard.google.com/>, accessed on: Jul. 31, 2023. 2022.
- [17] S. Girmay, F. Samsom, and A. M. Khattak, “Ai based login system using facial recognition,” in *2021 5th Cyber Security in Networking Conference (CSNet)*. IEEE, 2021, pp. 107–109.
- [18] K. Shah, B. Arora, A. Shinde, and S. Vaghasia, “Ai in entertainment–movie recommendation using cosine similarity,” in *2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*. IEEE, 2022, pp. 1–4.
- [19] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, “Explainability of deep vision-based autonomous driving systems: Review and challenges,” *International Journal of Computer Vision*, vol. 130, no. 10, pp. 2425–2452, 2022.
- [20] N. Muthukrishnan *et al.*, “Brief history of artificial intelligence,” *Neuroimaging Clinics*, vol. 30, no. 4, pp. 393–399, 2020.
- [21] N. Misra *et al.*, “Iot, big data, and artificial intelligence in agriculture and food industry,” *IEEE Internet of things Journal*, vol. 9, no. 9, pp. 6305–6324, 2020.

- [22] S. Arjunan and P. Sujatha, “Lifetime maximization of wireless sensor network using fuzzy based unequal clustering and aco based routing hybrid protocol,” *Applied Intelligence*, vol. 48, no. 8, pp. 2229–2246, 2018.
- [23] A. Chowdhury, S. A. Raut, and H. S. Narman, “Da-drls: Drift adaptive deep reinforcement learning based scheduling for iot resource management,” *Journal of Network and Computer Applications*, vol. 138, pp. 51–65, 2019.
- [24] A. A. Pise *et al.*, “Enabling artificial intelligence of things (aiot) health-care architectures and listing security issues,” *Computational Intelligence and Neuroscience*, vol. 2022, 2022.
- [25] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, “A survey of recent advances in edge-computing-powered artificial intelligence of things,” *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 849–13 875, 2021.
- [26] M. Castrucci, F. D. Priscoli, A. Pietrabissa, and V. Suraci, “A cognitive future internet architecture,” in *The Future Internet Assembly*. Springer, 2011, pp. 91–102.
- [27] K. Hwang and M. Chen, *Big-data analytics for cloud, IoT and cognitive computing*. John Wiley & Sons, 2017.
- [28] M. U. Farooq, M. Waseem, S. Mazhar, A. Khairi, and T. Kamal, “A review on internet of things (iot),” *International journal of computer applications*, vol. 113, no. 1, pp. 1–7, 2015.
- [29] Y. Perwej, K. Haq, F. Parwej, M. Mundouh, and M. Hassan, “The internet of things (iot) and its application domains,” *International Journal of Computer Applications*, vol. 182, no. 49, pp. 36–49, 2019.
- [30] “statista,” <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide>, accessed on: Sept. 17, 2022. 2022.
- [31] T. D. Fernández, “Taxonomía de transformación digital,” *Revista Cubana de transformación digital*, vol. 1, no. 1, pp. 4–23, 2020.
- [32] M. Ge, H. Bangui, and B. Buhnova, “Big data for internet of things: a survey,” *Future generation computer systems*, vol. 87, pp. 601–614, 2018.
- [33] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, “A review of machine learning and iot in smart transportation,” *Future Internet*, vol. 11, no. 4, p. 94, 2019.

REFERENCIAS

- [34] H. Sequeiros, T. Oliveira, and M. A. Thomas, “The impact of iot smart home services on psychological well-being,” *Information Systems Frontiers*, vol. 24, no. 3, pp. 1009–1026, 2022.
- [35] G. Enabler, “Market pulse report, internet of things (iot). discover key trends and insights on disruptive technologies in iot innovations,” *Accessed: Sep*, vol. 15, p. 2020, 2017.
- [36] D. C. Yacchirema, D. Sarabia-Jácome, C. E. Palau, and M. Esteve, “A smart system for sleep monitoring by integrating iot with big data analytics,” *IEEE Access*, vol. 6, pp. 35 988–36 001, 2018.
- [37] F. Al-Turjman, H. Zahmatkesh, and R. Shahroze, “An overview of security and privacy in smart cities’ iot communications,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, p. e3677, 2022.
- [38] J. C. Stewart, G. A. Davis, and D. A. Igoche, “Ai, iot, and aiot: Definitions and impacts on the artificial intelligence curriculum.” *Issues in Information Systems*, vol. 21, no. 4, 2020.
- [39] R. Minerva, A. Biru, and D. Rotondi, “Towards a definition of the internet of things (iot),” *IEEE Internet Initiative*, vol. 1, no. 1, pp. 1–86, 2015.
- [40] “Y.2060 : Visión general de la internet de las cosas,” <https://www.itu.int/rec/T-REC-Y.2060-201206-I/es>, accessed on: Sept. 17, 2022. 2022.
- [41] “The internet of things at the ietf,” <https://www.ietf.org/topics/iot/>, accessed on: Sept. 17, 2022. 2022.
- [42] R. Davies, “The internet of things: opportunities and challenges,” *EPRS: European Parliamentary Research Service*, 2015.
- [43] F. Xia, L. T. Yang, L. Wang, and A. Vinel, “Internet of things,” *International journal of communication systems*, vol. 25, no. 9, p. 1101, 2012.
- [44] B. Dorsemaine, J.-P. Gaulier, J.-P. Wary, N. Kheir, and P. Urien, “Internet of things: a definition & taxonomy,” in *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*. IEEE, 2015, pp. 72–77.
- [45] S. Haller, S. Karnouskos, and C. Schroth, “The internet of things in an enterprise context,” in *Future internet symposium*. Springer, 2008, pp. 14–28.
- [46] A. C. Sarma and J. Girão, “Identities in the future internet of things,” *Wireless personal communications*, vol. 49, no. 3, pp. 353–363, 2009.

- [47] A. Rayes and S. Salam, “Internet of things (iot) overview,” in *Internet of Things from hype to reality*. Springer, 2019, pp. 1–35.
- [48] L. Da Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
- [49] C. M. Ramya, M. Shanmugaraj, and R. Prabakaran, “Study on zigbee technology,” in *2011 3rd international conference on electronics computer technology*, vol. 6. IEEE, 2011, pp. 297–301.
- [50] Z. Alliance, “Zigbee alliance,” *WPAN industry group*, <http://www.zigbee.org/>. *The industry group responsible for the ZigBee standard and certification*, 2010.
- [51] P. Bhagwat, “Bluetooth: technology for short-range wireless apps,” *IEEE Internet Computing*, vol. 5, no. 3, pp. 96–103, 2001.
- [52] M. B. Yassein, W. Mardini, and A. Khalil, “Smart homes automation using z-wave protocol,” in *2016 International Conference on Engineering & MIS (ICEMIS)*. IEEE, 2016, pp. 1–6.
- [53] V. Coskun, B. Ozdenizci, and K. Ok, “A survey on near field communication (nfc) technology,” *Wireless personal communications*, vol. 71, no. 3, pp. 2259–2294, 2013.
- [54] G. Mulligan, “The 6lowpan architecture,” in *Proceedings of the 4th workshop on Embedded networked sensors*, 2007, pp. 78–82.
- [55] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, “A survey of lorawan for iot: From technology to application,” *Sensors*, vol. 18, no. 11, p. 3995, 2018.
- [56] M. S. Bali, K. Gupta, K. K. Bali, and P. K. Singh, “Towards energy efficient nb-iot: A survey on evaluating its suitability for smart applications,” *Materials Today: Proceedings*, vol. 49, pp. 3227–3234, 2022.
- [57] M. Fahmideh and D. Zowghi, “An exploration of iot platform development,” *Information Systems*, vol. 87, p. 101409, 2020.
- [58] M. A. Rodriguez, L. Cuenca, and A. Ortiz, “Fiware open source standard platform in smart farming-a review,” in *Working Conference on Virtual Enterprises*. Springer, 2018, pp. 581–589.
- [59] “What is the FIWARE Foundation,” <https://www.fiware.org/foundation/>, accessed on: Jul. 1, 2023. 2022.

REFERENCIAS

- [60] “Smart Data Models,” <https://www.fiware.org/smart-data-models/>, accessed on: Jul. 1, 2023. 2022.
- [61] O. C. Foundation, “IoTivity,” accessed on: Sept. 17, 2022. 2020. [Online]. Available: <https://iotivity.org/>
- [62] “Open Connectivity Foundation,” <https://openconnectivity.org/>, accessed on: Jul. 1, 2023. 2022.
- [63] “Linux Foundation,” <https://www.linuxfoundation.org/>, accessed on: Jul. 1, 2023. 2022.
- [64] L. Gürgen, C. Munilla, R. Druilhe, E. Gandrille, and J. Botelho do Nascimento, “sensinact iot platform as a service,” *Enablers for Smart Cities*, pp. 127–147, 2016.
- [65] “Eclipse Foundation,” <https://www.eclipse.org/org/foundation/>, accessed on: Jul. 1, 2023. 2022.
- [66] S. N. Z. Naqvi, S. Yfantidou, and E. Zimányi, “Time series databases and influxdb,” *Studienarbeit, Université Libre de Bruxelles*, vol. 12, 2017.
- [67] B. Rababah, T. Alam, and R. Eskicioglu, “The next generation internet of things architecture towards distributed intelligence: Reviews, applications, and research challenges,” *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 12, no. 2, 2020.
- [68] J. De Loof *et al.*, “Internet of things–architecture iot-a deliverable d1.5–final architectural reference model for the iot v3. 0,” *IoT-A (257521)*, pp. 1–499, 2013.
- [69] M. A. Jabraeil Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh, and F. Norouzi, “Iot architecture,” *Towards the Internet of Things*, pp. 9–31, 2020.
- [70] J. Opara-Martins, R. Sahandi, and F. Tian, “Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective,” *Journal of Cloud Computing*, vol. 5, no. 1, pp. 1–18, 2016.
- [71] S. Soursos *et al.*, “Towards the cross-domain interoperability of iot platforms,” in *2016 European conference on networks and communications (EuCNC)*. IEEE, 2016, pp. 398–402.
- [72] IEEE, “P2413 - standard for an architectural framework for the internet of things (iot),” <https://standards.ieee.org/ieee/2413/6226/>, [Online], accessed on: Sept. 17, 2022. 2022.

- [73] A. Rayes and S. Salam, “The things in iot: Sensors and actuators,” in *Internet of Things From Hype to Reality*. Springer, 2022, pp. 63–82.
- [74] T. Shao, D. Chowdhury, S. S. Gill, and R. Buyya, “Iot-pi: A machine learning-based lightweight framework for cost-effective distributed computing using iot,” *Internet Technology Letters*, vol. 5, no. 3, p. e355, 2022.
- [75] Z. Yang *et al.*, “Study and application on the architecture and key technologies for iot,” in *2011 International Conference on Multimedia Technology*. IEEE, 2011, pp. 747–751.
- [76] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, “Iot middleware: A survey on issues and enabling technologies,” *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1–20, 2016.
- [77] M. Hankel and B. Rexroth, “The reference architectural model industrie 4.0 (rami 4.0),” *ZVEI*, vol. 2, no. 2, pp. 4–9, 2015.
- [78] S.-W. Lin *et al.*, “Industrial internet reference architecture,” *Industrial Internet Consortium (IIC), Tech. Rep.*, 2015.
- [79] G. Fortino *et al.*, “Towards multi-layer interoperability of heterogeneous iot platforms: The inter-iot approach,” in *Integration, interconnection, and interoperability of IoT systems*. Springer, 2018, pp. 199–232.
- [80] AIOTI, “The alliance for internet of things innovation (aioti),” <http://www.aioti.org/>, [Online], accessed on: Sept. 17, 2022. 2022.
- [81] A. Bassi, M. A. Llorente, M. Montesinos, and R. Gravina, “Inter-iot architecture for platform interoperability,” in *Interoperability of Heterogeneous IoT Platforms*. Springer, 2021, pp. 49–94.
- [82] M. B. Alaya, S. Medjiah, T. Monteil, and K. Drira, “Toward semantic interoperability in onem2m architecture,” *IEEE Communications Magazine*, vol. 53, no. 12, pp. 35–41, 2015.
- [83] B. Duncan, A. Happe, and A. Bratterud, “Enterprise iot security and scalability: how unikernels can improve the status quo,” in *2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)*. IEEE, 2016, pp. 292–297.
- [84] P. Ferrari *et al.*, “Delay estimation of industrial iot applications based on messaging protocols,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 9, pp. 2188–2199, 2018.

REFERENCIAS

- [85] P. Zhang, M. Zhou, and G. Fortino, “Security and trust issues in fog computing: A survey,” *Future Generation Computer Systems*, vol. 88, pp. 16–27, 2018.
- [86] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, “A survey on network methodologies for real-time analytics of massive iot data and open research issues,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1457–1477, 2017.
- [87] M. Abomhara and G. M. Køien, “Security and privacy in the internet of things: Current status and open issues,” in *2014 international conference on privacy and security in mobile systems (PRISMS)*. IEEE, 2014, pp. 1–8.
- [88] I. Ali *et al.*, “Systematic literature review on iot-based botnet attack,” *IEEE Access*, vol. 8, pp. 212 220–212 232, 2020.
- [89] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in internet of things: The road ahead,” *Computer networks*, vol. 76, pp. 146–164, 2015.
- [90] Z. Yan, P. Zhang, and A. V. Vasilakos, “A survey on trust management for internet of things,” *Journal of network and computer applications*, vol. 42, pp. 120–134, 2014.
- [91] S. T. Ali, V. Sivaraman, A. Radford, and S. Jha, “A survey of securing networks using software defined networking,” *IEEE transactions on reliability*, vol. 64, no. 3, pp. 1086–1097, 2015.
- [92] R. Kemp, “Fourth industrial revolution,” *The Lawyer*, vol. 31, no. 21, p. 12, 2016.
- [93] S. J. Russell, *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.
- [94] K. Frankish and W. M. Ramsey, *The Cambridge handbook of artificial intelligence*. Cambridge University Press, 2014.
- [95] M. de Asuntos Económicos y Transformación Digital, “España digital 2026,” <https://espanadigital.gob.es/>, [Online], accessed on: Sept. 17, 2022. 2022.
- [96] A. L. Samuel, “Machine learning,” *The Technology Review*, vol. 62, no. 1, pp. 42–45, 1959.

- [97] E. G. Dada *et al.*, “Machine learning for email spam filtering: review, approaches and open research problems,” *Heliyon*, vol. 5, no. 6, p. e01802, 2019.
- [98] H. Liu and B. Lang, “Machine learning and deep learning methods for intrusion detection systems: A survey,” *applied sciences*, vol. 9, no. 20, p. 4396, 2019.
- [99] R. Sharma, B. Kaushik, and N. Gondhi, “Character recognition using machine learning and deep learning-a survey,” in *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*. IEEE, 2020, pp. 341–345.
- [100] S.-G. Kim *et al.*, “Ai for design: Virtual design assistant,” *CIRP Annals*, vol. 68, no. 1, pp. 141–144, 2019.
- [101] A. Nawrocka, A. Kot, and M. Nawrocki, “Application of machine learning in recommendation systems,” in *2018 19th International Carpathian Control Conference (ICCC)*. IEEE, 2018, pp. 328–331.
- [102] C. McClelland, “The difference between artificial intelligence, machine learning, and deep learning,” *Retrieved February*, vol. 18, p. 2021, 2017.
- [103] H. Choi, “Deep learning in nuclear medicine and molecular imaging: current perspectives and future directions,” *Nuclear medicine and molecular imaging*, vol. 52, no. 2, pp. 109–118, 2018.
- [104] IEEE-USA, “Ieee-usa position statement. artificial intelligence research, development and regulation,” <http://globalpolicy.ieee.org/wp-content/uploads/2017/10/IEEE17003.pdf>, [Online], accessed on: Sept. 17, 2022. 2017.
- [105] ETSI, “Etsi gr eni 004 v2.2.1,” https://www.etsi.org/deliver/etsi_gr/ENI/001_099/004/02.02.01_60/gr_ENI004v020201p.pdf, [Online], accessed on: Sept. 17, 2022. 2021.
- [106] E. Commission *et al.*, “A definition of ai: main capabilities and disciplines,” *Available online at. Accessed on*, vol. 10, 2019.
- [107] S. L. Andresen, “John mccarthy: father of ai,” *IEEE Intelligent Systems*, vol. 17, no. 5, pp. 84–85, 2002.
- [108] H. John, “Inteligencia artificial,” *Alianza Infomatica*, vol. 2, no. XXI, 1985.

REFERENCIAS

- [109] R. Bellman, *An introduction to artificial intelligence: can computers think?* Thomson Course Technology, 1978.
- [110] E. Charniak and D. McDermott, “Introduction to artificial intelligence. addison,” *Reading, MA*, 1985.
- [111] T. M. Mitchell and T. M. Mitchell, *Machine learning*. McGraw-hill New York, 1997, vol. 1, no. 9.
- [112] A. Samuel, “Eight-move opening utilizing generalization learning,” *IBM J*, vol. 3, no. 3, pp. 210–229, 1959.
- [113] A. L. Samuel, “Some studies in machine learning using the game of checkers. ii—recent progress,” *IBM Journal of research and development*, vol. 11, no. 6, pp. 601–617, 1967.
- [114] M. Abadi *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.
- [115] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.
- [116] L. Etaati, “Deep learning tools with cognitive toolkit (cntk),” in *Machine Learning with Microsoft Technologies*. Springer, 2019, pp. 287–302.
- [117] E. Cengil, A. Çınar, and E. Özbay, “Image classification with caffe deep learning framework,” in *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2017, pp. 440–444.
- [118] J. Hao and T. K. Ho, “Machine learning made easy: a review of scikit-learn package in python programming language,” *Journal of Educational and Behavioral Statistics*, vol. 44, no. 3, pp. 348–361, 2019.
- [119] “Scientific python,” <https://scientific-python.org/>, accessed on: Sept. 17, 2022. 2022.
- [120] D. Rao and B. McMahan, *Natural language processing with PyTorch: build intelligent language applications using deep learning*. .o’Reilly Media, Inc.”, 2019.
- [121] R. Collobert, S. Bengio, and J. Mariéthoz, “Torch: a modular machine learning software library,” Idiap, Tech. Rep., 2002.
- [122] S. Lang, F. Bravo-Marquez, C. Beckham, M. Hall, and E. Frank, “Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j,” *Knowledge-Based Systems*, vol. 178, pp. 48–50, 2019.

- [123] J. Brownlee, *Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras*. Machine Learning Mastery, 2016.
- [124] Y. Chen, J. E. Argentinis, and G. Weber, “Ibm watson: how cognitive computing can be applied to big data challenges in life sciences research,” *Clinical therapeutics*, vol. 38, no. 4, pp. 688–701, 2016.
- [125] U. Cortés, A. Cortés, and C. Barrué, “Trustworthy ai. the ai4eu approach,” *Proceedings of Science*, 2019.
- [126] E. Bisong, “An overview of google cloud platform services,” *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 7–10, 2019.
- [127] J. Soh, M. Copeland, A. Puca, and M. Harris, *Microsoft Azure: Planning, Deploying, and Managing the Cloud*. Springer, 2020.
- [128] M. Tawalbeh, M. Quwaidar, and A. T. Lo’ai, “Authorization model for iot healthcare systems: case study,” in *2020 11th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2020, pp. 337–342.
- [129] S. Armstrong, A. Sandberg, and N. Bostrom, “Thinking inside the box: Controlling and using an oracle ai,” *Minds and Machines*, vol. 22, no. 4, pp. 299–324, 2012.
- [130] “Google Vision AI,” <https://cloud.google.com/vision>, accessed on: Jul. 1, 2023. 2022.
- [131] “Azure Computer Vision,” <https://azure.microsoft.com/es-es/products/cognitive-services/computer-vision>, accessed on: Jul. 1, 2023. 2022.
- [132] “IBM Watson Natural Language Understanding,” <https://www.ibm.com/products/natural-language-understanding>, accessed on: Jul. 1, 2023. 2022.
- [133] “Lexalytics,” <https://www.lexalytics.com/>, accessed on: Jul. 1, 2023. 2022.
- [134] “Google Translate API,” <https://cloud.google.com/translate/>, accessed on: Jul. 1, 2023. 2022.
- [135] E. Strickland, “Ibm watson, heal thyself: How ibm overpromised and underdelivered on ai health care,” *IEEE Spectrum*, vol. 56, no. 4, pp. 24–31, 2019.

REFERENCIAS

- [136] “Aidoc,” <https://www.aidoc.com/>, accessed on: Jul. 1, 2023. 2022.
- [137] ITU, “Recomendación y.2066,” <https://www.itu.int/rec/T-REC-Y.2066-201406-I/es>, [Online], accessed on: Sept. 17, 2022. 2014.
- [138] R. B. Miller, “Response time in man-computer conversational transactions,” in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, 1968, pp. 267–277.
- [139] N. S. Lohitha and M. Pounambal, “Integrated publish/subscribe and push-pull method for cloud based iot framework for real time data processing,” *Measurement: Sensors*, vol. 27, p. 100699, 2023.
- [140] K. Jeffay, “The real-time producer/consumer paradigm: A paradigm for the construction of efficient, predictable real-time systems,” in *Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing: state of the art and practice*, 1993, pp. 796–804.
- [141] W. H. Organization, *Decade of healthy ageing: baseline report: summary*. World Health Organization, 2021.
- [142] G. Anderson and J. R. Knickman, “Changing the chronic care system to meet people’s needs,” *Health Affairs*, vol. 20, no. 6, pp. 146–160, 2001.
- [143] S. Martínez Pizarro, “Síndrome del cuidador quemado,” *Revista clínica de medicina de familia*, vol. 13, no. 1, pp. 97–100, 2020.
- [144] “ACTIVAGE,” <http://www.activageproject.eu/>, accessed on: Sept. 17, 2022. 2021.
- [145] R. Atmoko, R. Riantini, and M. Hasin, “Iot real time data acquisition using mqtt protocol,” in *Journal of Physics: Conference Series*, vol. 853, no. 1. IOP Publishing, 2017, p. 012003.
- [146] D. Tracey and C. Sreenan, “Oma lwm2m in a holistic architecture for the internet of things,” in *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*. IEEE, 2017, pp. 198–203.
- [147] E. FIWARE Foundation, “FIWARE,” accessed on: Sept. 17, 2022. 2020. [Online]. Available: <https://www.fiware.org/>
- [148] Minsait, “Onesait,” accessed on: Sept. 17, 2022. 2020. [Online]. Available: <https://www.onesait.com/platform>
- [149] UniversAAL, “universAAL IoT,” accessed on: Sept. 17, 2022. 2020. [Online]. Available: <https://www.universaal.info/>

- [150] E. Foundation, “Eclipse sensiNact,” accessed on: Sept. 17, 2022. 2019. [Online]. Available: <https://wiki.eclipse.org/SensiNact>
- [151] O. Project, “OpenIoT,” accessed on: Sept. 17, 2022. 2019. [Online]. Available: <http://www.openiot.eu/>
- [152] D. Yacchirema *et al.*, “Interoperability of IoT Platforms applied to the transport and logistics domain,” 2019.
- [153] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szymeja, and K. Wasielewska, “Towards semantic interoperability between Internet of Things platforms,” in *Integration, interconnection, and interoperability of iot systems*. Springer, 2018, pp. 103–127.
- [154] A. Project, “ACTIVAGE Core Ontology Documentation,” 2019. [Online]. Available: <http://docs.activageproject.eu/data-model/>
- [155] P. Szymeja, “Generic Ontology for IoT Platforms,” 2018. [Online]. Available: <https://docs.inter-iot.eu/ontology/>
- [156] M. Ganzha *et al.*, “From implicit semantics towards ontologies—practical considerations from the inter-iot perspective,” in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2017, pp. 59–64.
- [157] M. Marjani *et al.*, “Big iot data analytics: architecture, opportunities, and open research challenges,” *ieee access*, vol. 5, pp. 5247–5261, 2017.
- [158] M. Serrano, A. Gyrard, E. Tragos, and H. Nguyen, “Fiestaiot project: Federated interoperable semantic iot/cloud testbeds and applications,” in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 425–426.
- [159] A. Kazmi, Z. Jan, A. Zappa, and M. Serrano, “Overcoming the heterogeneity in the internet of things for smart cities,” in *Interoperability and Open-Source Solutions for the Internet of Things: Second International Workshop, InterOSS-IoT 2016, Held in Conjunction with IoT 2016, Stuttgart, Germany, November 7, 2016, Invited Papers 2*. Springer, 2017, pp. 20–35.
- [160] J. Soldatos *et al.*, “Openiot: Open source internet-of-things in the cloud,” in *Interoperability and Open-Source Solutions for the Internet of Things: International Workshop, FP7 OpenIoT Project, Held in Conjunction with SoftCOM 2014, Split, Croatia, September 18, 2014, Invited Papers*. Springer, 2015, pp. 13–25.

REFERENCIAS

- [161] “Docker,” <https://www.docker.com/>, accessed on: Jul. 1, 2023. 2023.
- [162] O. Foundation, “OpenID Connect 1.0,” 2020. [Online]. Available: <https://openid.net/connect/>
- [163] U. I. for Statistics, *International standard classification of education: ISCED 2011*. UNESCO Institute for Statistics Montreal, 2012.
- [164] U. I. for Statistics, “International standard classification of education fields of education and training 2013 (isced-f 2013): Detailed field descriptions.” 2015.
- [165] ACTIVAGE Project, “ACTIVAGE Marketplace,” 2019. [Online]. Available: <https://marketplace.activage.iti.gr>
- [166] “EBASI,” <https://diatomic.eu/index.php/pull-experiments/ebasi/>, accessed on: Sept. 17, 2022. 2022.
- [167] M. Cattan, M. White, J. Bond, and A. Learmouth, “Preventing social isolation and loneliness among older people: a systematic review of health promotion interventions,” *Ageing & society*, vol. 25, no. 1, pp. 41–67, 2005.
- [168] G. C. Wenger and V. Burholt, “Changes in levels of social isolation and loneliness among older people in a rural area: a twenty-year longitudinal study,” *Canadian Journal on Aging/la revue canadienne du vieillissement*, vol. 23, no. 2, pp. 115–127, 2004.
- [169] J. S. House, K. R. Landis, and D. Umberson, “Social relationships and health,” *Science*, vol. 241, no. 4865, pp. 540–545, 1988.
- [170] J. B. Turner and R. J. Turner, “Social relations, social integration, and social support,” *Handbook of the sociology of mental health*, pp. 341–356, 2013.
- [171] J. Holt-Lunstad, “Social isolation and health,” *Health affairs brief*, 2020.
- [172] K. Uemura, M. Yamada, and H. Okamoto, “The effectiveness of an active learning program in promoting a healthy lifestyle among older adults with low health literacy: a randomized controlled trial,” *Gerontology*, vol. 67, no. 1, pp. 25–35, 2021.
- [173] L. Vuegen, B. Van Den Broeck, P. Karsmakers, B. Vanrumste *et al.*, “Automatic monitoring of activities of daily living based on real-life acoustic

- sensor data: A preliminary study,” in *Fourth workshop on speech and language processing for assistive technologies (SLPAT): Proceedings*. Association for Computational Linguistics (ACL); Stroudsburg, 2013, pp. 113–118.
- [174] FIWARE, “Fiware-ngsi v2 specification,” <http://fiware.github.io/specifications/ngsiv2/stable/>, [Online], accessed on: Sept. 17, 2022. 2022.
- [175] “Sensor ultrasonido hc-sr04,” <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>, accessed on: Sept. 17, 2022. 2022.
- [176] “Módulo sensor de luz digital bh1750,” <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>, accessed on: Sept. 17, 2022. 2022.
- [177] “Leanstar md30-60@20kg,” https://www.lcsc.com/product-detail/Force-Sensors_LEANSTAR-MD30-60-20kg_C406745.html, accessed on: Sept. 17, 2022. 2022.
- [178] “<https://www.az-delivery.de/es/products/regen-sensor-modul>,” <https://www.az-delivery.de/es/products/regen-sensor-modul>, accessed on: Sept. 17, 2022. 2022.
- [179] J. Pino-Ortega, C. D. Gómez-Carmona, and M. Rico-González, “Accuracy of xiaomi mi band 2.0, 3.0 and 4.0 to measure step count and distance for physical activity and healthcare in adults over 65 years,” *Gait & Posture*, vol. 87, pp. 6–10, 2021.
- [180] V. Pardeshi, S. Sagar, S. Murmurwar, and P. Hage, “Health monitoring systems using iot and raspberry pi—a review,” in *2017 international conference on innovative mechanisms for industry applications (ICIMIA)*. IEEE, 2017, pp. 134–137.
- [181] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, “Device-to-device communications with wi-fi direct: overview and experimentation,” *IEEE wireless communications*, vol. 20, no. 3, pp. 96–104, 2013.
- [182] P. Srivastava, M. Bajaj, and A. S. Rana, “Overview of esp8266 wi-fi module based smart irrigation system using iot,” in *2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*. IEEE, 2018, pp. 1–5.

REFERENCIAS

- [183] “Anker bluethooth speaker,” <https://www.ankerkw.com/product/anker-soundcore-mini-3-pro-portable-bluetooth-speaker-black/>, accessed on: Sept. 17, 2022. 2022.
- [184] “Movo mc 1000,” <https://www.movophoto.com/products/mc1000-usb-conference-microphone>, accessed on: Sept. 17, 2022. 2022.
- [185] C. Walls, “Spring in action,” 2022.
- [186] H. M. Deitel *et al.*, “Java: como programar,” 2016.
- [187] K. Banker, D. Garrett, P. Bakkum, and S. Verch, “Mongodb in action: covers mongodb version 3.0,” 2016.
- [188] L. Soberón and J. Jesús, “Análisis comparativo de sistemas gestores de bases de datos postgresql y mysql en procesos crud,” 2020.
- [189] M. Wilde *et al.*, “Swift: A language for distributed parallel scripting,” *Parallel Computing*, vol. 37, no. 9, pp. 633–652, 2011.
- [190] R. Fielding *et al.*, “Hypertext transfer protocol–http/1.1,” World Wide Web, Tech. Rep., 1999.
- [191] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, “Foundations of json schema,” in *Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 263–273.
- [192] “Fiware catalog,” <https://www.fiware.org/catalogue/>, accessed on: Sept. 17, 2022. 2022.
- [193] “Context adapter,” <https://github.com/telefonicaid/context-adapter>, accessed on: Sept. 17, 2022. 2022.
- [194] “Context event processing (perseo),” <https://thinking-cities.readthedocs.io/en/release-v4.1/cep/>, accessed on: Sept. 17, 2022. 2022.
- [195] FIWARE, “Orion context broker,” https://fiware-training.readthedocs.io/es_MX/latest/ecosistemaFIWARE/ocb/, [Online], accessed on: Sept. 17, 2022. 2022.
- [196] “Fiware sth comet,” <https://fiware-sth-comet.readthedocs.io/en/latest/>, accessed on: Sept. 17, 2022. 2022.
- [197] R. Gómez, A. Camarero, and R. Molina, “Development of a vessel-performance forecasting system: Methodological framework and case study,” *Journal of Waterway, Port, Coastal, and Ocean Engineering*, vol. 142, no. 2, p. 04015016, 2016.

- [198] Prodevelop, “Posidonia port cdm,” <https://www.prodevelop.es/puertos/posidonia/posidonia-portcdm-castellano>, [Online], accessed on: Sept. 17, 2022. 2022.
- [199] R. Meijer, “Eta prediction: Predicting the eta of a container vessel based on route identification using ais data,” 2017.
- [200] I. Parolas, “Eta prediction for containerships at the port of rotterdam using machine learning techniques,” 2016.
- [201] A. Alessandrini, F. Mazzarella, and M. Vespe, “Estimated time of arrival using historical vessel tracking data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 7–15, 2018.
- [202] A. Dobrkovic, M.-E. Iacob, J. v. Hillegersberg, M. R. Mes, and M. Glandrup, “Towards an approach for long term ais-based prediction of vessel arrival times,” in *Logistics and Supply Chain Innovation*. Springer, 2016, pp. 281–294.
- [203] C. Pani, T. Vanelslander, G. Fancello, and M. Cannas, “Prediction of late/early arrivals in container terminals-a qualitative approach,” 2015.
- [204] O. Bodunov, F. Schmidt, A. Martin, A. Brito, and C. Fetzer, “Real-time destination and eta prediction for maritime traffic,” in *Proceedings of the 12th ACM international conference on distributed and event-based systems*, 2018, pp. 198–201.
- [205] AISHub, “Aishub ais data exchange,” <http://www.aishub.net>, [Online], accessed on: Sept. 17, 2022. 2022.
- [206] “The nmea 0183 protocol,” <https://www.tronico.fi/OH6NT/docs/NMEA0183.pdf>, accessed on: Sept. 17, 2022. 2022.
- [207] “COSIBAS cognitive services for iot-based scenarios,” <https://itea4.org/project/cosibas.html>, accessed on: Sept. 17, 2022. 2022.
- [208] S. Chazallet, *Python 3: los fundamentos del lenguaje*. Ediciones ENI, 2016.
- [209] “Pickle - python object serialization,” <https://docs.python.org/3/library/pickle.html>, accessed on: Sept. 17, 2022. 2022.
- [210] W. McKinney *et al.*, “pandas: a foundational python library for data analysis and statistics,” *Python for high performance and scientific computing*, vol. 14, no. 9, pp. 1–9, 2011.

REFERENCIAS

- [211] C. R. Harris *et al.*, “Array programming with numpy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.
- [212] E. Bisong, “Matplotlib and seaborn,” in *Building machine learning and deep learning models on google cloud platform*. Springer, 2019, pp. 151–165.
- [213] H. Liu and R. Setiono, “Incremental feature selection,” *Applied Intelligence*, vol. 9, no. 3, pp. 217–230, 1998.
- [214] T. Eriksen, G. Høye, B. Narheim, and B. J. Meland, “Maritime traffic monitoring using a space-based ais receiver,” *Acta Astronautica*, vol. 58, no. 10, pp. 537–549, 2006.
- [215] W. W. Online, “Marine, sailing and surfing weather api,” <https://www.worldweatheronline.com/developer/api/docs/marine-weather-api.aspx>, [Online], accessed on: Sept. 17, 2022. 2022.
- [216] K. Beck, *Test driven development: By example*. Addison-Wesley Professional, 2022.
- [217] “Interface with IoT, Robots and third-party systems,” <https://www.fiware.org/catalogue/>, accessed on: Sept. 17, 2022. 2022.
- [218] “Apache Airflow,” <https://airflow.apache.org/>, accessed on: Sept. 17, 2022. 2023.
- [219] “Apache Hadoop,” <https://hadoop.apache.org/>, accessed on: Sept. 17, 2022. 2023.
- [220] “Apache Spark,” <https://spark.apache.org/>, accessed on: Sept. 17, 2022. 2023.
- [221] M. Ghobakhloo, “Industry 4.0, digitization, and opportunities for sustainability,” *Journal of cleaner production*, vol. 252, p. 119869, 2020.
- [222] V. K. Madasu, T. V. S. N. Venna, and T. Eltaeib, “Solid principles in software architecture and introduction to resm concept in oop,” *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, vol. 2, no. 2, pp. 3159–0040, 2015.
- [223] “pyngsi,” <https://pypi.org/project/pyngsi/>, accessed on: Sept. 17, 2022. 2022.
- [224] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

- [225] I. Rish *et al.*, “An empirical study of the naive bayes classifier,” in *IJ-CAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, 2001, pp. 41–46.