UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Análisis de frameworks y soluciones para la IoT

Trabajo Fin de Máster

Máster Universitario en Ingeniería Informática

AUTOR/A: Gabaldón Ibáñez, Miguel Ángel

Tutor/a: Fons Cors, Joan Josep

CURSO ACADÉMICO: 2022/2023

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

Análisis de frameworks y soluciones para la IoT

Trabajo Final de Master

Máster Universitario en Ingeniería Informática

Autor: Miguel Ángel Gabaldón Ibáñez

Tutor: Joan Fons Cors

Curso 2022-2023

Resumen

Debido al aumento significativo en el número de dispositivos conectados y su aplicación en una amplia gama de dominios, se han desarrollado múltiples soluciones basadas en IoT que requieren de estrategias para su interconexión y gestión. Sin embargo, la elección de la plataforma o framework correcto puede ser un desafío dada la gran cantidad de opciones disponibles, cada una con sus propias funcionalidades, niveles de seguridad y soporte para computación en la nube, entre otros factores.

Este proyecto adopta un enfoque exploratorio para identificar, organizar y sintetizar las características y funcionalidades de las diferentes soluciones industriales, resultando en la creación de un catálogo de características básicas y avanzadas. Este catálogo, presentado en formato de tablas, tiene como objetivo ayudar a las organizaciones a seleccionar la plataforma o framework más adecuado para sus necesidades específicas de IoT.

Seguidamente, se desarrollan escenarios ilustrativos que representan diferentes situaciones o contextos, en los cuales se argumenta cuáles plataformas o frameworks son las más adecuadas según las necesidades de cada caso, siendo uno de estos una solución IoT destinada a la digitalización integral de una vivienda, con un enfoque particular en las necesidades de las personas con diabetes, a fin de convertirla en una '*Smart house*'.

El presente estudio proporciona un marco comprensible y práctico para la selección y aplicación de soluciones IoT, contribuyendo a una implementación más efectiva y adecuada de estas tecnologías en diversos ámbitos.

Palabras clave: Internet de las Cosas (IoT), Internet de las Cosas Industrial (IIoT), Plataforma en la nube, Gestión de las cosas, Catálogo de características de las plataformas IoT.

Resum

El present Treball Final de Màster (TFM) es centra en la investigació, anàlisi i aplicació de diferents plataformes i frameworks per a la Internet de les Coses (IoT). A causa de l'increment significatiu en el nombre de dispositius connectats i la seua aplicació en una àmplia gamma de dominis, s'han desenvolupat múltiples solucions basades en IoT que requereixen d'estratègies per a la seua interconnexió i gestió. No obstant això, la tria de la plataforma o framework correcte pot ser un desafiament donada la gran quantitat d'opcions disponibles, cada una amb les seues pròpies funcionalitats, nivells de seguretat i suport per a computació en el núvol, entre altres factors.

Aquest projecte adopta un enfocament exploratori per a identificar, organitzar i sintetitzar les característiques i funcionalitats de les diferents solucions industrials, resultant en la creació d'un catàleg de característiques bàsiques i avançades. Aquest catàleg, presentat en format de taules, té com a objectiu ajudar a les organitzacions a seleccionar la plataforma o framework més adequat per a les seues necessitats específiques d'IoT.

A continuació es desenvolupen escenaris il·lustratius que representen diferents situacions o contextos, en els quals s'argumenta quines plataformes o frameworks són les més adequades segons les necessitats de cada cas, sent una d'estes una solució IoT destinada a la digitalització integral d'una vivenda, amb un enfocament particular en les necessitats de les persones amb diabetis, amb l'objectiu de convertir-la en una 'Smart house'.

El present estudi proporciona un marc comprensible i pràctic per a la selecció i aplicació de solucions IoT, contribuint a una implementació més efectiva i adequada d'estes tecnologies en diversos àmbits."

Paraules clau: Internet de les Coses (IoT), Internet de les Coses Industrial (IIoT), Plataforma en el núvol, Gestió de les coses, Catàleg de característiques de les plataformes IoT.

Abstract

This Master Thesis (TFM) focuses on the research, analysis and application of different platforms and frameworks for the Internet of Things (IoT). Due to the significant increase in the number of connected devices and their application in a wide range of domains, multiple IoT-based solutions have been developed that require strategies for their interconnection and management. However, choosing the right platform or framework can be a challenge given the large number of options available, each with its own functionalities, security levels and support for cloud computing, among other factors.

This project adopts an exploratory approach to identify, organize and synthesize the features and functionalities of different industrial solutions, resulting in the creation of a catalog of basic and advanced features. This catalog, presented in table format, aims to help organizations select the most suitable platform or framework for their specific IoT needs.

Thereafter, illustrative scenarios are developed that represent different situations or contexts, in which it is argued which platforms or frameworks are the most appropriate according to the needs of each case, one of these being an IoT solution aimed at the digitization of a home, with a particular focus on the needs of people with diabetes, in order to turn it into a 'Smart house'.

This study provides a comprehensive and practical framework for the selection and application of IoT solutions, contributing to a more effective and appropriate implementation of these technologies in various areas.

Keywords: Internet of Things (IoT), Industrial Internet of Things (IIoT), Cloud platform, Things management, IoT platforms features catalog.

Tabla de contenidos

Índice de Tablas de Características

Índice de Apéndice

1. Introducción

1.1. Motivación

En la actualidad y gracias a la capacidad de desarrollar pequeños dispositivos con capacidades computacionales y conectados, está consiguiendo que se puedan desplegar, con una relativa facilidad y a un coste muy reducido, soluciones basadas en sensores, actuadores y controladores, que interactúan con el mundo físico. Además, su aplicabilidad es muy alta a prácticamente cualquier dominio, por lo que se está viendo un alto impacto de uso en esta última década.

La Internet de las Cosas propone una estrategia para lidiar con estos dispositivos, y tratar de coordinar el mundo físico (el que monitorizan, sobre el que actúan y controlan estos dispositivos) con el mundo digital (el de los procesos software, las aplicaciones informáticas). La IoT se basa en la definición de estrategias, abiertas y extensibles, para que estos dispositivos puedan interconectarse 'digitalmente' entre sí (en comunicaciones llamadas M2M o 'machine-to-machine') y con los sistemas informáticos, a través de APIs de interoperabilidad de diferentes tipos.

Una solución basada en la IoT es una solución que combina los servicios y dispositivos computacionales y de comunicaciones tradicionales (para construir software distribuido), pero, además, ahora cuenta con un gran número de pequeños dispositivos (muchas veces con escasas capacidades de cómputo o de comunicaciones, entre otras), construyendo soluciones más complejas, heterogéneas, dinámicas y más propensas a fallos.

En este contexto, empezaron a desarrollarse frameworks y plataformas IoT para tratar de facilitar el diseño, desarrollo e implantación de este tipo de soluciones. Estas plataformas deben ofrecer servicios para exponer los propios dispositivos y sus funciones, ofreciendo mecanismos de comunicación entre ellos y los sistemas, pero también otras funcionalidades relacionadas con aspectos de gestión de la infraestructura de dispositivos, seguridad, gestión de conflictos, actualizaciones, etc.

En la actualidad existen un gran número de estas soluciones a nivel industrial, aportando cada una su visión sobre el desarrollo, implantación y gestión de este tipo de soluciones. Entonces, si una empresa, institución u organización quiere desarrollar una solución de IoT, ¿cuál de estas soluciones debería utilizar? ¿Ofrecen todas los mismos constructores y tipos de herramientas? ¿Ofrecen los mismos niveles de seguridad? ¿Son open-source o funcionan a través de licencias? ¿Qué soporte ofrecen con respecto a la computación en la nube?, entre otras muchas dudas.

Este Trabajo de Final de Master se plantea con un objetivo exploratorio y de análisis, en el que se inicialmente se realizará una búsqueda de las soluciones industriales recopilando y organizando características y funcionalidades para cada solución. En una segunda fase, se tratará

de organizar estas características y sintetizarlas, con el objetivo de proponer un catálogo de características básicas y avanzadas que puedan ofrecer estos entornos. Este catálogo se presentará en formato de tabla para facilitar la visualización y el entendimiento de la información Este catálogo de características debe servir para definir los aspectos clave que requiere nuestra aplicación y por tanto que define qué tipo de frameworks o plataformas serían las más adecuadas.

En la tercera fase de este proyecto, procederemos a elaborar una serie de escenarios ilustrativos, cuidadosamente seleccionados, que representarán diversas situaciones o contextos en las que, basándonos en los datos y análisis realizados previamente, argumentaremos en función de los requerimientos específicos de cada situación cuales son los frameworks o plataformas que mejor se adaptan para conseguir abarcar todas sus necesidades. De esta manera, esta fase proporcionará un marco práctico y contextual para comprender mejor la aplicabilidad y utilidad de las diferentes características en situaciones reales.

Por último, abordaremos la selección de nuestra opción preferida para el desarrollo de una solución IoT. Al igual que en la etapa previa, detallaremos las características fundamentales que buscamos en la plataforma seleccionada. Nuestra elección se orientará hacia una solución que permite la informatización integral de una vivienda, con el objetivo de convertirla en una *'Smart house'* diseñada específicamente para las necesidades de personas con diabetes.

1.2. Objetivos

El propósito principal de este Trabajo Final de Máster es llevar a cabo un profundo estudio y análisis sobre las soluciones IoT presentes en el mercado y sus múltiples características, con el fin de crear un marco de entendimiento y evaluación de las soluciones existentes que faciliten la toma de decisiones a la hora de elegir una plataforma IoT para cada proyecto. A continuación, se detallan los objetivos específicos:

- Análisis y clasificación de las soluciones IoT actuales: Se busca tener una panorámica de las herramientas y plataformas más utilizadas en el ámbito industrial, conociendo su naturaleza, ya sea open-source o bajo licencia, su nivel de seguridad, su interoperabilidad con otros sistemas y el soporte que brindan en términos de computación en la nube.

- Identificación de factores clave: Se extraerán los componentes esenciales de estas soluciones para poder entender qué hace a una plataforma o herramienta ser más adecuada para ciertos escenarios o necesidades.

- Creación de un catálogo de características: Se trabajará en consolidar un compendio de características, tanto básicas como avanzadas, que definan y diferencien a cada una de las soluciones estudiadas.

- Propuesta de criterios de selección: Se pretende proporcionar un método estructurado que permita a las empresas, instituciones u organizaciones decidir con precisión cuál es la solución IoT más adecuada para sus proyectos específicos.

A continuación, en el siguiente punto, se expondrá cómo este Trabajo Final de Máster busca alinearse y contribuir a los Objetivos de Desarrollo Sostenible, reflejando el impacto esperado de este estudio en un marco global.

1.3. Impacto esperado

El desarrollo y avance de la tecnología, específicamente en el ámbito de la Internet de las Cosas (IoT), tiene el potencial de contribuir significativamente a la consecución de los Objetivos de Desarrollo Sostenible (ODS) propuestos por la ONU. A continuación, se detallan los ODS más directamente relacionados con este Trabajo Final de Máster:

1. ODS 3 - Salud y bienestar: Al orientar una solución IoT para la informatización de viviendas con el objetivo específico de atender a personas con diabetes, el trabajo apunta a mejorar la calidad de vida y el bienestar de este grupo poblacional. Las herramientas tecnológicas pueden facilitar la monitorización y control de la enfermedad, y así contribuir a una vida más sana.

2. ODS 9 - Industria, innovación e infraestructura: El análisis de soluciones y plataformas IoT contribuye al desarrollo de infraestructuras resilientes y promueve la industrialización inclusiva y sostenible. Además, este tipo de trabajos impulsa la innovación al explorar las mejores prácticas y herramientas disponibles para la creación de soluciones IoT.

3. ODS 11 - Ciudades y comunidades sostenibles: Al promover la creación de 'Smart houses' orientadas a necesidades específicas, se está contribuyendo a la generación de espacios urbanos más inclusivos, seguros y sostenibles. La IoT tiene el potencial de mejorar la calidad de vida en las ciudades al optimizar el uso de recursos.

Tras entender el profundo impacto potencial de la IoT en diversos ámbitos, especialmente en la consecución de los Objetivos de Desarrollo Sostenible, es esencial abordar cómo se llevó a cabo este estudio. A continuación, se presenta la metodología adoptada para esta investigación, delineando cada fase y subrayando su relevancia en el contexto general del trabajo.

1.4. Metodología

Dada la naturaleza exploratoria de este trabajo, hemos optado por una metodología que, aunque no se ajusta estrictamente a los enfoques tradicionales de desarrollo, está diseñada para cumplir con los objetivos establecidos. El proceso se divide en varias etapas:

1. Selección y Análisis de Plataformas: Esta fase inicial consistió en la identificación y selección de un conjunto representativo de plataformas IoT, basándose tanto en su popularidad como en su aplicabilidad en distintos contextos. Cada plataforma fue analizada detalladamente, estudiando documentación oficial, artículos técnicos, y análisis previamente realizados por otros investigadores.

2. Investigación Documental: Se realizó una exhaustiva revisión de la literatura y documentos pertinentes, como se menciona en el estado del arte. Esta fase tuvo como objetivo profundizar en la comprensión de las capacidades y limitaciones de las plataformas seleccionadas, así como identificar características comunes y diferenciadoras entre ellas.

3. Identificación de Características Clave: A partir de la extensa revisión documental y el análisis de diferentes plataformas, se identificaron una serie de características esenciales y deseables que deberían poseer las soluciones basadas en IoT. Estas características fueron recopiladas no solo a partir de los análisis individuales de las plataformas, sino también de documentos técnicos y literatura especializada en el campo.

4. Tabulación de Características por Plataforma: Con el objetivo de ofrecer una comparación clara y directa entre las distintas soluciones, se elaboraron tablas que reflejaban las características identificadas para cada plataforma. Esta representación gráfica facilita el proceso de selección y decisión al poder visualizar de manera comparativa las fortalezas y debilidades de cada plataforma.

5. Elaboración de Escenarios Ilustrativos: Para proporcionar un contexto práctico y facilitar la comprensión de la utilidad real de las distintas características y soluciones, se diseñaron varios escenarios representativos de situaciones o contextos reales. A través de estos escenarios, y basándonos en la información recopilada previamente, se argumentó qué plataformas IoT son las más adecuadas para cada caso específico, teniendo en cuenta los requerimientos y necesidades propios de cada situación.

6. Selección y Justificación de la Plataforma Preferida: Finalmente, con toda la información y análisis en mano, se procedió a seleccionar la plataforma que mejor se adaptaba a las necesidades de un proyecto concreto: convertir una vivienda en una 'Smart house' diseñada específicamente para las necesidades de personas con diabetes.

Para visualizar mejor el proceso seguido, se adjunta una figura (Ilustración 1) que representa la secuencia de pasos:

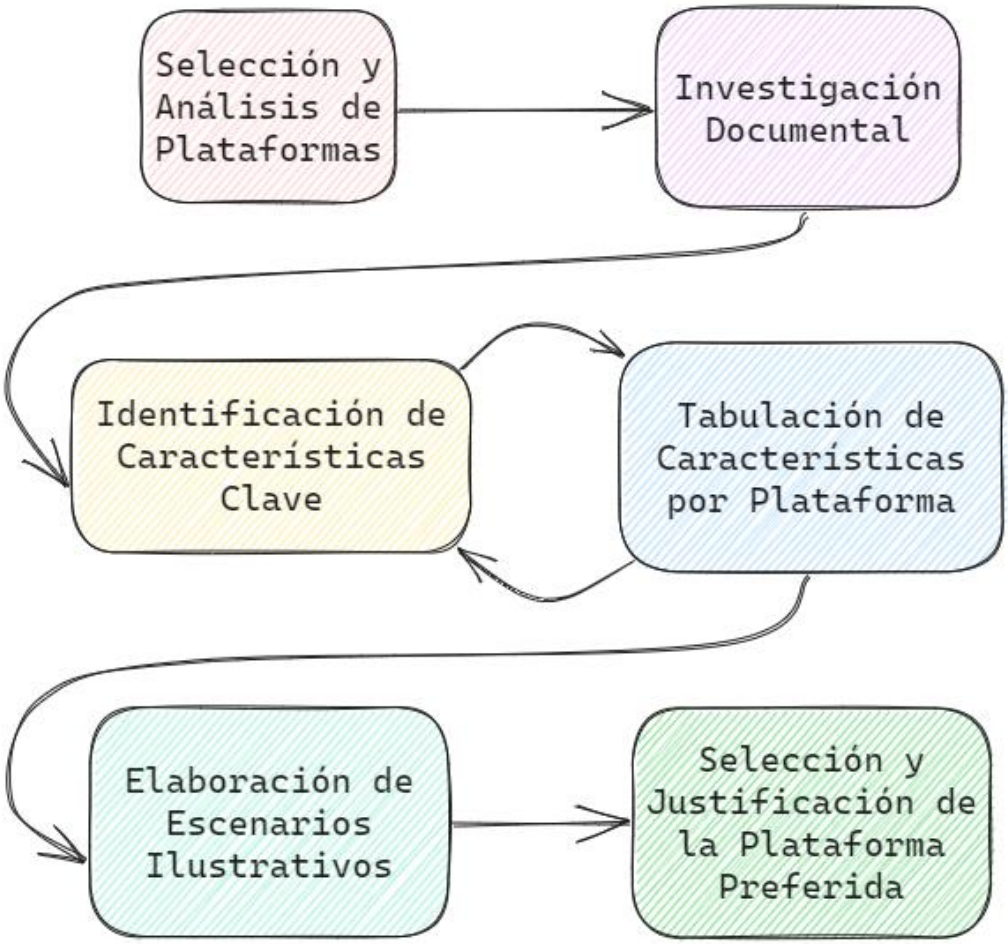

Ilustración 1. Metodología: Secuencia de pasos

Es preciso mencionar que, aunque la metodología aquí descrita pueda parecer lineal, el proceso de investigación y análisis fue dinámico. La recopilación y estructuración de información, así como la redacción, se adaptaron según las ideas y hallazgos que emergían durante el desarrollo del trabajo. Las notas y apuntes tomados en el transcurso del proceso se integraron cuidadosamente en el documento final, asegurando una presentación coherente y comprensiva del estudio realizado.

2. Estado del arte

2.1. Análisis de trabajos existentes

En el contexto actual de la Ingeniería Informática y en particular del Internet de las Cosas (IoT), es notable la presencia de distintas soluciones y plataformas. Diversos trabajos han abordado el análisis y comparativas de estas soluciones, aunque no de una manera tan exhaustiva como se propone en este Trabajo Final de Máster (TFM). A continuación, procedo a enumerarlos y a exponer de que tratan comparándolos con este proyecto, para así remarcar en la medida de lo posible, como se diferencian de este y como este consigue sobrepasarlos.

Uno de los documentos encontrados fue publicado en IEEE [1]. En él se proporciona una lista de las funcionalidades, requisitos y mejores prácticas de una plataforma IoT en la nube, para posteriormente realizar una comparativa general a través de tablas de diferentes plataformas y frameworks de IoT, sin profundizar en los detalles.

Otro trabajo, denominado "*A survey of IoT cloud platforms*" [2], realiza un análisis de las plataformas en la nube IoT más populares y proporciona una comparativa de los pros y los contras de manera bastante escueta. Aunque ofrece una visión general útil, no proporciona un análisis profundo de las plataformas y su aplicabilidad.

El tercer estudio [3] se enfoca en el *benchmarking* de dos de los principales servicios de IoT en la nube: Microsoft Azure IoT Hub y Amazon Web Services IoT. Mide su rendimiento en relación con el RTT (tiempo de ida y vuelta), mostrando resultados significativos en función de ciertos factores y situaciones de prueba. A pesar de ser un análisis útil, está limitado a solo dos servicios de la nube.

En el caso del "*Study of Various Internet of Things Platforms*" [4], se introducen y discuten varias plataformas IoT y middleware, señalando sus puntos clave y comparándolas en función de algunos parámetros generales. Aunque es el trabajo que mejor describe cada plataforma, su enfoque es general y no tan específico como se busca en este TFM.

El proyecto dirigido de Pedro Nel Martinez [5] ofrece un estudio comparativo aplicado a plataformas en la nube que ofrecen servicios IoT, con el propósito de identificar los criterios que podrían determinar la idoneidad de una plataforma en particular para un escenario IoT determinado. Este trabajo sirvió como una base para organizar el planteamiento de la solución propuesta en este TFM, y proporcionó un abundante número de documentos para respaldar nuestra investigación, tales como el estudio comparativo de plataformas middleware IoT [5], estudio de modelos de seguridad para comunicación end-to-end en arquitecturas para IoT en cloud [6] y aquellos que considero que han sido de gran ayuda, los artículos de investigación mencionados al inicio de este punto, [1] y [2]. Sin embargo, presenta limitaciones en su enfoque.

A pesar de que Martinez se centra en identificar diferencias entre las plataformas IoT contemporáneas basándose en una serie de características, la forma en la que presenta la información es, me atrevería a decir, burda y bruta. La información se expone de manera que no parece haber conexiones claras entre los distintos fragmentos de información, mezclándola de tal manera que provoca que el lector no pueda seguir un hilo claro de lectura y razonamiento. De hecho, parece que el trabajo se encuentra inacabado ya que las tablas en las que debería haberse realizado la comparativa están sin completar.

Finalmente, el TFG de Rodrigo Martinez Jacobson [8] lleva a cabo un análisis de plataformas seleccionadas por el autor, consideradas como las más populares del mercado. Realiza una comparación superficial basándose en las ventajas y desventajas que presentan. Aunque el autor realiza un experimento práctico para profundizar en el funcionamiento de las plataformas más destacadas, el análisis no es tan detallado como se pretende en este TFM.

2.2. Conclusiones

A pesar de la valiosa información encontrada en estos trabajos, queda claro que ninguno de ellos aborda el tema con la profundidad y exhaustividad que se busca en este TFM. La mayoría de ellos se limitan a ciertos aspectos o ciertas soluciones, sin proporcionar un análisis completo y conectado de las plataformas y sus características.

Para concluir, me gustaría remarcar que no se ha encontrado ningún trabajo que trate de abarcar, analizar y comparar de una manera tan exhaustiva el mercado de plataformas con la profundidad que pretende alcanzar este proyecto. Esto puede deberse a la densidad de la información que es posible obtener hoy en día, lo que dificulta realizar un estudio profundo de las diferentes plataformas y sus ventajas cuando se desea desarrollar una aplicación.

3. Caracterización de soluciones IoT

3.1. Componentes esenciales de la pila de soluciones IoT

La Internet de las Cosas constituye un paradigma tecnológico fascinante y en rápida evolución que facilita la interacción entre dispositivos, datos y decisiones. Aprovecha los avances en informática, redes y procesamiento de datos para transformar nuestra interacción con el mundo físico. En este punto, nos adentramos en los entresijos de los componentes esenciales de la pila de soluciones IoT, que consisten en dispositivos, conectividad, plataformas, análisis y aplicaciones.

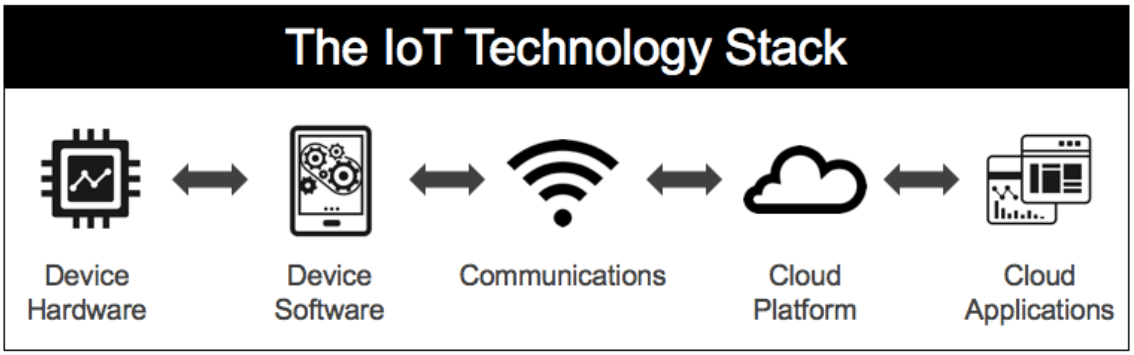

Ilustración 2 Pila de soluciones IoT

La base de cualquier sistema IoT son sus dispositivos, que forman la primera capa de la pila de soluciones IoT. Estos dispositivos, a menudo denominados "dispositivos inteligentes" o "dispositivos conectados", abarcan desde sensores que miden factores ambientales hasta actuadores que realizan acciones físicas, pasando por *gateways* (pasarelas) que proporcionan funciones de comunicación. Los sensores pueden ser de cualquier tipo: de temperatura en un sistema de calefacción doméstico inteligente, de frecuencia cardíaca en un monitor de fitness o de humedad en un sistema de riego agrícola. Los actuadores, por su parte, ejecutan acciones como ajustar la temperatura ambiente o activar los aspersores a partir de los datos aportados por los sensores. Las pasarelas, especialmente cruciales en un entorno empresarial, sirven de intermediarias para los dispositivos que no pueden comunicarse directamente con la plataforma IoT, recopilando y preprocesando los datos antes de enviarlos a la red.

La conectividad, la segunda capa de la pila de soluciones IoT, permite que los datos generados por los dispositivos lleguen a las unidades de procesamiento, donde pueden analizarse y utilizarse. Existen multitud de protocolos de comunicación que se adaptan a las distintas especificaciones de los dispositivos y a los requisitos de los casos de uso: Wi-Fi, Bluetooth, redes celulares, comunicación por satélite y conexiones por cable. La elección del protocolo depende de

numerosos factores, como el alcance requerido, el consumo de energía, el caudal de datos o el entorno operativo.

A continuación, viene la plataforma IoT, una capa de middleware crucial que sirve de puente entre los componentes de hardware y las herramientas de procesamiento de datos. Esta plataforma de software basada en la nube o local es responsable de varias tareas, como pueden ser: gestión y registro de dispositivos, almacenamiento y procesamiento seguro de datos y provisión de herramientas analíticas.

La cuarta capa, la analítica, transforma los datos brutos en información comprensible y práctica. Con la continua afluencia de datos procedentes de una gran cantidad de dispositivos, es indispensable subrayar la importancia de un análisis de datos eficaz y competente a la hora de cribar este torrente de datos para extraer información valiosa. Existen numerosos tipos de análisis, entre los que destacamos: el descriptivo, de diagnóstico, predictivos y prescriptivos. Los análisis descriptivos proporcionan una instantánea en tiempo real del estado actual del sistema, mientras que los análisis de diagnóstico profundizan en las causas de los sucesos observados. Los análisis predictivos utilizan modelos estadísticos y técnicas de previsión para predecir resultados futuros, y los análisis prescriptivos sugieren el mejor curso de acción basado en estas predicciones.

La última capa abarca las aplicaciones, esencialmente el aspecto del sistema IoT orientado al usuario. Estas aplicaciones de software convierten los conocimientos derivados del análisis de datos en servicios tangibles. La capa de aplicación puede incluir sistemas de supervisión para el seguimiento en tiempo real, sistemas de control para la gestión de dispositivos, sistemas de automatización que aprovechan el aprendizaje automático para el funcionamiento inteligente o sistemas de apoyo a la toma de decisiones que ayudan en la planificación estratégica.

En conclusión, la eficacia y eficiencia de un sistema IoT dependen de la perfecta interacción entre los cinco componentes esenciales mencionados con anterioridad. A medida que el ecosistema IoT siga expandiéndose, también lo harán la complejidad y las capacidades de estos componentes.

3.2. Comprendiendo las plataformas IoT y sus principales características

A medida que nos adentramos en un mundo cada vez más conectado, el Internet de las Cosas (IoT) se ha convertido rápidamente en una parte integral de nuestra vida cotidiana. Desde los hogares inteligentes y la asistencia sanitaria hasta la agricultura y el transporte, el IoT está cambiando fundamentalmente la forma en que interactuamos con el mundo que nos rodea. Para el funcionamiento de este vasto sistema interconectado son fundamentales las plataformas IoT, el corazón y el cerebro de los dispositivos inteligentes.

Una plataforma IoT, en esencia, es el centro neurálgico de un sistema IoT, alojado normalmente en una infraestructura en la nube, encargado de facilitar el procesamiento de datos, la obtención de información significativa y la conectividad de los dispositivos conectados dentro del ecosistema IoT. Constituye el puente fundamental entre los datos recopilados a nivel de dispositivo (cosas) y la capa de aplicación en la que se utilizan los datos para proporcionar perspectivas significativas. Destaca por su capacidad para gestionar grandes volúmenes de dispositivos, a menudo del orden de miles o incluso millones. Estas características, junto con su modelo de precios de pago por uso, proporciona flexibilidad, escalabilidad y rentabilidad, lo que la hace muy atractiva para las empresas.

Por lo que, existen dos aspectos que consideramos relevantes en el dominio de las plataformas IoT:

1. Las funcionalidades que nos ofrecen para gestionar y configurar una solución
2. Los modelos de comunicación que nos proporcionan, que nos caracterizan el tipo de soluciones que podemos aplicar

La siguiente sección proporciona información sobre estos aspectos.

3.3. Funcionalidades básicas de las plataformas IoT

Ahora nos centraremos en las funcionalidades básicas que ofrecen las plataformas IoT, las cuales serán abordadas detalladamente tras su mención:

- Gestión de Dispositivos
- Recopilación y procesamiento de Datos
- Análisis de Datos y Perspectivas
- Integración e Interoperabilidad
- Seguridad
- Interfaz de Usuario
- Modelos de servicios de computación en la nube
    - Infraestructura como Servicio (IaaS)
    - Plataforma como Servicio (PaaS)
    - Software como Servicio (SaaS)

3.3.1. Gestión de dispositivos

Una plataforma IoT es responsable de gestionar y mantener los dispositivos IoT conectados a ella. Esto incluye funciones como el aprovisionamiento de dispositivos (añadir y autorizar nuevos dispositivos), las actualizaciones de firmware y software, la resolución de problemas, la gestión de la seguridad y el diagnóstico del rendimiento de los dispositivos. La plataforma garantiza que los dispositivos no sólo funcionen según lo esperado, sino que también permanezcan seguros frente a posibles amenazas.

3.3.2. Recopilación y procesamiento de datos

Los dispositivos IoT generan continuamente una enorme cantidad de datos que deben recopilarse y procesarse. Las plataformas IoT tienen la funcionalidad esencial de recopilar estos datos, a menudo en tiempo real, y procesarlos para su posterior análisis. Esto puede incluir datos brutos de sensores de dispositivos, archivos de registro o mensajes de error.

3.3.3. Análisis de datos y perspectivas

Una vez recopilados y procesados los datos, hay que transformarlos en información práctica. Aquí es donde realmente brilla el poder de IoT. Las plataformas de IoT utilizan herramientas como la supervisión de los datos en tiempo real, algoritmos de aprendizaje automático para predecir patrones futuros y la IA para analizar los datos, obtener información y facilitar la toma de decisiones. Esta funcionalidad permite a las empresas tomar decisiones informadas basadas en datos y perspectivas en tiempo real, mejorando así la eficiencia y la productividad.

3.3.4. Integración e interoperabilidad

El ecosistema IoT es una confluencia de múltiples tecnologías, protocolos, dispositivos y aplicaciones. Por lo tanto, las plataformas IoT deben garantizar una integración e interoperabilidad perfectas entre estos componentes. Utilizan protocolos de comunicación estándar como MQTT, HTTP, CoAP, etc., y ofrecen API (interfaces de programación de aplicaciones) para integrarse con otros sistemas de software.

3.3.5. Seguridad

En el contexto del IoT, la seguridad es un aspecto crítico y de vital importancia. Las plataformas IoT deben garantizar el intercambio seguro de datos entre los dispositivos y la plataforma. Esto incluye cifrar los datos en reposo y en tránsito, garantizar el control de acceso y proporcionar mecanismos para la detección y mitigación de amenazas.

3.3.6. Interfaz de usuario

Las plataformas IoT también proporcionan una interfaz de usuario que permite a los usuarios gestionar fácilmente sus dispositivos IoT, ver datos y extraer perspectivas. Puede ser en forma de cuadros de mando, herramientas de generación de informes y otras ayudas a la visualización.

3.3.7. Modelos de servicios de computación en la nube

La funcionalidad de una plataforma IoT se realiza a menudo a través de servicios de computación en la nube. Estos modelos de servicio proporcionan la columna vertebral que sustenta el sólido funcionamiento de una plataforma IoT. Para entender este funcionamiento, profundizaremos en los tres principales modelos de servicios de computación en nube: Infraestructura como Servicio (*IaaS*), Plataforma como Servicio (*PaaS*) y Software como Servicio (*SaaS*).

3.3.7.1. Infraestructura como servicio (*IaaS*)

IaaS es un modelo de computación en nube que proporciona tanto elementos fundacionales como infraestructura virtualizada. Gestionada por proveedores externos, es esencial para las operaciones de las empresas y constituye una parte fundamental de una plataforma de IoT en la nube. Proporciona la infraestructura esencial para desplegar componentes *PaaS* y *SaaS*. Al ofrecer recursos informáticos virtualizados a través de Internet permite a los usuarios ejecutar máquinas virtuales, almacenar datos, escalar recursos bajo demanda y acceder a servidores, operando esencialmente un centro de datos virtual en la nube.

Aunque la infraestructura la gestionan los proveedores de servicios, los usuarios mantienen el control sobre ciertos aspectos como los sistemas operativos y las aplicaciones desplegadas. El servicio permite a las empresas ajustar los recursos para satisfacer demandas fluctuantes, lo que lo convierte en una solución flexible para empresas con necesidades de infraestructura cambiantes o impredecibles, incluidas las que trabajan con IoT.

IaaS proporciona los elementos fundamentales que necesita una plataforma de IoT en la nube para funcionar, como potencia de procesamiento, almacenamiento y capacidades de red. Entre los principales proveedores de *IaaS* se encuentran Amazon Web Services (AWS), Microsoft Azure, Google Compute Engine, Rackspace Open Cloud e IBM SmartCloud Enterprise.

3.3.7.2. Plataforma como servicio (*PaaS*)

PaaS es una capa de servicios de computación en nube construida sobre *IaaS*, diseñada para proporciona a los desarrolladores de software una plataforma para crear, ejecutar y probar aplicaciones sin necesidad de gestionar la infraestructura subyacente, como servidores, redes y almacenamiento.

PaaS ofrece un amplio conjunto de herramientas, bibliotecas, servicios y marcos que los desarrolladores pueden utilizar para crear y desplegar aplicaciones de forma más eficiente. Aunque los ingenieros no tienen control sobre la gestión y el control de la infraestructura en sí (está en manos del proveedor de servicios), los desarrolladores pueden ejercer control sobre las aplicaciones y sus configuraciones, e incluso configurar el entorno de alojamiento de aplicaciones para adaptarlo a sus necesidades específicas.

Aunque *PaaS* se puede utilizar en una amplia gama de escenarios, es particularmente beneficioso para aplicaciones en dispositivos IoT a través de la nube, ya que permite un proceso más ágil y *eficiente* en el desarrollo, prueba y despliegue de estas aplicaciones. El enfoque de PaaS se centra principalmente en el aspecto de desarrollo y despliegue de aplicaciones de software, lo que permite a los desarrolladores concentrarse más en sus labores y despreocuparse de la gestión de la infraestructura. Algunas opciones de *PaaS* industrial son Predix de GE, Sentience de Honeywell, MindSphere de Siemens, Cumulocity, Bosch IoT y Carriots.

3.3.7.3. Software como servicio (*SaaS*)

SaaS proporciona a los usuarios aplicaciones de software listas para usar a través de Internet, generalmente accesibles a través de un navegador web. Como servicio construido sobre la infraestructura *IaaS*, *SaaS* utiliza su potencia de procesamiento subyacente y sus capacidades de almacenamiento para ofrecer aplicaciones, que en el ámbito IoT, permiten controlar los sistemas IoT e interactuar con ellos.

Con *SaaS*, los usuarios pueden acceder cómodamente a las aplicaciones y utilizarlas sin necesidad de gestionar la infraestructura o instalar software en sus dispositivos, ya que son los proveedores los que encargan de estas tareas, esto no sólo simplifica el proceso, sino que también supone un importante ahorro de tiempo y costes, al ofrecer facilidad de mantenimiento y asistencia.

SaaS se centra fundamentalmente en la entrega de aplicaciones de software a los usuarios finales, lo que lo convierte en un elemento integral en el mundo de IoT. Algunos ejemplos habituales de *SaaS* son Google Apps y Cisco WebEx, mientras que Industrial Machinery Catalyst on the Cloud de Siemens es un ejemplo de SaaS industrial que utiliza la infraestructura de AWS.

En esencia, una plataforma de IoT en la nube reúne componentes *IaaS*, *PaaS* y *SaaS* para proporcionar una solución integrada y sin fisuras para gestionar, desarrollar y operar sistemas de IoT.

3.4. Escenarios principales en los que las empresas implementan soluciones IoT

En esta sección vincularemos el concepto de plataforma IoT a su aplicación práctica en diversos escenarios empresariales de IoT siendo estos: Comunicación de dispositivo a dispositivo, Mando y control centralizados, Monitorización remota e Inteligencia empresarial.

3.4.1. Comunicación de dispositivo a dispositivo

A menudo denominada comunicación *Machine-to-Machine* (M2M), consiste en conectar dos dispositivos locales o remotos. Con las plataformas IoT que orquestan estas comunicaciones basándose en reglas predefinidas y en la lógica empresarial, aumenta la eficiencia y se reducen las interrupciones de la producción. Por ejemplo, un termostato puede comunicarse con un sistema HVAC para controlar la temperatura, o los equipos de dos unidades de producción separadas pueden conectarse para equilibrar los niveles de producción en caso de interrupción.

3.4.2. Mando y control centralizados

Esta función permite controlar los dispositivos de forma remota a través de software, con aplicaciones de escritorio, web y móviles que actúan como controles remotos. Esta funcionalidad reduce los costes de asistencia mediante el acceso remoto y permite a los técnicos e ingenieros controlar los equipos desde lejos.

3.4.3. Monitorización remota

Los dispositivos IoT pueden transmitir telemetría e información de estado a la nube, lo que permite la supervisión continua de los estados de los dispositivos. Se pueden configurar alertas basadas en reglas predefinidas, lo que permite una respuesta rápida a cualquier mal funcionamiento del dispositivo. Además, los algoritmos de aprendizaje automático pueden utilizar los datos históricos acumulados para predecir y realizar el mantenimiento de los dispositivos.

3.4.4. Inteligencia empresarial

Los dispositivos IoT generan una gran cantidad de datos, proporcionando información para mejorar la eficiencia operativa, la eficiencia de la producción y la optimización de los recursos. Los datos de telemetría pueden agregarse a lo largo del tiempo para descubrir información procesable, como patrones de conducción, eficiencia del combustible, optimización de rutas y gestión de flotas en el caso de los vehículos conectados. Estos datos también pueden integrarse en los almacenes de datos de la empresa para su posterior análisis.

3.5. El futuro de la IoT

Como se ha comentado al inicio de este punto, el mundo se encuentra cada vez más conectado y la IoT se ha convertido rápidamente en una parte integral de nuestra vida cotidiana, y basándonos en la predicción realizada por Gartner [1] donde se afirmaba que en 2021 habrían 25.000 millones de dispositivos conectados [2] , se puede corroborar que este es un negocio muy lucrativo, y que según un informe del *McKinsey Global Institute* titulado "*Disruptive Technologies: Advances that will transform life, business, and the global economy*" [3] (Tecnologías disruptivas: Avances que transformarán la vida, los negocios y la economía mundial), IoT tiene potencial para crear efectos económicos por valor de entre 2,7 y 6,2 billones de dólares de aquí a 2025. Esto implica que muchas empresas hayan comenzado a implementar sus propias soluciones IoT lo que ha derivado en una competencia cada vez es mayor por ser los líderes en el mercado de esta tecnología.

Los especialistas del sector tienen que empezar a buscar ya una plataforma de IoT en la nube o arriesgarse a quedarse atrás lo que nos lleva a otro problema. Con este auge del desarrollo de IoT, Gartner también predice un auge de la Plataforma como Servicio (PaaS) de IoT, afirmando que para 2020, más de la mitad de las aplicaciones desarrolladas en una PaaS estarán centradas en la IoT [4].

[1] https://www.forbes.com/companies/gartner/
[2] https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends#:~:text=Intelligence%20(AI)-,Gartner,-forecasts%20that%2014.2
[3] https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/disruptive-technologies
[4] http://www.gartner.com/newsroom/id/3241817

4. Análisis de plataformas IoT

4.1. Criterio de elección de las plataformas y frameworks IoT

Al comienzo de este proyecto no se conocían apenas plataformas en la nube, y concretamente ninguna enfocada al ámbito del IoT, más allá de aquellas con las que trabajamos durante el curso académico (Amazon Web Services).

Debido a esto, para alcanzar a tener el repertorio de plataformas seleccionadas que poseemos a día de hoy, la selección se basó en 3 puntos:

1. Recomendaciones por parte del tutor: A partir de los conocimientos del tutor en el uso de plataformas IoT en los últimos 15 años, tanto a nivel de investigación como a nivel industrial, pude hacerme una idea del tipo de características, además de orientarme en cómo iniciar y dirigir el trabajo en la búsqueda de información y extracción de indicadores relevantes. Algunas de las plataformas que conocí gracias a él y que aparecen en este abajo son por ejemplo Sofia2, Sentilo o Carriots.

2. Conocimiento de la existencia de plataformas de computación en la nube ofertadas por grandes compañías: Pese a mi breve conocimiento acerca de este tema, era consciente de la existencia de gigantes tecnológicos que proporcionaban servicios de computación en la nube, por lo que decidí investigar si daban implementaciones que se adaptasen al ámbito del IoT. Estoy hablando de: Amazon Web Service (AWS), Google Cloud, IBM Watson y Microsoft Azure.

3. Investigación de las opciones existentes mediante la lectura de artículos académicos y contenido web [9]: Tras una investigación exhaustiva, se recopiló una gran cantidad de plataformas y frameworks de IoT. Posteriormente, se realizó un filtrado de estas, seleccionando aquellas que se mencionaban con mayor frecuencia en los diferentes medios escritos consultados. Este criterio se empleó para identificar las plataformas más populares y utilizadas de entre el conjunto de opciones disponibles.

Una vez explicado el método de selección, procederemos a analizar cada una de las plataformas seleccionadas de manera individual.

4.2. Plataformas y frameworks IoT escogidos

4.2.1. Altair SmartWorks IoT

4.2.1.1. ¿Qué es?

Ilustración 3. Logo de Altair SmartWorks

SmartWorks IoT es una solución basada en la nube, diseñada para ayudar a las empresas de todos los tamaños en los sectores de la robótica, la energía, la maquinaria industrial, la electrónica, la fabricación y otros diversos sectores a crear aplicaciones de IoT seguras para la web y los dispositivos móviles.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[5].

[5] https://help.altair.com/smartworks/index.htm
https://www.altair.com.es/smartworks-iot/#:~:text=Altair%20SmartWorks%20IoT%3A%20AnythingDB%20Overview&text=It%20enables%20a%20contextual%2C%20digital,interfaces%20to%20access%20those%20entities
https://forum.altairsmartcore.com/smartcore-overview
(Última fecha de consulta: 15/05/2022)

4.2.1.2. ¿Qué ofrece?

Este servicio se encuentra entre aquellos agrupados en Altair SmartCore (Ilustración 4).

SmartWorks IoT permite a los profesionales modelar, visualizar y manejar activos de IoT y desplegar y ejecutar aplicaciones en entornos híbridos, en la nube o en las instalaciones.

Ilustración 4. Arquitectura de Altair SmartWorks

La solución permite a los equipos de TI gracias al conjunto de herramientas que ofrece implementar clústeres de computación en el edge (en un entorno IoT, el término 'Edge' se refiere a la computación realizada en o cerca del lugar donde se generan los datos. Gracias a la proximidad con las fuentes de datos, los dispositivos y sensores de IoT pueden procesar, analizar y tomar decisiones basadas en los datos recopilados en tiempo real, sin necesidad de transmitir estos datos a un centro de datos centralizado para su procesamiento.), entrenar y ejecutar modelos de machine learning, implementar lógicas empresariales de aplicaciones complejas, realizar transformaciones de datos, almacenar conjuntos de datos de forma segura, visualizar datos en tiempo real mediante el uso de gráficos y dashboard que se actualizan automáticamente al segundo y mucho más las cuales se explicaran más detalladamente a continuación.

- *Functions*

Permite a los desarrolladores escribir código en el lenguaje que prefieran, desplegarlo en la infraestructura de forma automática y ejecutar el código en función de varios desencadenantes, desde solicitudes HTTP hasta eventos de dispositivos, bases de datos o plataformas. Puede

construir gemelos digitales o comparar resultados simulados con resultados reales. Estas se dividen en 2 partes:

- *Worker*: La función a ejecutar.
- *EventTriggers*: Componente que captura eventos y mensajes y los redirigen a una o más Funciones.
  Evento->Solicitud HTTP-> invocar Worker

Ejemplo: Actualizar el estado de la CPU en función de la carga de uso de la misma, desencadenante (*Trigger*), se recibe un mensaje indicando el nivel de carga, si supera un límite, se "avisa" a un Worker para que ejecute su función, por ejemplo, modificar el estado de la CPU a "*High usage*".

- AnythingDB

Base de datos de documentos para almacenar y acceder a los metadatos. Estos metadatos se separan en Colecciones que agrupan objetos similares. Esta separación permite controlar el acceso a colecciones enteras de objetos y hace que sus consultas en el front-end sean mucho más específicas. Las Colecciones se pueden crear siguiendo un esquema predefinido o con uno propio (básicamente una forma de definir la agrupación de objetos en JSON, por ejemplo, un "Ordenador", color, CPU, SO…)

- Visualización en tiempo real

Permite visualizar los datos almacenados y mostrarlos en: Treemaps, Heat Maps, Scatter Plots, Horizon Graphs y una amplia gama de otras excelentes visualizaciones. Dichas visualizaciones han sido diseñadas para una comprensión rápida y una interpretación fácil de estática, series temporales, transmisión en tiempo real y conjuntos de datos históricos. La visualización en tiempo real ofrece datos actualizados al segundo (en tiempo real), actualizando automáticamente los gráficos con funciones de detección de anomalías y comentarios de los usuarios.

- *Stream Processing*

El *Stream Processing* (Procesamiento de Flujos) es un método para rastrear y analizar flujos de información de un evento, y eventualmente obtener conclusiones estructuradas útiles a partir de esa información en bruto. El Procesamiento de Flujos consiste en una combinación de datos de múltiples fuentes para inferir eventos o patrones que puedan demostrar actividades inusuales o anomalías, requiriendo en consecuencia una acción inmediata. Este método se conoce comúnmente como Procesamiento de Eventos Complejos o CEP. El motor CEP de Altair se llama Panopticon Streams y está construido de manera que pueda cooperar con diferentes motores CEP.

- EdgeOps

SmartWorks IoT EdgeOps es un conjunto de herramientas para construir, mantener y mejorar continuamente el código en dispositivos con recursos limitados en el borde de los ecosistemas de productos inteligentes. Este conjunto de herramientas consta de dos componentes principales:

  - *Edge Compute Platform*: Plataforma de desarrollo de vanguardia que se ejecuta en dispositivos con recursos limitados, por ejemplo, Raspberry Pi o Intel NUC. Se ocupa de garantizar la veracidad de los datos, conectarse a varios protocolos de comunicación, monitorear el uso de recursos, desarrollo para arquitecturas de chips específicas y establecer acceso remoto con una terminal o registros. Aprovecha una versión reducida de Kubernetes para administrar gran parte de la gestión de implementación.

  - *Cloud Management Interface*: Proporciona herramientas útiles para realizar un seguimiento de todo su código implementado, monitorearlo y empaquetar el código para futuras implementaciones.

4.2.1.3. ¿Cómo funciona?

Conexión entre SmartWork IoT y el objeto (Objeto real que se representa en una Colección de AnythingDB como dispositivo). La conexión se puede crear a través de una API REST para acceder a todos los recursos, la cual está protegida mediante *Oauth protocol*, permitiendo al usuario definir su control de acceso personalizado.

Los datos deben enviarse a través de HTTP o MQTT (estándar para la mensajería IoT. Permite la mensajería entre el dispositivo y la nube y entre la nube y el dispositivo) a SmartWorks IoT

4.2.1.4. Precios

Plataforma por suscripción. El coste de la suscripción depende del número de dispositivos y datos que se utilicen. Está disponible en una variedad de planes de precios, incluyendo el nivel gratuito, el nivel estándar, el nivel de empresa, que comienza en $ 1,500 por año.

El soporte se extiende a través de la documentación y el portal de la comunidad. También ofrece la posibilidad de trabajar con la plataforma creando un espacio de trabajo el cual puede contener hasta 2 dispositivos de manera gratuita.

4.2.2. Amazon Web Service IoT Core

4.2.2.1. ¿Qué es?

Ilustración 5. Logo de AWS IoT Core

Amazon Web Services (AWS) es una plataforma de servicios de nube que proporciona un pool de servicios de infraestructura de red tales como bases de datos, almacenamiento de información, inteligencia artificial, o el que en este caso nos importa, internet de las cosas, entre otros. AWS IoT Core es una plataforma IoT centrada en facilitar la conexión entre los dispositivos IoT y la nube de AWS, permitiendo la integración con los diversos servicios que AWS ofrece.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[6].

[6] https://aws.amazon.com/es/iot/
https://aws.amazon.com/es/iot-core/
https://docs.aws.amazon.com/es_es/iot/latest/developerguide/what-is-aws-iot.html
(Última fecha de consulta: 22/06/2022)

4.2.2.2. ¿Qué ofrece?

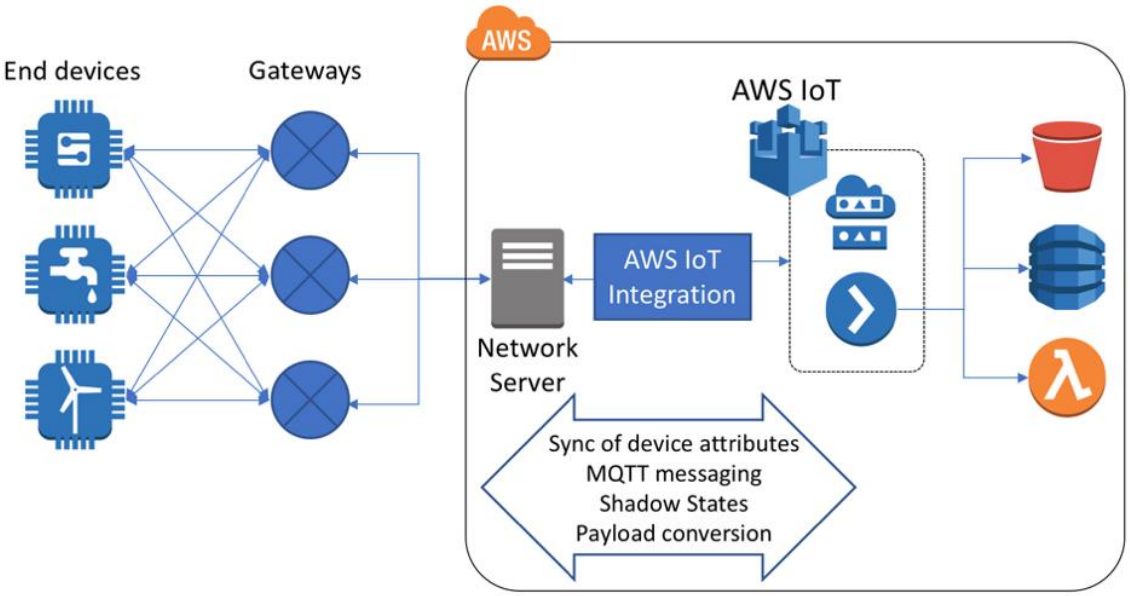

Ilustración 6. Integración de AWS IoT Core con otros servicios

AWS IoT ofrece un conjunto integral de servicios para la computación en la nube, facilitando la conexión e interacción de los dispositivos IoT con otros dispositivos y aplicaciones de la nube, proporcionando también almacenamiento y seguridad a través de la consola de AWS. La plataforma incluye un registro de dispositivos para su identificación, *Secure Device Gateway* para la comunicación segura, y un *Software Development Kit* (SDK) compatible con una amplia gama de dispositivos de fabricantes de hardware como Intel, Texas Instruments, Broadcom y Qualcomm.

Entre los diferentes servicios enfocados al ámbito IoT que ofrece AWS IoT destaca ***IoT Core***, que se encarga de administrar la autenticación, la conexión y la comunicación de los dispositivos con los servicios de AWS y entre sí.

Su punto de entrada (*Device Gateway*), admite los protocolos MQTT (*Message Queuing and Telemetry Transport*), MQTT over WSS(*WebSockets Secure*) HTTPS(*Hypertext Transfer Protocol - Secure*) y WebSocket. Amazon afirma que el módulo es lo suficientemente escalable para manejar desde unos pocos dispositivos hasta billones de ellos, y a cada uno se le asigna una identidad única. Debido a la autenticación y el cifrado proporcionados en todos los puntos de conexión, IoT Core y los dispositivos nunca intercambian datos no verificados.

Los mensajes examinados son procesados por el motor de reglas (*Rules Engine*) que los enruta a un dispositivo o a un servicio de AWS en la nube, como ***AWS Lambda*** (una plataforma informática sin servidor), ***Amazon Kinesis*** (una solución para procesar big data en tiempo real) o ***Amazon S3*** (*Simple Storage Service*) (un servicio de almacenamiento), por nombrar solo algunos.

Una característica esencial de IoT Core es *Device Shadow*, que mantiene una representación virtual del estado de cada dispositivo, permitiendo así que las aplicaciones de la nube puedan interactuar con un dispositivo incluso cuando este está offline. La plataforma ofrece SDKs para aplicaciones de Android e iOS, así como para C embebido, C++, JavaScript y Python. Amazon acelera el desarrollo al proporcionar una gran colección de plantillas junto con una herramienta visual de arrastrar y soltar llamada *IoT Things Graph* que simplifica la creación de flujos de trabajo en los componentes de IoT.

La funcionalidad ofrecida por IoT Core puede ser incrementada/completada haciendo uso de otros servicios del conjunto de servicios enfocados en el IoT:

- **AWS IoT Device Management**: Permite organizar, rastrear, controlar, actualizar y escalar flotas de dispositivos grandes y diversas de forma remota. Independientemente de un tipo de dispositivo, el servicio es compatible con cualquier cosa de IoT, desde microcontroladores hasta refrigeradores conectados.
- **AWS IoT Device Defender** verifica continuamente las configuraciones de IoT con respecto a los requisitos de seguridad y envía alertas cuando detecta brechas.
- **AWS IoT Events** está diseñado para identificar cambios complicados en el comportamiento de los equipos en miles de dispositivos y reaccionar ante ellos según reglas predefinidas.
- **AWS SiteWise** resulta útil cuando necesita recopilar y organizar datos a nivel industrial. El servicio se conecta al equipo de un fabricante a través de una puerta de enlace, recopila y preprocesa los datos y luego los envía a la nube de AWS.
- **AWS IoT 1-Click** se utiliza para hacer que un grupo de dispositivos realice acciones específicas (como enviar mensajes de alerta) con un clic de botón.

En adición a todo lo comentado, AWS ofrece una capa de seguridad robusta con mecanismos de autenticación y cifrado integral en todos los puntos de conexión, con el fin de asegurar la privacidad y la veracidad de los datos. Cada dispositivo o cliente conectado debe tener una credencial para interactuar con AWS IoT. Todo el tráfico hacia y desde AWS IoT se envía de forma segura a través de Transport Layer Security (TLS). Los mecanismos de seguridad de la nube de AWS protegen los datos mientras se mueven entre AWS IoT y otros servicios de AWS.

4.2.2.3. Precios[7]

AWS IoT Core permite conectar de manera confiable y segura miles de millones de dispositivos de IoT y dirigir billones de mensajes de IoT a los servicios de AWS y a otros dispositivos sin necesidad de administrar la infraestructura. Se paga solo por los componentes específicos que se utilicen. No hay ninguna tarifa de servicio por uso mínima ni obligatoria. Se factura de manera independiente el uso de conectividad, mensajería y sombras de dispositivos, registro y del motor de reglas. Este enfoque proporciona claridad y un precio bajo independientemente del tipo de carga de trabajo. El nivel gratuito de AWS se encuentra

[7] https://aws.amazon.com/es/iot-core/pricing/

disponible para los clientes de AWS IoT Core durante 12 meses a partir de la fecha de creación de la cuenta de AWS

Resumen de precios de AWS IoT Core extraído de la "Calculadora de precios de AWS"[8]

- **Conectividad**: 0,096 USD (por millón de minutos de conexión)
- **Mensajeria**:
  - Mensajería de MQTT y HTTP
    - Hasta mil millones de mensajes: N/D (por millón de mensajes)
    - Siguientes 4 mil millones de mensajes: N/D (por millón de mensajes)
    - Más de 5 mil millones de mensajes: 0,84 USD (por millón de mensajes)
  - Mensajería **LoRaWAN**
    - Hasta mil millones de mensajes: 2,30 USD (por millón de mensajes)
    - Siguientes 4 mil millones de mensajes: 1,50 USD (por millón de mensajes)
    - Más de 5 mil millones de mensajes: 1,20 USD (por millón de mensajes)
  - Mensajería de **LoRaWAN FUOTA**
    - Hasta 250 mil tareas FUOTA: 0,006 USD (por tarea)
    - Más de 250 mil tareas FUOTA: 0,003 USD (por tarea)
- **Servicios de registro y sombra del dispositivo**: 1,25 USD (por millón de operaciones)
- **Servicio de motor de reglas**
  - Reglas activadas: 0,15 USD (por millón de reglas activadas o de acciones ejecutadas)
  - Acciones ejecutadas: 0,15 USD (por millón de reglas activadas o de acciones ejecutadas)
- **Device Advisor**: Device Advisor se puede utilizar de forma gratuita.

[8] https://aws.amazon.com/es/iot-core/pricing/?nc=sn&loc=4
(Última fecha de consulta: 22/06/2023)

4.2.3. Carriots

4.2.3.1. ¿Qué es?

Ilustración 7. Logo de Carriots

Carriots es una plataforma de almacenamiento y desarrollo especializada en proyectos relacionados con Internet of Things (IoT) y *Machine to Machine* (M2M). Más concretamente, es una completa plataforma (AEP por sus siglas en inglés de *Application Enablement Platform*) para la creación rápida de aplicaciones IoT para conectar y administrar dispositivos, recopilar y analizar datos, e integrarse con otros sistemas empresariales.

Recientemente, en marzo de 2017, la compañía Carriots fue vendida al 100% a la compañía Altair, empresa dueña de la plataforma IoT comentada en el punto 3.2.1 Altair SmartWorks IoT.

James R. Scapa, Fundador, Presidente y CEO de Altair, señala que "Combinamos Altair con las tecnologías de nuestros socios y con la plataforma de Carriots para convertirnos en líderes del campo de IoT, incluyendo aplicaciones para las estrategias de nuestros clientes en el campo del Gemelo Digital (Digital Twin)".

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[9].

4.2.3.2. ¿Qué ofrece?

La plataforma Carriots permite construir potentes aplicaciones y servicios con una gran cantidad de funcionalidades con su "amigable" API y la SDK basada en Groovy). La principal característica que ofrece es la de integrar la aplicación del cliente de forma sencilla con otros sistemas de información externos usando sus potentes API's, *webservices* y el servicio de

[9] https://www.altairsmartworks.com/pdf/Carriots_Product_Sheet_Spanish.pdf https://www.nettrotter.io/en/carriots-platform#:~:text=Carriots%20is%20an%20application%20hosting,few%20lines%20of%20Groovy%20code.
(Última fecha de consulta: 18/05/2022)

alojamiento desatendido que ofrecen. Además de los ya mencionados servicios, esta plataforma ofrece otros servicios, tales como:

- Escalado automático para gestionar de 1 a millones de dispositivos (no hay limitación de cuánto pueden escalar tus proyectos tanto en volumen como en rapidez)
- Prototipos/plantillas de proyectos de los que partir para crear de una manera más sencilla y rápida tu propia aplicación
- Arquitectura de 7 niveles para la gestión de proyectos para varios clientes con tantas aplicaciones M2M como necesites.
- Alojamiento sin preocupaciones: Carriots está en la nube (no se aun como se almacena, mediante qué tipo de BD o como)
- Los módulos más comunes en los proyectos de M2M:
    - recolección de datos
    - almacenamiento de información
    - seguridad
    - gestión de dispositivos
    - …

4.2.3.3. Precios

En su día, previo a la adquisición de esta plataforma, ofrecía sus servicios para la creación de aplicaciones M2M de forma totalmente gratuita.

4.2.4. DeviceHive

4.2.4.1. ¿Qué es?

Ilustración 8 Logo de DeviceHive

DeviceHive (DH) es una plataforma de gestión de servicios en la nube de IoT de código abierto, con licencia de Apache versión 2.0, enfocada en el análisis de big data.

Técnicamente es plataforma basada en microservicios escalable, independiente del hardware y de la nube con API de administración de dispositivos en diferentes protocolos (REST, WebSockets y MQTT), que le permite configurar y monitorear la conectividad de los dispositivos, controlarlos y analizar el comportamiento.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[10].

[10] https://docs.devicehive.com/docs
https://docs.devicehive.com/docs/101-overview
https://devicehive.com/services/
https://docs.devicehive.com/docs/cassandra-plugin
(Última fecha de consulta: 19/05/2022)

4.2.4.2. ¿Qué ofrece?

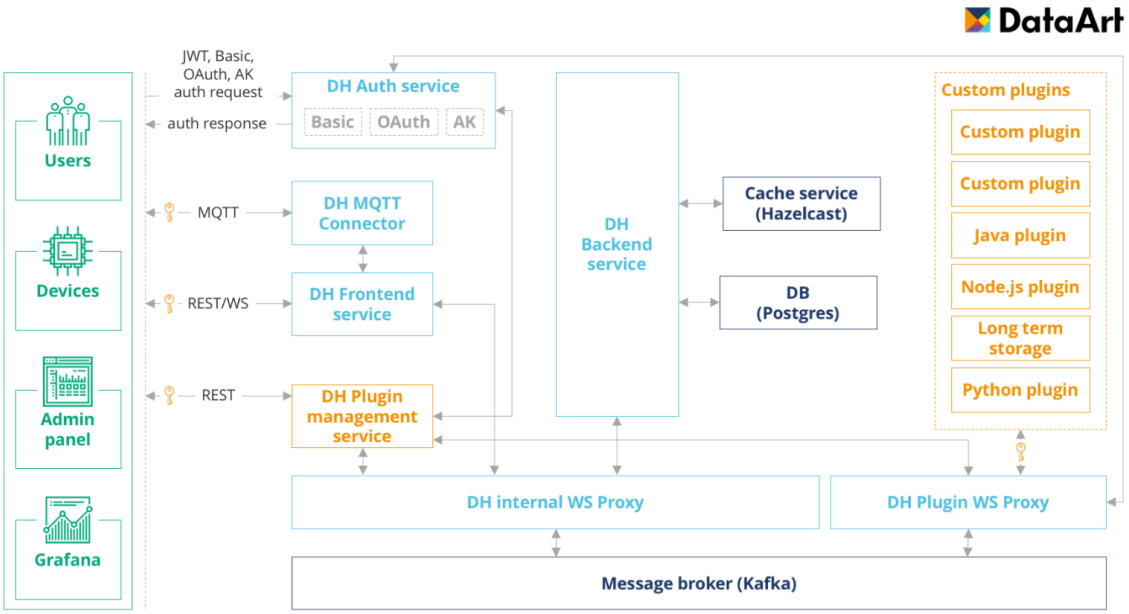

Ilustración 9. Plataforma DeviceHive

Como se indica en su documentación[11], esta plataforma cubre todo el flujo desde la Transición, Validación y Recopilación de Datos hasta trabajos de Machine Learning e Inteligencia Artificial. También proporciona herramientas de monitoreo para que pueda comenzar su descubrimiento sin estar obligado a conectar hardware real a la plataforma al inicio.

Ofrece la posibilidad de desplegar el servidor tanto en Kubernetes como en Docker (Instrucciones despliegue en Docker[12]) para trabajar así en tu "nube privada". A su vez brinda la opción de acceder a tu propio servidor DH alojado en su nube gracias a la herramienta *Playground*[13], la cual permite de una manera sencilla, crear proyectos IoT. Para facilitar la creación de una instancia, DeviceHive brinda un manual de instalación[14].

En cuanto a la conexión de dispositivos, aquellos que admitan el uso de protocolos API REST, WebSockets o API MQTT podrán conectarse a la plataforma, de igual manera, aquellos dispositivos que son compatibles con Python, Node.js o Java, como placas Linux, dispositivos Android Things, etc., se pueden conectar fácilmente instalando la biblioteca cliente de DeviceHive.

DeviceHive es un sistema basado en microservicios, construido con alta escalabilidad y disponibilidad. Una plataforma que no solo puede escuchar cientos de dispositivos simultáneamente, sino también escalar a la cantidad requerida de instancias para garantizar la

[11] https://docs.devicehive.com/docs/101-overview
[12] https://docs.devicehive.com/docs/deployment-with-docker
[13] https://playground.devicehive.com/
[14] https://github.com/devicehive/devicehive-java-server/wiki/Getting-started

seguridad y disponibilidad de los datos. Entre estos servicios (los cuales se muestran en la Ilustración 9) podemos encontrar:

- *PostgresSQL*: Almacenamiento de datos, este se realiza en la base de datos relacional *PostgresSQL*.
- Servicio de cache (*Cache service*) *Hazelcast IMDG*: Se encarga de almacenar los datos que son necesarios de manera temporal para su uso en trabajos analíticos. También facilita el rápido acceso inmediato a los datos cuando esto sea necesario. Almacena la información durante 2 minutos (limite que puede ser modificado), periodo tras el cual serán eliminadas.
- Bus de mensajería *Websocket Kafka Proxy*: Se encarga de la comunicación entre servicios y equilibra la carga entre ellos. Servicio de mensajería rápido, distribuido tolerante a fallos.
- *DH Backend Service*: Responsable de almacenar datos en Hazelcast, administrar suscripciones y recuperar datos por solicitud de otros servicios, ya sea de Hazelcast o de base de datos. No tiene una API de acceso público, toda la comunicación con él se realiza a través del bus de mensajería.
- DH Auth Service: Toda autenticación ejecutada en DH se realiza empleando JSON Web Tokens (JWT), cuyas ventajas son:
    - El token no tiene estado y es autónomo, no es necesario almacenarlo en el lado del servidor.
    - Portabilidad - un token se puede usar con múltiples backends.
    - Descentralizado, se puede generar en el servidor independiente.
    - Reduce el tiempo de cómputo y de red.

4.2.4.3. Precios

El conjunto de herramientas de DeviceHive es completamente gratuito. El gasto que conlleva esta solución depende del hosting escogido para el despliegue de la plataforma, ya sea en una nube privada o en otras tales como Amazon Web Services, Azure o Google Cloud (a excepción de la que ofrece DeviceHive mediante Playground).

4.2.5. Google Cloud IoT Core

4.2.5.1. ¿Qué es?

Ilustración 10. Logo de Google Cloud IoT Core

Google Cloud IoT Core es una plataforma de gestión de dispositivos IoT completamente gestionada, que forma parte integral del conjunto de productos y servicios de Google Cloud. Este servicio proporciona un conjunto de herramientas para administrar, analizar, almacenar y procesar datos procedentes de dispositivos IoT, desde unos pocos hasta millones, repartidos por todo el mundo de manera segura y eficiente. Ofrece la posibilidad de manejar los datos tanto localmente como en la nube, y es accesible para empresas de todos los tamaños. La plataforma aprovecha el poder de la analítica y el aprendizaje automático para extraer conclusiones valiosas o entender patrones a partir de los datos recolectados por los dispositivos IoT.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[15].

[15] https://cloud.google.com/solutions/iot?hl=es
https://cloud.google.com/iot/docs/concepts/overview?hl=es
https://cloud.google.com/iot/docs?hl=es#docs
https://cloud.google.com/iot/docs/concepts/device-security?hl=es
https://cloud.google.com/iot/quotas?hl=es
(Última fecha de consulta: 18/06/2022)

4.2.5.2. ¿Qué ofrece?

Los componentes principales de Cloud IoT Core son el administrador de dispositivos y los puentes de protocolo:

- Un administrador de dispositivos para registrar dispositivos en el servicio a fin de supervisarlos y configurarlos.
- Dos puentes de protocolo (MQTT y HTTP) que los dispositivos pueden usar para conectarse a Google Cloud Platform. Para la conexión y comunicación de los dispositivos.
    - MQTT es un protocolo estándar de publicación y suscripción que se usa y se admite con frecuencia en dispositivos incorporados, y que también es común en las interacciones entre máquinas.
    - HTTP es un protocolo "sin conexión": con el puente HTTP, los dispositivos no mantienen una conexión a Cloud IoT Core. En su lugar, envían solicitudes y reciben respuestas. Cloud IoT Core solo es compatible con HTTP 1.1 (no 2.0).

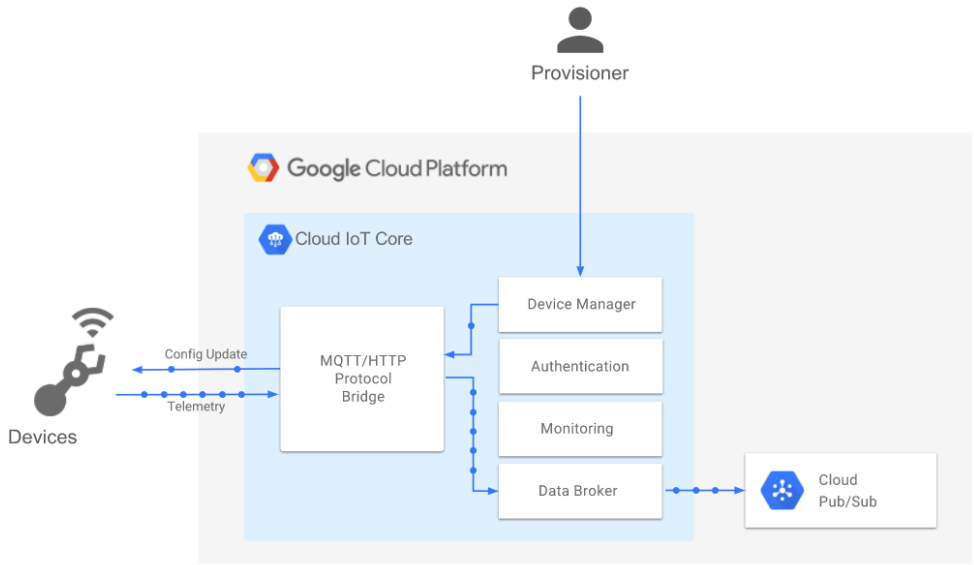

Ilustración 11. Componentes del servicio IoT Core y flujo de datos

Los datos de telemetría del dispositivo (datos de eventos (por ejemplo, medidas del entorno) que se envían desde los dispositivos a la nube) se reenvían a un topic de Cloud Pub/Sub, que se puede usar para activar Cloud Functions. También puedes realizar análisis de transmisiones con Cloud Dataflow o análisis personalizados con tus propios suscriptores.

El siguiente diagrama resume los componentes del servicio y el flujo de los datos:

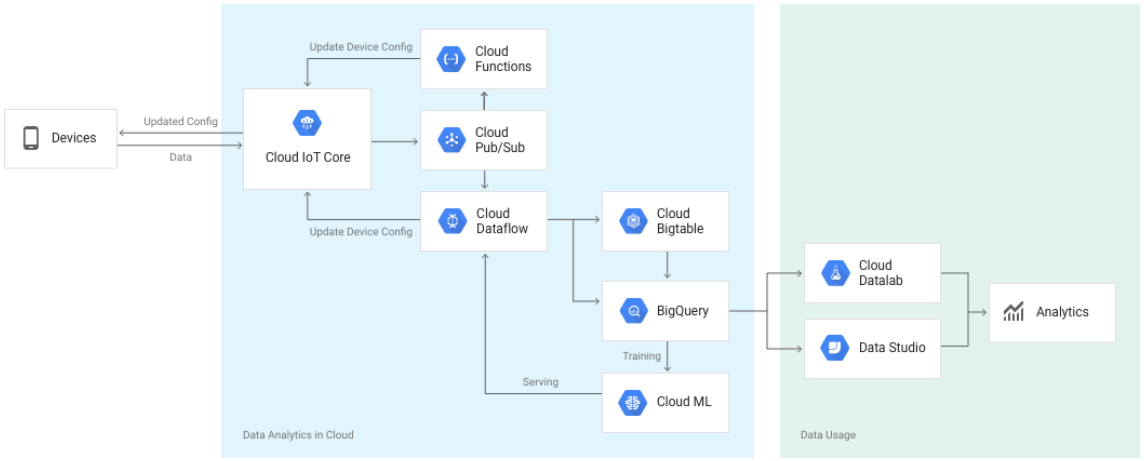

Ilustración 12. Integración de Cloud IoT Core con otros servicios

Seguridad

Ya que la seguridad es una preocupación fundamental cuando se implementan y administran dispositivos de IoT, Cloud IoT Core ofrece las siguientes funciones de seguridad:

- Autenticación de clave pública/privada por dispositivo con tokens web JSON (JWT, RFC 7519) (Los JWT son válidos durante un tiempo limitado, por lo que las claves vulneradas vencerán).
- Compatibilidad con algoritmos de RSA o de curva elíptica para verificar firmas con aplicación forzosa para tamaños de claves sólidos.
- Compatibilidad para rotar las claves por dispositivo permitiendo que se registren claves simultáneas, y compatibilidad con hora de vencimiento por credencial.
- Conexión TLS 1.2, con autoridades de certificación raíz (necesarias para MQTT).
- El acceso a la API de Cloud IoT Core se controla mediante funciones y permisos de la Administración de identidades y accesos (IAM).

Transferencia Datos

La transferencia es el proceso de importar información de los dispositivos a los servicios de Google Cloud. Google Cloud ofrece distintos servicios de transferencia, según si los datos son telemetría o información operativa sobre los dispositivos y la infraestructura de IoT.

- MQTT en IoT Core

IoT Core proporciona un agente seguro de MQTT (Message Queuing Telemetry Transport) para los dispositivos administrados por IoT Core. Este estándar permite que los dispositivos restringidos envíen telemetría en tiempo real y reciban de forma inmediata mensajes enviados desde la nube a un dispositivo con la función de administración de la configuración. El agente de MQTT de IoT Core se conecta de forma directa con Pub/Sub.

- Pub/Sub

Pub/Sub proporciona un servicio de transferencia de mensajes duradero en todo el mundo. Permite crear temas de transmisiones/canales, lo que habilita que los distintos componentes de la aplicación del cliente se subscriban a transmisiones de datos específicas. En adición, puede actuar como un amortiguador y nivelador de velocidad para las transmisiones de datos de entrada y los cambios en la arquitectura de la aplicación. Pub/Sub también se conecta de forma nativa a otros servicios de Google Cloud, lo que te ayuda a conectar la transferencia, las canalizaciones de datos y los sistemas de almacenamiento. Estos servicios son:

  - **Cloud Functions** para crear funciones independientes e instruir a los dispositivos sobre cómo reaccionar ante eventos específicos.
  - **Cloud Dataflow** para preprocesar datos en tiempo real.
  - **Cloud Bigtable** para ingerir y almacenar grandes volúmenes de datos.
  - **BigQuery** para analizar datos en tiempo real, crear y entrenar modelos de aprendizaje automático.
  - **Cloud Datalab** para desarrollar visualizaciones y prácticas analíticas personalizadas.
  - **Data Studio** para visualizar información extraída de BigQuery mediante plantillas prediseñadas.

Almacenamiento de datos

Los datos del mundo físico vienen en varias formas y tamaños. Google Cloud ofrece una gama de soluciones de almacenamiento, que incluyen desde BLOB de datos sin estructurar, como imágenes o transmisiones de video, hasta almacenamiento de entidades estructuradas de dispositivos o transacciones y bases de datos clave-valor de alto rendimiento para datos de eventos y telemetría.

- Almacena estados en IoT Core

Debido a que los dispositivos de IoT pueden pasar algo de tiempo en el modo de suspensión de baja energía y pueden existir en redes que tienden a ser poco confiables, a menudo es útil almacenar parte del estado de un dispositivo en la nube. De esa manera, los datos de estado pueden estar disponibles incluso cuando los dispositivos en sí están sin conexión por un tiempo.

El último estado conocido de un dispositivo puede informarse y almacenarse en IoT Core para que las aplicaciones lo recuperen. La información de estado que se envía por MQTT o HTTP se conserva en IoT Core y está disponible en la nube, incluso si el dispositivo se desconectó o no tiene conexión.

- Almacena datos de aplicación en Datastore y Firebase

Cuando se necesita que los datos de telemetría (Los datos recopilados por el dispositivo) o estado estén disponibles para aplicaciones web o para dispositivos móviles, estos pueden ser almacenados procesados o sin procesar en bases de datos estructuradas, pero sin esquemas, como Datastore y Firebase Realtime Database,

4.2.5.3. Precios

El precio de Cloud IoT Core se establece de acuerdo con el volumen de datos intercambiados con la plataforma mensualmente

Volumen de datos mensual	Precio por MB	Dispositivos registrados	Cargo mínimo*
Hasta 250 MB	0,00 $	Ilimitados, dentro de las cantidades máximas de consultas por segundo	1024 bytes
De 250 MB a 250 GB	0,0045 $	Ilimitados, dentro de las cantidades máximas de consultas por segundo	1024 bytes
De 250 GB a 5 TB	0,0020 $	Ilimitados, dentro de las cantidades máximas de consultas por segundo	1024 bytes
5 TB y más	0,00045 $	Ilimitados, dentro de las cantidades máximas de consultas por segundo	1024 bytes

Ilustración 13. Tabla de precios de Google Cloud IoT Core

Estos precios son exclusivamente referentes al servicio *Cloud IoT Core*, si en adición a este servicio, también se emplea *Cloud Pub/Sub* se debería de añadir a la factura total el precio de envió de mensajes.

- Mensajes inferiores a 1024 Bytes (1KB): 1024 * precio por MB del nivel de volumen de datos mensual
- Mensajes superiores a 1024 Bytes (1KB): Tamaño del mensaje * precio por MB del nivel de volumen de datos mensual

Mensajes inferiores a 1024 Bytes	Mensajes superiores a 1024 Bytes (1KB)
1024 * PRECIO_MB	Tamaño del mensaje (en Bytes) * PRECIO_MB

PRECIO_MB = Precio por MB del nivel de volumen de datos mensual

4.2.6. IBM Watson IoT

4.2.6.1. ¿Qué es?

Ilustración 14. Logo de IBM Watson IoT

IBM Watson™ IoT Platform es un servicio totalmente gestionado y alojado en la nube en IBM Cloud que facilita la derivación de valor de los dispositivos de Internet de las cosas (IoT). Está diseñado para analizar, gestionar, transformar, almacenar y exponer los datos de los dispositivos IoT.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[16].

[16] https://www.ibm.com/cloud/internet-of-things
https://cloud.ibm.com/docs/IoT?topic=IoT-about_iotplatform&locale=es
https://www.ibm.com/docs/en/mapms/1_cloud?topic=devices-about-platform-service
https://ibm-watson-iot.github.io/iot-python/application/api/dsc/
https://cloud.ibm.com/docs/IoT?topic=IoT-api_overview
https://cloud.google.com/iot/docs/concepts/device-security?hl=es
https://cloud.ibm.com/docs/IoT?topic=IoT-back_up
https://www.ibm.com/mysupport/s/question/0D50z00006PE7OfCAL/watson-iot-platform-scalability-performance-best-practice?language=es
https://cloud.ibm.com/catalog/services/internet-of-things-platform
https://internetofthings.ibmcloud.com/
(Última fecha de consulta: 10/07/2022)

4.2.6.2. ¿Qué ofrece?

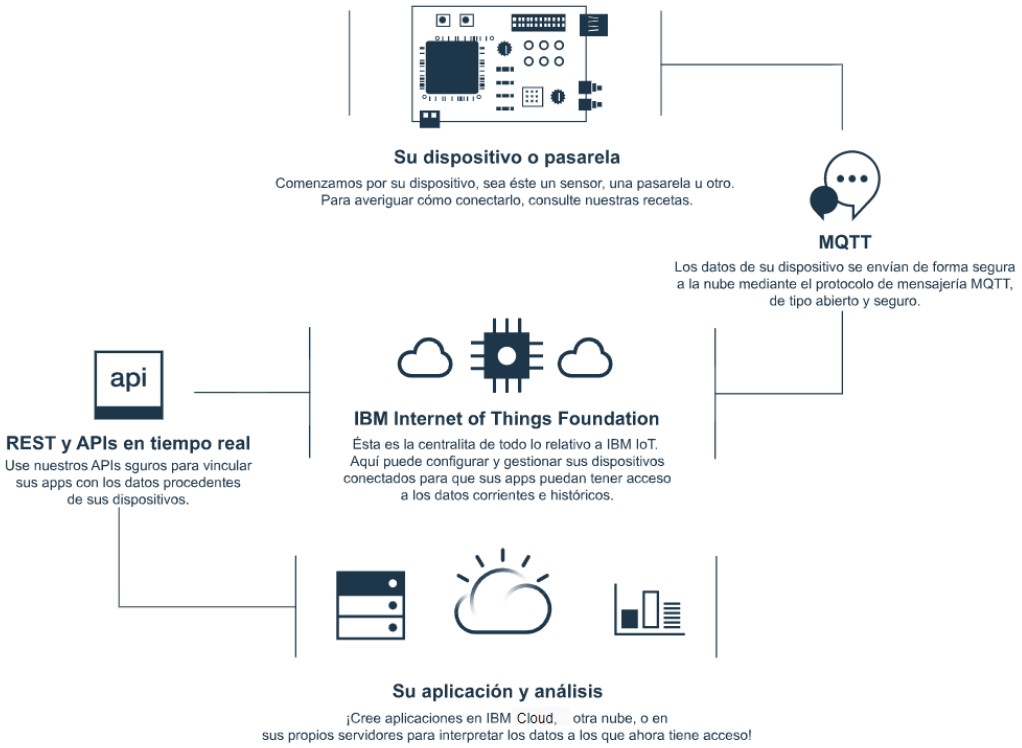

Ilustración 15. Esquema de funcionamiento de Watson IoT

Permite configurar y gestionar los dispositivos conectados al servicio y, de esta forma, permite que las aplicaciones creadas y enlazadas a los dispositivos puedan tener acceso a los datos en tiempo real y al histórico.

Por otro lado, ofrece APIs seguros que vinculan las aplicaciones con los datos procedentes de los dispositivos conectados a la plataforma. Además, permite el análisis de los datos sin salir de la plataforma mediante Bluemix.

IBM ha agrupado la plataforma en cuatro grandes bloques:

- IBM Watson IoT Platform Connect: Bloque que engloba la comunicación entre los dispositivos y la nube.
- IBM Watson IoT Platform Information Management: Permite la integración con datos de terceros y el almacenamiento de los datos recibidos por los dispositivos.
- IB Watson IoT Analytics: Provee a la plataforma la capacidad de realizar análisis predictivos, cognitivos, en tiempo real y contextuales.
- IBM Watson IoT Risk Management: Permite añadir a la plataforma una capa proactiva de seguridad y protección ante anomalías en el mundo del IoT

Protocolos de comunicación

Puede conectar aplicaciones, dispositivos y pasarelas a IBM Watson™ IoT Platform mediante el protocolo de MQTT. También puede utilizar la API REST HTTP para conectar dispositivos a Watson IoT Platform. Aquellos que se encuentren conectados a la plataforma enviarán los datos y estos podrán ser analizados y transformados, para luego quedar expuestos mediante las APIs REST y en tiempo real de la plataforma.

Almacenamiento de datos[17]

- PostgresSQL
- Se emplean otros servicios que ofrece IBM para ello:
  - IBM Cloud Object Storage data buckets
  - Db2 Warehouse on Cloud: Un almacén de datos de nube elástico y totalmente gestionado creado para el análisis de alto rendimiento y el aprendizaje automático.

Seguridad

Las API de IBM Watson™ IoT Platform soportan la autenticación y la autorización de los usuarios mediante Cloud Identity and Access Management (IAM).

Copia de seguridad de datos

Los datos se almacenan fuera del servicio principal de Watson IoT Platform, proporcionando redundancia geográfica y habilitando que los servicios se restauren en caso de una incidencia importante.

Escalabilidad

Watson IoT es un servicio en la nube y, como tal, a medida que aumente la demanda de cualquier servicio, se escalará la capacidad para satisfacer esa demanda. Los usuarios no tienen que preocuparse por eso, se hace automáticamente.

4.2.6.3. Precios

La plataforma ofrece un plan gratuito de hasta 500 dispositivos y 200 MB de datos al mes. Para consumos superiores, ofrece tres tipos de planes. Todos los planes son medidos, y están basados en el intercambio de datos

[17] https://dataplatform.cloud.ibm.com/docs/content/wsj/manage-data/cos_buckets.html

200MB of data metrics each month

The Standard plan includes the Lite plan and unlimited registered devices. Pay for what you use by tier.

Tier 1: 1 MB - 450GB	$0.001 USD per MB Exchanged
Tier 2: 450GB - 7TB	$0.0007 USD per MB Exchanged
Tier 3: 7TB and above	$0.00014 USD per MB Exchanged
Tier 1: 1 MB - 450GB	$0.003 USD per MB Analyzed
Tier 2: 450GB - 7TB	$0.0021 USD per MB Analyzed
Tier 3: 7TB and above	$0.00042 USD per MB Analyzed
Tier 1: 1 MB - 450GB	$0.0005 USD per MB Analyzed at Edge
Tier 2: 450GB - 7TB	$0.00035 USD per MB Analyzed at Edge
Tier 3: 7TB and above	$0.00007 USD per MB Analyzed at Edge

Ilustración 16. Tabla de precios de IBM Watson IoT

4.2.7. IoTsens

4.2.7.1. ¿Qué es?

Ilustración 17. Logo de IoTsens

IoTSens es una empresa proveedora de soluciones verticales de IoT para recolectar, integrar, almacenar y analizar la información de verticales como Smart Industrial, Smart Water, Smart Environment y Smart City. Pertenece al Grupo Gimeno.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[18].

4.2.7.2. ¿Qué ofrece?

Cabe destacar que esta plataforma está disponible para su uso exclusivo para aquellos sensores producidos a su vez por Grupo Gimeno, entre los que podemos encontrar sensores de sonido, de residuos, de humedad y temperatura y calidad del aire entre otros. Permite integrar todos los sensores del cliente en una misma plataforma que se encarga de gestionar dichos dispositivos a gran escala de una forma sencilla y eficiente

Aseguran que su plataforma IoTsens Cloud Platform facilita la toma de decisiones inmediatas y eficientes en las ciudades inteligentes en las cuales se utiliza, permitiendo una gestión inteligente de los recursos disponibles. Gracias a esto, consiguen ofrecer valor añadido a los ciudadanos mejorando su calidad de vida a través de proporcionar servicios eficientes, colaborativos y de calidad en su entorno.

La plataforma IoTsens ofrece funcionalidades entre las que destacan, que su configuración sea abierta para que de este modo pueda ser fácilmente integrada con sistemas de terceros y, que su estructura sea modular lo que le permite que cada una de las capas que la componen se personalicen y evolucionen horizontalmente de manera transversal al resto de capas. A su vez, permite el análisis en tiempo real de los datos que recibe, como realizar cálculos pesados y

[18] https://www.iotsens.com/
https://www.iotsens.com/producto/plataforma-iotsens/
(Última fecha de consulta: 20/05/2022)

procesos de aprendizaje automático que sirven de apoyo a la hora de determinar los KPIs (*Key Performance Indicator*, indicador clave de rendimiento, los cuales son una serie de métricas que se utilizan para sintetizar la información sobre la eficacia y productividad de las acciones que se lleven a cabo en un negocio, en este caso la plataforma, con el fin de poder tomar decisiones y determinar aquellas que han sido más efectivas a la hora obtener la máxima efectividad). Los mecanismos de comunicación que emplean los sensores de esta compañía para acceder a la plataforma se basan en protocolos abiertos, más concretamente en LoRaWAN.

Entre el catálogo de soluciones del que dispone encontramos:

- *Ciudad*: Solución End-to-End que proporciona la información necesaria para la toma inteligente de decisiones en entornos urbanos creando así ecosistemas Digitales, Conectados y Sostenibles.
- *Agua*: Solución integral para la gestión del agua en diferentes ámbitos de aplicación con el fin de digitalizar las instalaciones y conseguir una mayor eficiencia en su gestión.
- *Edificios*: Solución para la gestión y control de edificios a través de la recopilación de información relevante con el fin de aumentar la eficiencia, la seguridad y la accesibilidad del edificio.
- *Irrigación*: Solución integral para la automatización de las zonas de riego con el fin de obtener la información necesaria para la toma inteligente de decisiones en cultivos y jardines resultando en optimización de recursos y eficiencia en su gestión.

4.2.7.3. Precios

IoTsens no ofrece una tabla de precios y no facilita una estimación según una métrica definida. Es necesario contactar con ellos vía email para obtener información al respecto (se realizó una consulta para ello, pero no se recibió respuesta).

Cabe destacar que la información ofrecida por esta empresa acerca de su plataforma y su funcionamiento es muy escueta y pobre, a lo que se puede añadir el defecto de que no cuentan con una documentación para aprender sobre ella.

4.2.8. Kaa

4.2.8.1. ¿Qué es?

Ilustración 18. Logo de Kaa

Kaa es una plataforma de IoT como servicio (PaaS) de extremo a extremo aplicable a cualquier escala de proyectos IoT empresariales. Se monta desde pequeñas empresas emergentes hasta grandes empresas y tiene modelos de implementación avanzados para soluciones de IoT de múltiples nubes. Se basa principalmente en microservicios flexibles y su principal ventaja es que se puede empezar a utilizar en cuestión de segundos ya que ha sido diseñada para con el objetivo de aportar una experiencia de usuario optimizada, dando acceso a casi todas las principales capacidades de Kaa, sin necesidad de conocimientos técnicos previos.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[19].

4.2.8.2. ¿Qué ofrece?

Ofrece características que permiten construir aplicaciones, manejar dispositivos conectados a traves de la nube, procesamiento de datos end-to-end (e2e, proceso que lleva un sistema o servicio de principio a fin y ofrece una solución funcional completa, normalmente sin necesidad de obtener nada de un tercero), analizar la telemetría de los dispositivos, y mucho más. Todas sus características son implementadas mediante microservicios. Kaa se basa en una arquitectura de microservicios flexible, es decir, que se puede configurar por separado cada característica, añadir más y/o agregar otras de 3ros. El siguiente diagrama ilustra la arquitectura funcional general de Kaat.

[19] https://www.kaaiot.com/
https://www.kaaiot.com/products/overview
https://www.kaaiot.com/advantages/platform
https://docs.kaaiot.io/KAA/docs/current/Welcome/
(Última fecha de consulta: 23/05/2022)

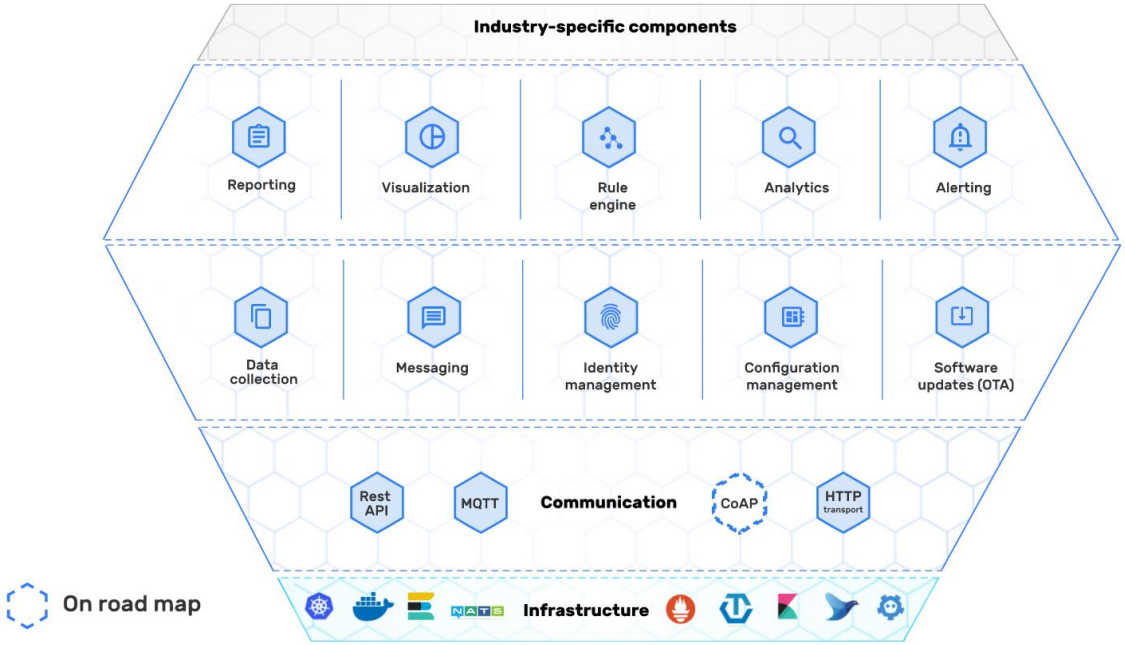

Ilustración 19. Arquitectura funcional de Kaa

Entre las características/funciones que ofrece podemos encontrar:

Comunicación Dispositivos-Cloud

La plataforma Kaa es compatible con los protocolos de comunicación ligeros para la conexión de dispositivos, como MQTT y HTTP. Al mismo tiempo, Kaa es *transport-agnostic* (agnóstica en cuanto al transporte) es decir, admite cualquier protocolo de trasmisión que se haya creado para emplearlo para comunicación añadiéndolo a la plataforma.

Si se da el caso en el que un dispositivo no tiene conexión IP, Kaa utiliza una arquitectura de Gateway (Puerta de enlace) donde el *gateway* se comunica con el dispositivo mediante un protocolo de red local o de proximidad y se encarga de enviar el mensaje a la nube (conversión de mensaje a nivel de transporte).

Gestión de dispositivos

Kaa habilita la función de administración de dispositivos proporcionando un registro de *digital twins* (gemelos digitales) que representan: cosas, dispositivos y otras entidades manejadas por la plataforma. Permite almacenar características de dispositivos como MAC, localización, versión de software, gracias a lo cual se pueden filtrar los dispositivos al realizar búsquedas. Kaa rastrea el dispositivo a lo largo de su ciclo de vida

Recopilación de datos

Kaa proporciona un protocolo para recopilar datos de dispositivos conectados, el cual garantiza la entrega fiable de datos. En caso de que durante el procesamiento de datos algo falle,

notifica al dispositivo para que no elimine esos datos y posteriormente sean reenviados y dado el caso de una comunicación exitosa, eliminados.

Para no sobrecargar la red, el protocolo admite el procesamiento por lotes. Brinda a sus dispositivos la capacidad de almacenar datos en búfer localmente antes de cargarlos en un mensaje, en adición, los gateways pueden realizar la función de almacenamiento y reenvió, lo cual es de gran utilidad en implementaciones de IoT con conectividad intermitente.

Permite recopilar datos estructurados y no estructurados (desde tipos primitivos a mapas clave-valor)

Procesamiento y análisis de datos

La plataforma cuenta con adaptadores de recopilación de datos que permiten enviar datos a varias bases de datos o sistemas de análisis de datos como Cassandra, MongoDB o InfluxDB entre otros.

Los datos crudos y no estructurados también pueden transformarse en series temporales bien estructuradas, lo que resulta conveniente para la analítica, el análisis de patrones, la visualización, la elaboración de gráficos, etc.

Visualización de datos

La función de visualización de datos de Kaa comprende un amplio conjunto de widgets (todos ellos configurables), como indicadores, gráficos, mapas, tablas, etc… Los widgets también permiten interactuar con los dispositivos enviando comandos, cambiando la configuración y los metadatos, etc… Cabe destacar que Kaa puede conectarse a herramientas de exploración y visualización de datos de terceros si es que sus API son abiertas para la integración de sistemas de terceros.

Actualizaciones Over the air

Permite implementar y distribuir nuevas características IoT para los dispositivos que ya están en uso.

Seguridad

Por defecto, la comunicación Kaa con los dispositivos está asegurada con TLS o DTLS. Los dispositivos han de autentificar que sus credenciales sean válidas con el servidor mediante *pre-shared keys* (claves precompartidas), tokens, combinaciones de *login* y *password* (contraseñas), certificados TLS).

Asimismo, Kaa cuenta con una gestión flexible del ciclo de vida de las credenciales, que permite aprovisionar, suspender/desconectar y revocar las credenciales del dispositivo.

4.2.8.3. Precios

Kaa ofrece diferentes opciones y rangos de precios a la hora de contratar un plan de servicios los cuales se segmentan en función del tipo de hospedaje que se le dará a la plataforma:

- En la nube, Plataforma IoT como servicio (PaaS): Kaa Cloud
- Instancia privada única de la plataforma alojada por KaaIoT en una infraestructura dedicada: KaaIoT-Hosted
- Instancias privadas ilimitadas alojadas en la infraestructura de elección del cliente: Self-Hosted

Subscription plans and pricing

KAA CLOUD | KAAIOT-HOSTED | SELF-HOSTED

Kaa Cloud Plans	5	15	50	100	250	500	1000
Endpoints included	5	15	50	100	250	500	1000
Plan fee	14 Days Free	$14.99	$44.99	$79.99	$175	$325	$625
Price per additional active device / month	n/a	n/a	n/a	n/a	n/a	n/a	$0.5
Storage size	5 GB	15 GB	50 GB	100 GB	250 GB	500 GB	1000 GB

KAA CLOUD | KAAIOT-HOSTED | SELF-HOSTED

KaaIoT-Hosted Plans	250	500	1000	2500
Endpoints included	250	500	1000	2500
Plan fee	$499	$749	$1,199	$1,999
Price per additional active device / month	$1.5	$1.25	$1.00	$0.65
Storage size	250 GB	500 GB	1000 GB	2500 GB
Virtual machine configuration	3 x (4 vCPU cores, 8 GB RAM)	3 x (4 vCPU cores, 8 GB RAM)	3 x (4 vCPU cores, 8 GB RAM)	7 x (4 vCPU cores, 8 GB RAM)

KAA CLOUD | KAAIOT-HOSTED | SELF-HOSTED

License terms	25	100	250	500	1000	2500	Unlimited
White labeling			✓	✓	✓	✓	✓
Hosting in your clients' infrastructure			Optional	Optional	Optional	Optional	Optional

Ilustración 20. Tabla de precios de las diferentes opciones que ofrece Kaa IoT Platform

4.2.9. Macchina.IO EDGE

4.2.9.1. ¿Qué es?

Ilustración 21. Logo de Macchina.IO

Macchina.IO EDGE simplifica el desarrollo de aplicaciones IoT edge escalables con un framework potente y ligero para dispositivos basados en Linux, como Raspberry Pi, Beaglebone o similares. Proporciona las herramientas necesarias para la creación de aplicaciones y para que estas se comuniquen con diversos sensores, dispositivos y servicios en la nube.

Se trata de una Open Service Platform(OSP) la cual permite la creación, el despliegue y la gestión de aplicaciones modulares y dinámicamente extensibles, basadas en un modelo de plug-in y servicios. Las aplicaciones creadas con OSP pueden ampliarse, actualizarse y gestionarse incluso cuando se despliegan sobre el terreno. Cabe destacar que no proporciona una solución de hosting, por lo tanto, la plataforma siempre se ejecutará en una nube contratada por el usuario o bien en una privada.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[20].

[20] https://macchina.io/index.html
https://docs.macchina.io/edge/00100-MacchinaIntroduction.html
https://macchina.io/resources.html#remote
https://macchina.io/blog/internet-of-things/provide-secure-remote-access-iot-edge-devices/
(Última fecha de consulta: 26/05/2022)

4.2.9.2. ¿Qué ofrece?

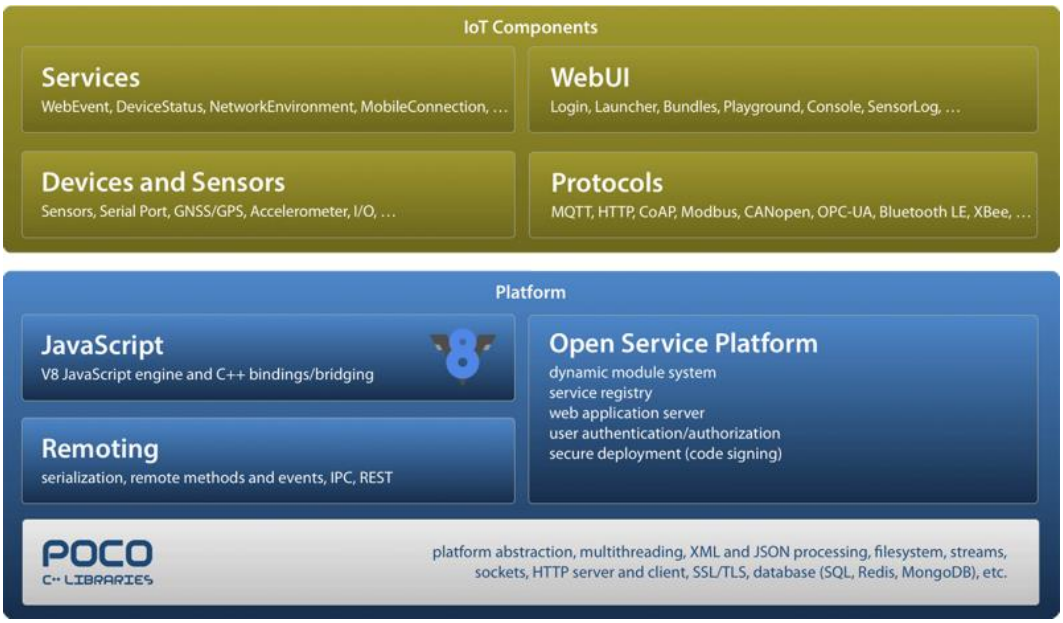

Ilustración 22. Arquitectura de Macchine.IO EDGE

Como se muestra en la Ilustración 22, EDGE se compone de 2 agrupaciones, los componentes IoT, donde se encuentra todo lo relacionado con la interacción directa con los dispositivos, y la plataforma, dentro de la cual se agrupan las herramientas necesarias para la recolección y análisis de datos

Implementa un entorno de ejecución de JavaScript y C++ habilitado para la web, modular y extensible, y proporciona bloques de construcción fácilmente disponibles y fáciles de usar que permiten que las aplicaciones se comuniquen con varios sensores y dispositivos, así como con servicios en la nube.

Permite realizar las aplicaciones para dispositivos de manera independiente del hardware en el que se va a utilizar, de esta forma se facilita el cambio de plataforma de hardware o la compatibilidad con varias plataformas de hardware o dispositivos diferentes con una única base de código. A su vez, define interfaces genéricas para varios tipos de sensores y dispositivos. Sobre la base de estas interfaces, existen diferentes implementaciones que permiten disponer de sensores y dispositivos específicos en macchina.io EDGE. Existen interfaces e implementaciones para tipos de sensores genéricos como sensores de temperatura o humedad, receptores GNSS/GPS, acelerómetros, disparadores, puertos GPIO, dispositivos de puerto serie, etc…

EDGE posee una amplia compatibilidad con protocolos que le permiten comunicarse con redes de sensores, sistemas de automatización o servicios en la nube. Incluye soporte para HTTP, MQTT, REST, JSON-RPC, SOAP, UPnP™, Modbus, OPC-UA, CANopen, S7, etc. En cuanto al

almacenamiento de los datos recibidos, la OSP utiliza SQLite como base de datos integrada la cual está disponible para código JavaScript como C++. En ella se registran los datos de los sensores.

En cuanto a su funcionamiento, mediante los componentes IoT facilitados por Macchina.IO se definen los dispositivos y los sensores a través de las interfaces genéricas mencionadas previamente. Estas interfaces se encargarán de mapear la distribución de los puertos GPIO y los sensores conectados a los dispositivos.

La plataforma se comunica con los dispositivos de internet de las cosas mediante diferentes protocolos, para posteriormente, una vez establecida la comunicación, recoger la información y la guardará en la base de datos para más adelante exponer la información o analizarla.

4.2.9.3. Precios

Macchino.IO EDGE cuenta con 2 planes de precios, el primero, gratuito, el Open Source (GPL) está orientado para evaluar, experimentar y construir una prueba de concepto con la plataforma, mientras que el plan de pago Licencia Comercial, es para empresas que construyen dispositivos IoT profesionales. Este último plan cuenta con una serie de características que no ofrece el gratuito como pueden ser:

- Soporte REST, SOAP, JSON-RPC y HTTP para el C++ Remoting framework.
- APIs REST para la gestión de dispositivos.
- Generador de código WSDL y XML Schema (XSD) a C++.
- Soporte OPC-UA y CANopen para la integración de dispositivos de automatización industrial.
- Autenticación y autorización de usuarios avanzada, basada en bases de datos, con integración opcional de LDAP.

4.2.10. Mainflux

4.2.10.1. ¿Qué es?

Ilustración 23. Logo de Mainflux

Mainflux es una plataforma en la nube IoT moderna, escalable, segura, de código abierto y libre de patentes escrita en Golang y puede desplegarse en un modelo local, híbrido o en la nube. Se utiliza como middleware de IoT para crear soluciones de IoT complejas. La plataforma también se puede ejecutar en el edge. La implementación de Mainflux en un *gateway* le permite recopilar, almacenar y analizar datos, organizar y autenticar dispositivos.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[21].

4.2.10.2. ¿Qué ofrece?

Esta plataforma de IoT de código abierto libre de patentes ofrece un gran número de herramientas ventajosas para la recopilación y gestión de datos y programación de eventos. Ofrece comunicación bidireccional (plataforma – dispositivos IoT/gateways) a través de los protocolos PUB/SUB (Servidor HTTP), MQTT Broker, WebSocket y CoAP.

En la Ilustración 23Ilustración 24 se muestran los diferentes microservicios que conforman la arquitectura de Mainflux IoT Platform, los cuales se agrupan en:

- Users: Administra los usuarios de la plataforma y las políticas de autenticación.
- Things: Administra las cosas de la plataforma, los canales (representa un canal de comunicación. Sirve como *topic* de mensaje que puede ser consumido por todos los dispositivos conectados a él) y las políticas de acceso.

[21] https://mainflux.com/cloud.html
https://docs.mainflux.io/
(Última fecha de consulta: 29/05/2022)

- *Protocol adapters* (Adaptador del protocolo de comunicación): Proporcionan interfaces (HTTP, MQTT, WebSockets, CoAP, OPC-UA y LoRa) para acceder a los canales de comunicación
- NGINX: Proxy inverso con reenvío de autenticación

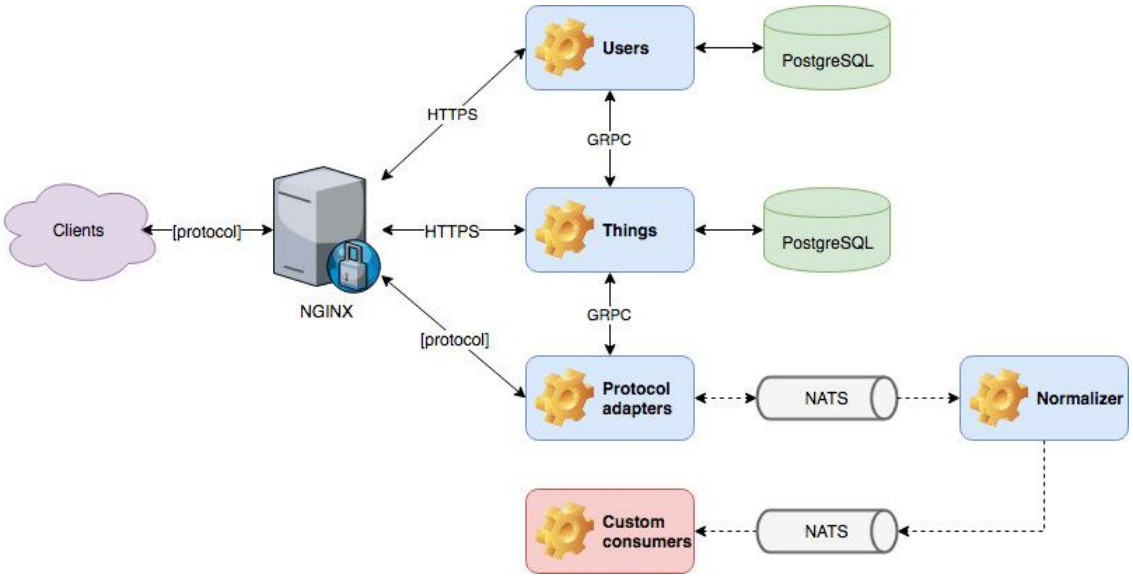

Ilustración 24. Concepto general de la arquitectura de la plataforma Mainflux IoT

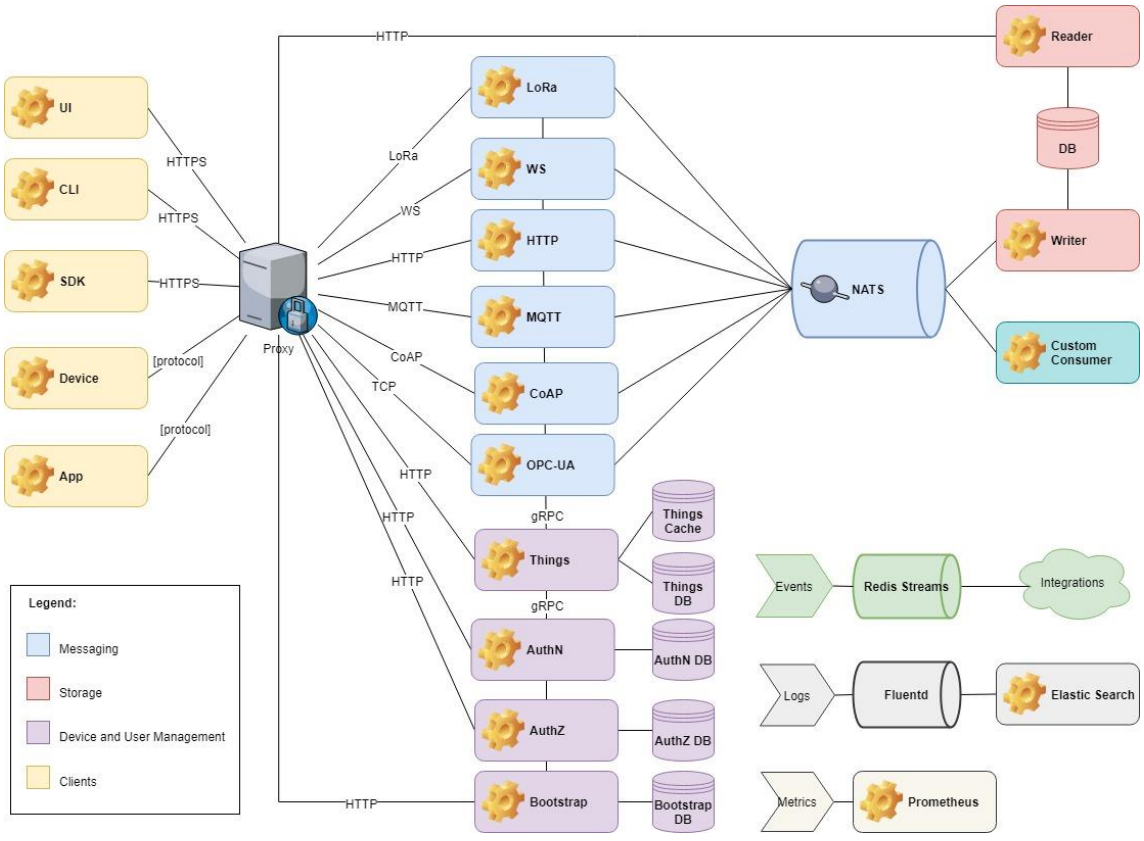

Ilustración 25. Arquitectura mostrada en profundidad de la plataforma Mainflux IoT

En la Ilustración 25 observamos la arquitectura mostrada en la más atrásIlustración 24, a los conjuntos de *users, things, protocol adapters* y el proxy NGINX podemos añadir:

- *Normalizer* (Normalizador): Normaliza los mensajes SenML y genera el flujo de mensajes "procesados"
- NATS: Bus de eventos del sistema

A continuación, profundizaremos en los servicios que ofrece Mainflux

Servicio de aprovisionamiento (Provision Service)

Es el proceso de configuración de una plataforma IoT en el que el operador del sistema crea y configura las diferentes entidades utilizadas en la plataforma, es decir los usuarios (representa al usuario real (humano) del sistema, canales (como se definió anteriormente, representan los canales de comunicación donde se publican y son consumidos los mensajes enviados por los dispositivos/cosas) y cosas (dispositivos IoT/aplicaciones conectadas a Mainflux).

Es parte del proceso de configuración de las aplicaciones IoT donde conectamos los dispositivos edge con la plataforma en la nube. Para el aprovisionamiento podemos utilizar Mainflux CLI, una interfaz de línea de comandos que se puede descargar como un *asset* (activo) por separado, o puede ser construido con la ayuda de la herramienta *GNU Make*.

Protocolos de comunicación

Una vez que se aprovisiona un canal y un dispositivo se conecta a él, puede empezar a publicar mensajes en él.

- HTTP. API Documentation
- MQTT
- WebSocket. Mainflux soporta MQTT-over-WS, en lugar de un protocolo WS puro, lo que brinda numerosos beneficios para las aplicaciones de IoT, como las características de QoS y PUB/SUB.
- CoAP. CoAP (Constrained Application Protocol) es un protocolo de software que se encuentra en el nivel de capa de aplicación del modelo OSI y está apuntado a correr en dispositivos simples, permitiendo que puedan comunicarse sobre internet.
  El CoAP-adapter implementa el protocolo CoAP utilizando UDP subyacente. Para enviar y recibir mensajes a través de CoAP, puede utilizar CoAP CLI.

Almacenamiento

Mainflux admite varias bases de datos de almacenamiento en las que se almacenan los mensajes:

- CassandraDB
- MongoDB
- InfluxDB
- PostgreSQL

Los mensajes son almacenados en formato SenML y JSON. Para ello Mainflux ofrece dos tipos de transformadores, normalizadores, uno para cada tipo mencionado previamente.

Seguridad

Mainflux es un sistema altamente seguro. Tiene un servicio de autenticación (basada en token) y autorización con control de acceso detallado TLS y DTLS (HTTPS, WSS, MQTT con TLS, CoAP con DTLS) dedicado que protege el sistema de accesos no autorizados y dispositivos y aplicaciones no autorizados.

Todos los mensajes y todo el tráfico de la red proveniente o hacia Mainflux están encriptados según los últimos estándares de seguridad (TLS v1.3).

Twins Service

El servicio Twins escucha el servidor NATS e intercepta los mensajes que pasan a través del agente NATS. Cada mensaje de Mainflux contiene información sobre el subcanal y el tema utilizado para enviar un mensaje. El servicio Twins compara esta información con las definiciones de atributos de los gemelos que se conservan en la base de datos, obtiene los gemelos correspondientes y actualiza sus respectivos estados

4.2.10.3. Precios

Mainflux ofrece planes de precios en los cuales los precios pueden variar, desde modos de instalación y planes de soporte absolutamente gratuitos hasta variantes empresariales personalizadas totalmente gestionadas.

- Gratis: Mainflux es una plataforma IoT de código abierto (Apache v2.0) que puedes alojar en cualquier plataforma compatible con Linux o Docker.
- 500$/mes - 5000$/mes incluso más: Desplegamos, alojamos y gestionamos Mainflux en una nube privada dedicada, siguiendo las directrices de DevOps con multi-AZ, autoescalado, cloudwatch, snapshots y sandboxes bajo demanda.

4.2.11. Microsoft Azure IoT

4.2.11.1. ¿Qué es?

Ilustración 26. Logo de Microsoft Azure

Internet de las cosas (IoT) de Azure es un conjunto de servicios en la nube administrados por Microsoft que permiten conectar, supervisar y controlar miles de millones de recursos de IoT. También incluye seguridad y sistemas operativos para dispositivos y equipos, junto con datos y análisis que ayudan a las empresas a crear, implementar y administrar aplicaciones de IoT.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[22].

4.2.11.2. ¿Qué ofrece?

Gracias a las tecnologías y servicios de IoT Azure que se ofrecen, Microsoft permite crear una amplia variedad de soluciones IoT mediante principalmente 2 ofertas:

- *Application platform as a service* (*aPaaS*) (Plataforma de aplicaciones como servicio) **IoT Central**.
- *Platform as a service* (*PaaS*) (Plataforma como servicio) **IoT Hub**.

En primer lugar, explicaremos ambos tipos de plataformas. *aPaaS* proporciona un entorno en la nube para compilar administrar y entregar aplicaciones a los clientes. Las ofertas de *aPaaS* se encargan del escalado y de la mayor parte de la configuración. En cambio, *PaaS* es un modelo de computación en la nube en el que se adaptan las herramientas de hardware y software de Azure

[22] https://azure.microsoft.com/es-es/overview/iot/#overview
Documentación IoT Hub: https://docs.microsoft.com/es-es/azure/iot-hub/
https://docs.microsoft.com/es-es/security/benchmark/azure/baselines/iot-hub-security-baseline?toc=%2Fazure%2Fiot-hub%2FTOC.json
https://azure.microsoft.com/es-es/pricing/details/iot-hub/
Documentación IoT Central: https://docs.microsoft.com/es-ES/azure/iot-central/
https://azure.microsoft.com/es-es/services/iot-central/#overview
(Última fecha de consulta: 06/07/2022)

a una tarea o función en específico. Al contrario que en *aPaaS*, en *PaaS* es el desarrollador el que ha de hacerse responsable de gestionar el escalado y la configuración.

IoT Central acelera el ensamblado y el funcionamiento mediante la unión previa de componentes de plataforma como servicio (PaaS). Con su interfaz de usuario (*UI*) web y una API REST listas para usar, puede supervisar fácilmente las condiciones de dispositivos y gestionar millones de ellos y sus datos de forma remota a lo largo de sus ciclos de vida. Además, puede actuar sobre la información del dispositivo mediante la extensión de *IoT Intelligence* en aplicaciones de línea de negocio. Azure IoT Central también ofrece recuperación ante desastres integrada, multiinquilino y disponibilidad global.

Dado el caso en el que el desarrollador precise de un mayor grado de control y personalización que el que ofrece Azure IoT Central, es aquí donde destaca IoT Hub, permitiendo crear soluciones IoT personalizadas desde cero o ampliar una creada mediante IoT Central.

4.2.11.3. IoT Hub

Azure IoT Hub es un servicio administrado, hospedado en la nube, que actúa como centro de mensajes para la comunicación entre una aplicación de IoT y los dispositivos conectados. Puede conectar millones de dispositivos y sus soluciones de back-end de forma fiable y segura. La mayoría de los dispositivos se pueden conectar a un centro de IoT.

Se admiten varios patrones de mensajería, como la telemetría entre dispositivos y la nube (*device-to-cloud telemetry*), carga de archivos desde los dispositivos y métodos de solicitud-respuesta para controlar los dispositivos desde la nube. IoT Hub también admite la monitorización para ayudar a realizar un seguimiento de la creación de dispositivos, las conexiones de dispositivos y fallos de los mismos. En adición, se adapta a millones de dispositivos conectados simultáneamente y a millones de eventos por segundo para soportar sus cargas de trabajo de IoT.

Puede integrar IoT Hub con otros servicios de Azure para crear soluciones *end-to-end* completas. Por ejemplo:

- *Azure IoT Device Provisioning Service* para el aprovisionamiento automatizado de dispositivos.
- *Azure Time Series Insights* para almacenar y analizar datos de series temporales de rutas de acceso activas, en tiempo real (*Hoth path*) e inactivas (*Cold path*) de dispositivos IoT.
- *Azure Stream Analytics* para ejecutar cálculos analíticos en tiempo real sobre los datos que fluyen desde los dispositivos conectados.
- *Azure IoT Edge* para ejecutar *IA* (Inteligencia Artificial), servicios de terceros o su propia lógica de negocios en dispositivos IoT Edge.
- *Azure Event Grid* para permitir que la empresa pueda reaccionar rápidamente ante eventos críticos de forma fiable, escalable y segura.

- *Azure Logic Apps* para automatizar los procesos de negocio.
- *Azure Machine Learning* para agregar modelos de aprendizaje automático e IA a la solución.

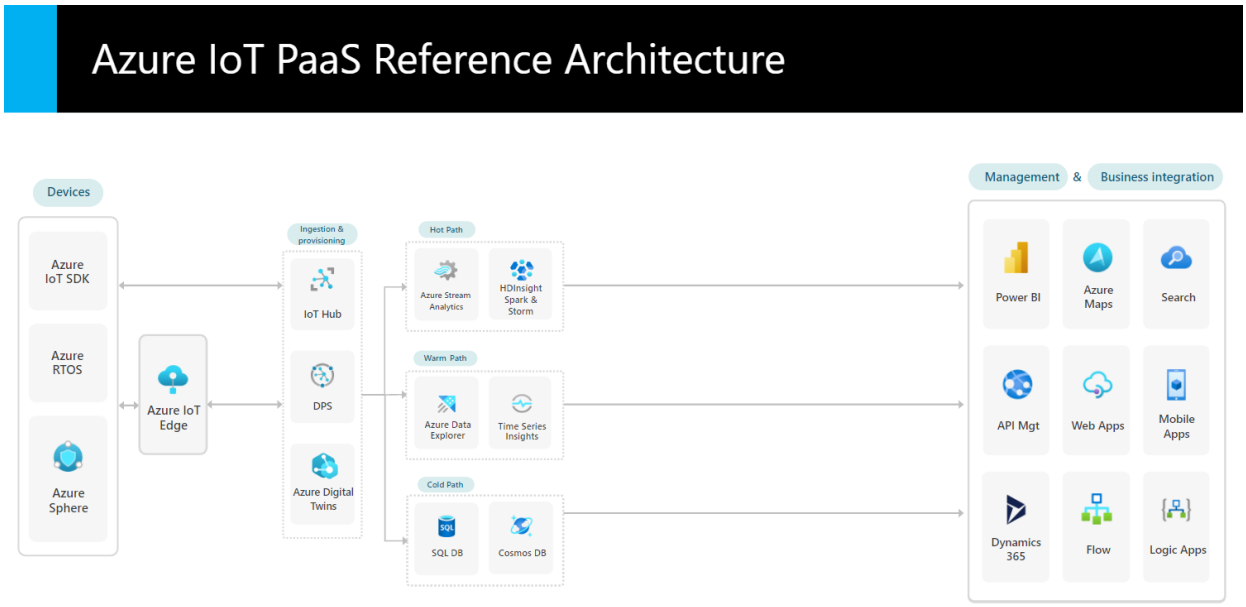

Ilustración 27. Diagrama de los servicios Azure en una arquitectura de IoT basada en PaaS

Las propiedades básicas de la funcionalidad de mensajería de IoT Hub son la confiabilidad y durabilidad de los mensajes. Estas propiedades permiten la resistencia a la conectividad intermitente en el dispositivo y a los picos de carga del procesamiento de eventos en la nube. Implementa al menos una vez garantías de entrega para la mensajería del dispositivo-nube y de la nube-dispositivo. IoT Hub permite la recepción de mensajes de aquellos dispositivos que sean compatibles con los protocolos de comunicación siguientes: MQTT, MQTT sobre WebSockets, AMQP, AMQP sobre WebSockets y HTTPS.

En cuanto al ámbito de la seguridad, en lo referente a la autenticación e identidad de los dispositivos, cada IoT Hub tiene un registro de identidades que almacena información acerca de los dispositivos y módulos que pueden conectarse a él. Para que un dispositivo o un módulo se pueda conectar, debe haber una entrada para ese dispositivo o módulo en el registro de identidades del IoT Hub.

Un dispositivo o un módulo se puede autenticar en el centro de IoT en función de las credenciales almacenadas en el registro de identidades. Se admiten dos métodos de autenticación entre el dispositivo e IoT Hub. Puede usar una:

- Autenticación basada en tokens de SAS
- Autenticación de certificado X.509

Respecto a la seguridad en la comunicación dispositivo-nube, después de seleccionar el método de autenticación, la conexión a Internet entre el dispositivo IoT e IoT Hub se protege con el estándar de Seguridad de la capa de transporte (TLS).

4.2.11.4. IoT Central

Como se ha comentado previamente, IoT Central es un conjunto de capacidades *ready-made* que permiten tener una plataforma de desarrollo de aplicaciones como servicio (*aPaaS*) para construir aplicaciones extensibles y totalmente administradas hospedada en Azure. Está construida sobre todas las funcionalidades de Azure, entre las que destacan las funciones de supervisión y gestión de dispositivos para conectar, reconfigurar y actualizarlos.

El módulo incluye numerosas plantillas de aplicaciones para diferentes sectores con el fin de acelerar la velocidad de desarrollo. Si se combina con Azure IoT Hub, permite crear aplicaciones más complejas, capaces de soportar millones de dispositivos. Esta solución es ideal para las empresas que no quieren dedicar muchos recursos a la arquitectura del sistema [Ilustración 28]

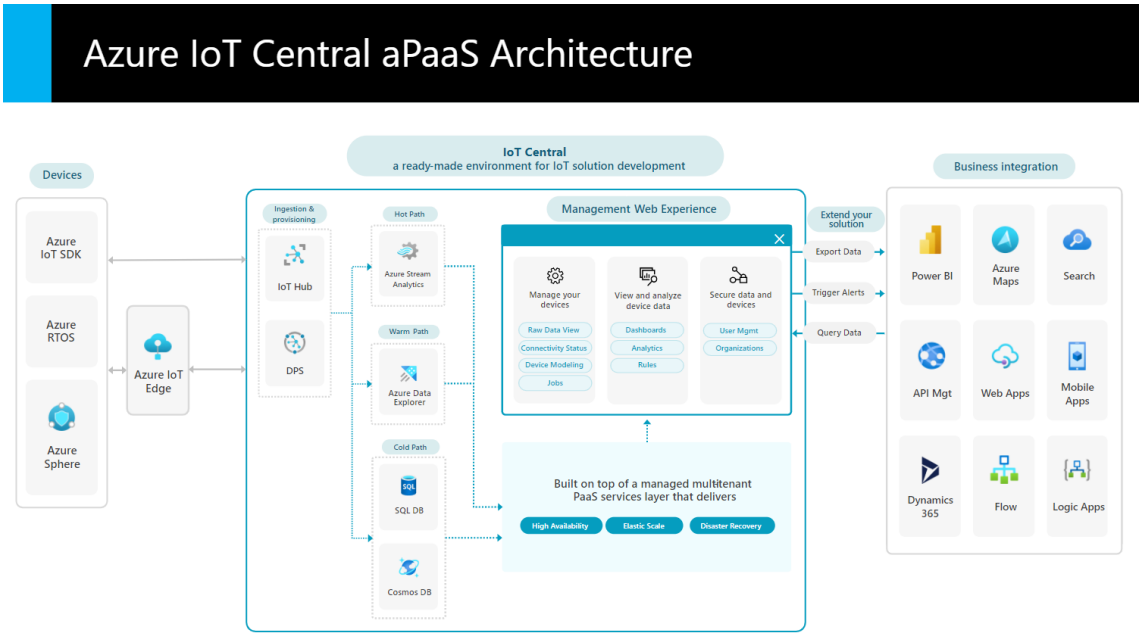

Ilustración 28. Diagrama de una arquitectura basada en IoT Central

El flujo de eventos seguidos en este diagrama sería el siguiente. En primer lugar, IoT Central recibe eventos de dispositivos y telemetría a través de los SDK de dispositivos de Azure IoT, Azure RTOS, Azure Sphere o Azure IoT Edge.

En este ejemplo de arquitectura basada en IoT Central, la plataforma está compuesta por varios servicios *PaaS* de Azure gracias a los cuales proporciona las siguientes funcionalidades desde el inicio:

- Servicios de ingesta y aprovisionamiento de datos.

- Almacenamiento y análisis de datos de acceso frecuente, intermedio y en frío.
- Una capa *PaaS* gestionada que ofrece alta disponibilidad y recuperación de desastres (*HADR*, *High Availability Disaster Recovery*) y escalado elástico.
- Una experiencia de usuario web de gestión que le permite gestionar dispositivos con la vista de datos sin procesar (datos brutos), el estado de conectividad, el modelado de dispositivos y los trabajos; ver y analizar los datos de los dispositivos con paneles, análisis y reglas y proteger los datos y los dispositivos con la gestión de usuarios y organizaciones.

Es crucial entender que, al ser una solución integral, IoT Central elimina la necesidad de preocuparse por detalles técnicos a nivel de infraestructura. A través de sus características y capacidades, la plataforma se encarga de las preocupaciones más comunes relacionadas con la gestión de dispositivos y telemetría, permitiendo a las empresas centrarse en obtener el máximo valor de sus datos y dispositivos.

IoT Central amplía las soluciones activando alertas, exportando datos y apoyando las consultas de datos. En adición, se puede integrar con aplicaciones de línea de negocio como Power BI, Azure Maps, Search, API Management, Web Apps, Mobile Apps, Dynamics 365, Flow o Logic Apps.

4.2.11.5. Precios

El costo de Azure IoT Hub y Azure IoT Central varía dependiendo del nivel de servicio, del número de mensajes enviados y de la ubicación del centro de datos de Azure utilizado. Ambos servicios ofrecen un nivel gratuito para usos básicos. A medida que aumenta el uso, como el número de mensajes o de dispositivos conectados, las tarifas también aumentan.

Es importante tener en cuenta que, al igual que con otras soluciones basadas en Azure, otros servicios que se utilicen en conjunto con IoT Hub o IoT Central como Azure Stream Analytics o Azure Data Lake, incurrirán en costos adicionales.

Es recomendable consultar la página de precios de Azure[23] para obtener la información más actualizada y detallada acerca del costo de Azure IoT Hub, Central y de los otros servicios de Azure.

[23] Calculador de precios: https://azure.microsoft.com/en-us/pricing/calculator/
(Última fecha de consulta: 06/07/2022)

4.2.12. OpenRemote

4.2.12.1. ¿Qué es?

Ilustración 29. Logo OpenRemote

OpenRemote es una plataforma IoT 100% de código abierto que permite crear una amplia gama de aplicaciones profesionales como por ejemplo gestión de energía, multitudes o de activos más genéricos. Toda la documentación consultada para realizar este análisis se encuentra a pie de página[24].

4.2.12.2. ¿Qué ofrece?

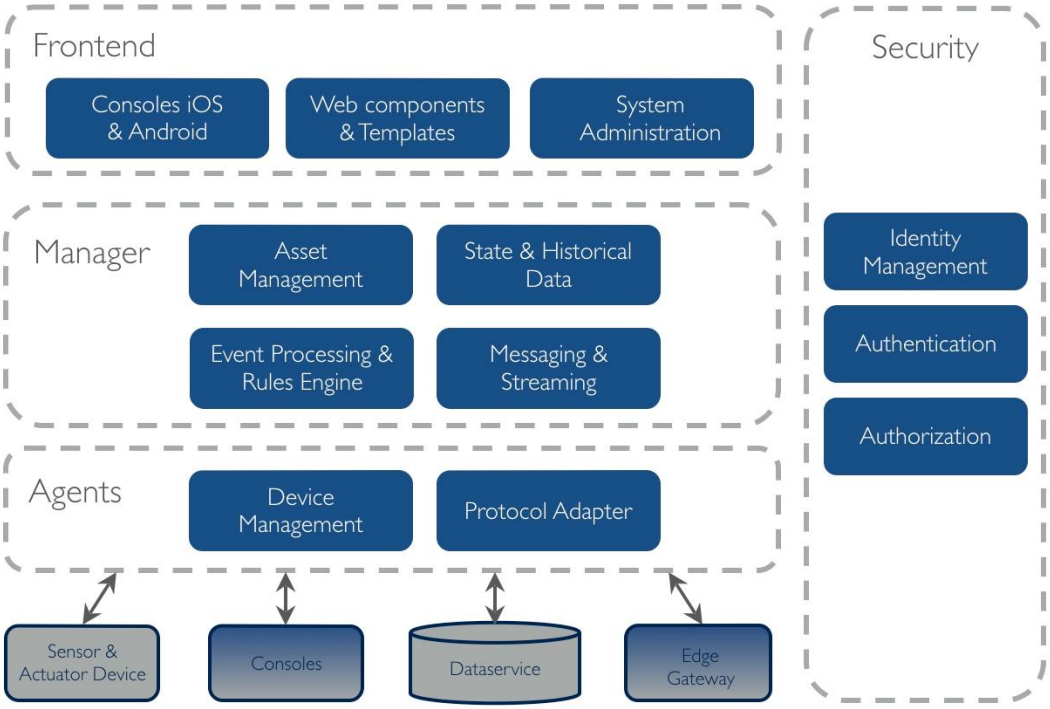

Ilustración 30. Arquitectura de OpenRemote

[24] https://openremote.io/
https://github.com/openremote/openremote/wiki
(Última fecha de consulta: 13/06/2022)

Como se puede observar en la Ilustración 30 la arquitectura de esta plataforma se estructura en 4 módulos diferentes. El apartado de *Manager* (Núcleo del sistema OpenRemote, una aplicación Java autónoma que forma un intermediario de contexto de IoT que captura el estado actual de los activos del sistema) proporciona la API para monitorear y administrar el sistema:

- API HTTP basada en JAX-RS: Request-Response API con documentación en vivo disponible a través de Swagger UI.
- API de eventos de WebSockets: API de publicación y suscripción basada en eventos, donde los eventos son del tipo SharedEvent.
- API de eventos MQTT (MQTT Broker): Publish-Subscribe API.

OpenRemote *Manager* tiene su propio motor de reglas. Las reglas ejecutan acciones cuando se detectan estados de activos o secuencias de eventos que coinciden. Por ejemplo, cuando la temperatura en una habitación varia. Estas reglas se pueden utilizar para una amplia gama de funciones, desde la simple vinculación de atributos hasta algoritmos complejos basados en reglas, modelos predictivos. Hay 3 tipos diferentes de lenguajes de programación de reglas:

- *Groovy rules*:

El lenguaje de *scripting* de reglas Groovy está disponible en el Manager (técnico) y puede utilizarse para reglas complejas.

- *WHEN-THEN rules*:

El modelo de objetos de reglas WHEN-THEN está pensado para que los usuarios de la aplicación puedan crear reglas de flujo de trabajo basadas en eventos, utilizando una interfaz de usuario front-end.

- *Flow rules*:

Modelo está diseñado para usuarios de aplicaciones que desean vincular atributos mediante una conversión. Su objetivo principal es permitir la vinculación de atributos (p. ej., un interruptor KNX con una luz Velbus) o el procesamiento de atributos para crear nuevos atributos 'virtuales' (p. ej., el consumo de energía es la suma de tres submedidores individuales).

OpenRemote *Frontend* se encarga de simplificar la creación y el despliegue de interfaces de usuario como:

- Tablero de control de múltiples inquilinos
- Panel de control domótico
- Tablero de control de ciudad inteligente

Dentro del apartado *Agents* (agentes), se encuentran las interfaces para las APIs de terceros y los protocolos de servicio que se emplean para que los dispositivos/cosas (*things*) se conecten al módulo *Manager*, es decir, los agentes son un tipo especial de activo (representación física o lógica de una *Thing*) que vincula servicios/dispositivos externos con su sistema OpenRemote a través de los diferentes protocolos aceptados por la plataforma (HTTP/TCP/IP/...). Existen 2 tipos de agentes:

- *Specialised agents* (Agentes especializados) (Velbus, Z-Wave, KNX, etc.)
- *Generic agents* (Agentes genéricos) (HTTP, TCP, UDP, WS, MQTT, etc.)

Por último, dentro de *Security* encontramos los servicios de Autenticación y Autorización, los cuales son gestionados mediante Keycloak. Concretamente, para la Autorización se emplea el protocolo estándar *OAuth 2.0*

4.2.12.3. Precios

La plataforma OpenRemote ofrece sus servicios de manera completamente gratuita, ya que es Open-Source. El gasto que conlleva esta solución depende del hosting escogido para el despliegue de la plataforma, sea en una nube privada o en otras tales como Amazon Web Services, Azure o Google Cloud.

4.2.13. Predix Platform

4.2.13.1. ¿Qué es?

Ilustración 31. Logo de GE Predix Platform

La plataforma Predix es una Plataforma como Servicio (PaaS) lanzada por GE basada en la nube enfocada en el sector industrial y análisis de datos del mismo. La plataforma ofrece las capacidades compartidas que requieren las aplicaciones industriales: conectividad de datos *edge-to-cloud* segura y escalable, análisis y aprendizaje de máquinas, procesamiento de grandes volúmenes de datos y gemelos digitales centrados en activos.

Diseñada como una plataforma de aplicación distribuida, la plataforma Predix está optimizada para la gestión de datos de alto volumen, baja latencia e integración intensiva y resultados basados en análisis.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[25]

[25] https://www.ge.com/digital/iiot-platform
https://www.ge.com/digital/documentation/predix-platforms/index.html
https://programmerclick.com/article/3834460536/
(Última fecha de consulta: 21/05/2022)

4.2.13.2. ¿Qué ofrece?

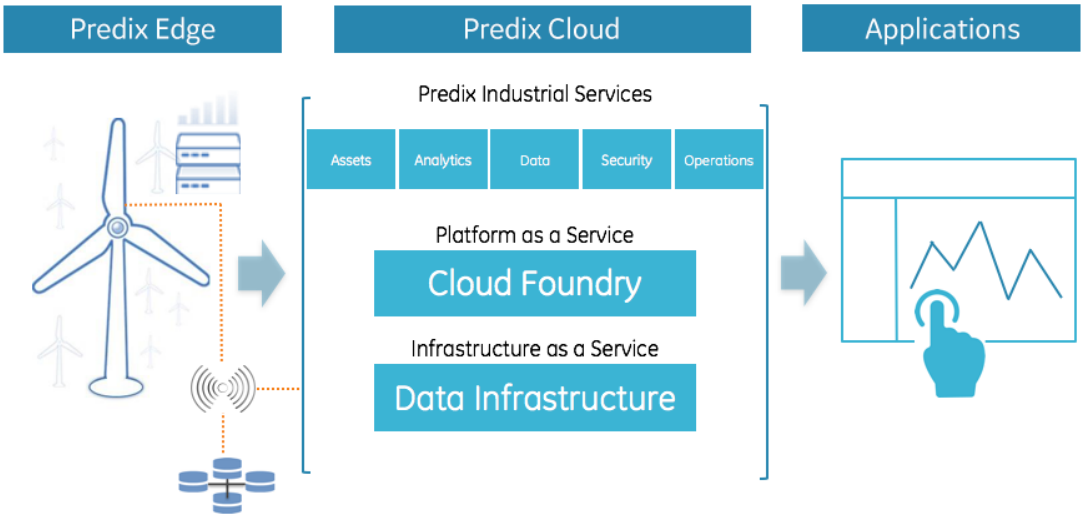

Ilustración 32. Arquitectura de GE Predix Platform

La arquitectura de la plataforma Predix aglutina los principales componentes de Predix: *Predix Edge, Predix Cloud* y *Applications/Services.*

Predix Edge es la capa de software responsable de recoger los datos de los activos industriales y enviarlos a *Predix Cloud*, así como de ejecutar las aplicaciones locales, como por ejemplo los *edge analytics* (análisis de borde) el cual se emplea para supervisar el estado de estos activos. Predix Edge está disponible en gateways y máquinas virtuales.

El *Predix Cloud* del lado de la plataforma es el núcleo de toda la solución Predix. Proporciona una gran cantidad de servicios como pueden ser, la recopilación y análisis de *big data* industrial, el modelado de *Digital Twins* o el desarrollo de aplicaciones industriales en torno a la idea de los datos industriales como núcleo. A continuación, se comentarán brevemente algunos microservicios que integra *Cloud*:

- *Infrastructure Predix*: Proporciona tres arquitecturas de implementación: nube pública (AWS, Azure), nube privada y Country Cloud. Por lo tanto, Predix Cloud admite la implementación privada.
- *Security:* Predix Cloud proporciona muchos mecanismos de seguridad, incluida la gestión de identidad, el cifrado de datos, la protección de aplicaciones, el registro y la auditoría.
- *Bus de datos:* Esta parte incluye funciones tales como la inyección de datos, el procesamiento y el almacenamiento de datos heterogéneos. Admite la importación y el procesamiento de datos de transmisión y datos por lotes.
- *Entorno de desarrollo de alta productividad*: Proporciona un entorno visual de desarrollo de aplicaciones que incluye Predix Studio para crear rápidamente aplicaciones industriales utilizando *drag and drop* (arrastrar y soltar).
- *Entorno de desarrollo de alto control*: Proporciona un entorno de desarrollo a nivel de código (basado en Cloud Foundry) junta a una serie de microservicios que pueden integrarse rápidamente;

- *Digital Twin Development Environment*: Proporciona herramientas de modelado rápido para el desarrollo de modelos, incluidos: modelos de dispositivos, modelos de análisis y bases de conocimiento.

La *Application* proporciona una gran variedad de servicios industriales como pueden ser predicción completa de fallas, estado del equipo, optimización de la eficiencia de producción, administración de energía, optimización de programación entre otro. Usando una combinación de datos y mecanismos, su objetivo es resolver los problemas de calidad, eficiencia y consumo de energía con el fin de ayudar a las empresas industriales.

4.2.13.3. Precios

Predix Platform y los servicios individuales Predix solo se encuentran incluidos para su compra en el paquete *Predix Essentials*. *Predix Essentials* es una versión empaquetada y preconfigurada de la *Predix Platform* destinada a dar soporte inmediato a las aplicaciones de GE Digital y a los casos de uso típicos del Industrial IoT,

GE no ofrece una estimación o tabla de precios en lo referente al paquete Predix Essentials. Es necesario contactar con ellos vía email para obtener información al respecto.

4.2.14. Sentilo

4.2.14.1. ¿Qué es?

Ilustración 33. Logo de Sentilo

Sentilo es una plataforma de sensores y actuadores de código abierto creada por por el Ayuntamiento de Barcelona SentiloBCN y diseñada para encajar en la arquitectura Smart City o IoT. Su objetivo es explotar la información "generada por la ciudad" y la capa de sensores desplegada para recoger y difundir esta información.

Sentilo está dirigido a municipios u organizaciones que necesitan procesar mucha información recibida del terreno generada por los diferentes tipos de dispositivos IoT que se encuentran desplegados por el terreno. A esta diversidad de hardware se le llamará heterogénea ya que en una ciudad pueden existir una gran cantidad de cosas/ sensores tanto de marcas, como con softwares o protocolos de comunicación diferentes.

Es gracias a esta heterogeneidad en los dispositivos por lo que fue concebida esta plataforma. Surgió con la idea principal de crear un servicio de gestión de datos e infraestructuras orientado a múltiples plataformas, huyendo de las soluciones TIC verticales organizadas en "silos", ya que estas soluciones de silo hacen que las ciudades dependan demasiado de tecnologías, soluciones o proveedores específicos que crean compartimentos aislados donde las aplicaciones no pueden acceder a los datos de otras aplicaciones

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[26].

[26] https://www.sentilo.io/wordpress/
https://drive.google.com/file/d/0B6s7sIsOrMfOVTJXOFBRZTU4bjQ/view?resourcekey=0-gSR5dQDX38TQ1ovE27Jq-w
https://sentilo.readthedocs.io/en/latest/index.html
https://sentilo.readthedocs.io/en/latest/architecture.html
(Última fecha de consulta: 01/06/2022)

4.2.14.2. ¿Qué ofrece?

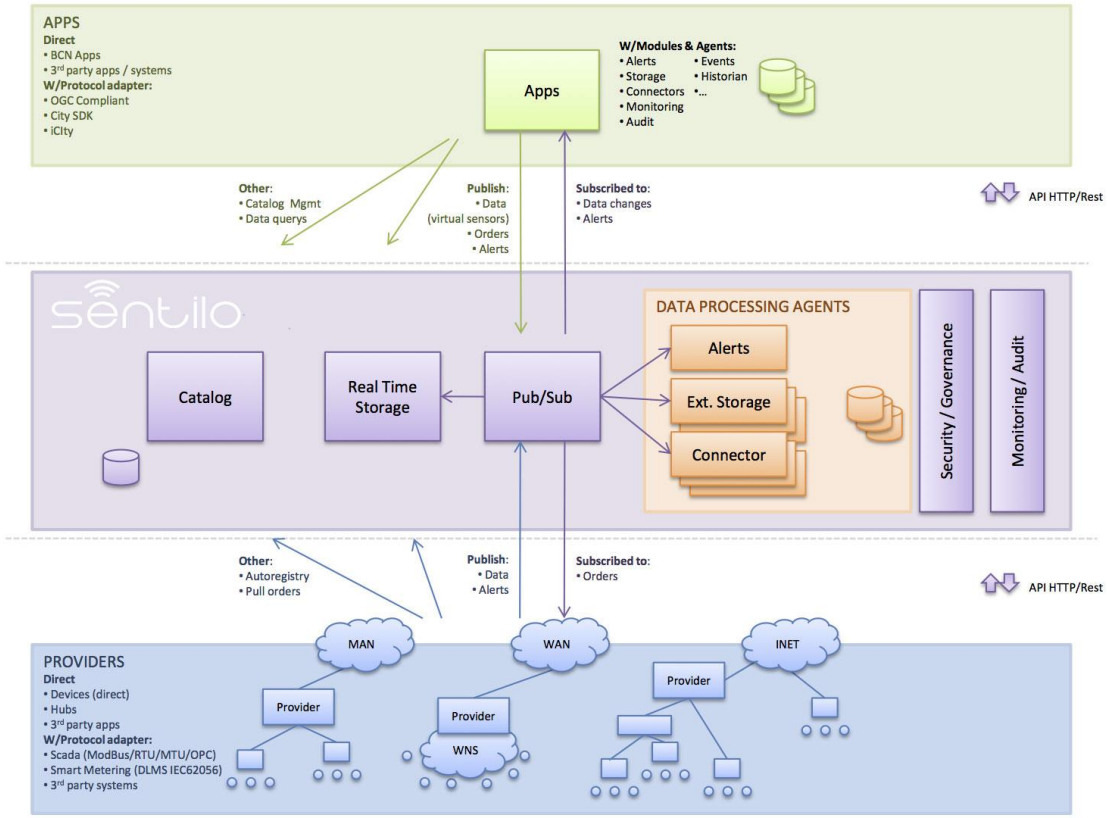

Ilustración 34. Arquitectura de la plataforma Sentilo

En el diagrama representado en la Ilustración 34 se describe la arquitectura de la plataforma Sentilo. En ella podemos observar que se compone de diferentes bloques claramente marcados. Esta arquitectura modular facilita el desarrollo de nuevas funcionalidades y permite agregar funcionalidad sin influir y modificar el resto del sistema. Cabe destacar que esta arquitectura modular permite un crecimiento horizontal ilimitado (De servidores individuales a grandes clústeres).

El primer módulo que se puede observar partiendo desde la izquierda es *Catalog,* un servicio que permite dar de alta o modificar los sensores/componentes de un cliente, o consultar las características de un sensor o proveedor. En cuanto al *Real Time Storage* (Almacenamiento en tiempo real), Sentilo emplea Redis, motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes (clave/valor), donde la plataforma almacena toda la información recibida. Existe la posibilidad de configurarlo para realizar copias de seguridad periódicas en el sistema de archivos. También es el motor de publicación/suscripción *(Pub/Sub)*.

Sentilo permite a los clientes publicar y recuperar información y suscribirse a eventos del sistema gracias al módulo *Pub/Sub Server*. Este módulo es un proceso Java independiente que utiliza Redis como mecanismo de publicación/suscripción. La comunicación del cliente con este

mecanismo se realiza mediante la API REST proporcionada por la plataforma, la cual facilita la conexión de sensores y aplicaciones y ofrece una serie de servicios que se pueden agrupar en:

- *Datos*: Proporciona operaciones para publicar, recuperar, eliminar datos.
- *Order*: Proporciona operaciones para publicar, recuperar, eliminar pedidos.
- *Alarma*: Proporciona operaciones para publicar, recuperar, eliminar alarmas.
- *Subscribe*: proporciona operaciones para suscribir, recuperar y cancelar suscripciones.
- *Catálogo*: proporciona operaciones para insertar, actualizar, consultar y eliminar recursos del catálogo (sensores, componentes y alertas).

A su vez, Sentilo ofrece la posibilidad de ampliar la funcionalidad central de la plataforma a través de un sistema Plug & Play gracias a procesos Java, que, al igual que el módulo Pub/Sub, también utiliza el mecanismo de publicación y suscripción de Redis. A estos procesos Java se les denomina *Agentes*. En Sentilo podemos encontrar los siguientes *Agentes*:

- Agente de base de datos relacional: Se utiliza para exportar datos históricos a una base de datos relacional.
- Agente de alerta: Es utilizado para procesar cada dato recibido por la plataforma y validarlo con las reglas de negocio configuradas en el catálogo.
- Agente de monitor de actividad: Es utilizado para cargar los eventos en Elasticsearch.
- Agente Historiador: Es utilizado para subir los eventos a OpenTSDB.

Para finalizar, en el ámbito de la seguridad, Sentilo ofrece seguridad en relación a las peticiones a la API. En cuanto a la autenticación, Sentilo identifica al solicitante gracias al mecanismo de autenticación basado en tokens. Un token es un identificador único para cada aplicación de proveedor o cliente el cual debe enviarse en cada solicitud como un parámetro de encabezado de la solicitud HTTP denominada IDENTITY_KEY. Para la autorización de una acción sobre un recurso especifico, Sentilo utiliza un sistema de permisos que comprueba que la entidad autorizada (proveedor o aplicación) tiene permiso para administrar, escribir o leer en un recurso.

4.2.14.3. Precios

La plataforma Sentilo se encuentra a disposición del público de manera totalmente gratuita, ya que es un proyecto Open Source (Código abierto) ofrecido por la Diputació de Barcelona27. El gasto que conlleva esta solución depende del hosting escogido para el despliegue de la plataforma, sea en una nube privada o en otras tales como Amazon Web Services, Azure o Google Cloud.

[27] https://www.sentilo.io/wordpress/sentilo-makes-barcelona-the-worlds-smartest-city/

4.2.15. Sofia2

4.2.15.1. ¿Qué es?

Ilustración 35. Logo de la plataforma IoT Sofia2

Sofia2 es una Plataforma IoT & Big Data que surge a partir de un programa europeo de I+D denominado SOFIA (Smart Objects for Intelligent Applications) cuyo propósito es lograr la interoperabilidad entre diferentes aplicaciones. La plataforma cuenta con capacidades como despliegue en la nube, Big Data (para almacenamiento y explotación de la información mediante técnicas analíticas), escalabilidad (tanto para añadir nuevos equipos como capacidad) y multiplataforma (desarrollada para que las aplicaciones que la utilizan puedan crearse en cualquier lenguaje de desarrollo y utilizada en cualquier tipo de dispositivos (ordenadores, *smartphones* (teléfonos inteligentes), tablets, etc...)).

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[28].

4.2.15.2. ¿Qué ofrece?

La plataforma está compuesta por una serie de módulos que se dividen en 2 categorías, aquellos que dan soporte a los sistemas IoT, es decir, los *Módulos IoT* y aquellos que añaden capacidades avanzadas de procesamiento en tiempo real y de analítica Big Data sobre la plataforma, *Módulos Big Data*. En este documento nos centraremos en los Módulos IoT.

[28] https://sofia2.readthedocs.io/en/latest/index.html
https://sofia2.readthedocs.io/en/latest/seguridad.html
https://sofia2.readthedocs.io/en/latest/manuals/basico/seguridad.html
(Última fecha de consulta: 25/06/2022)

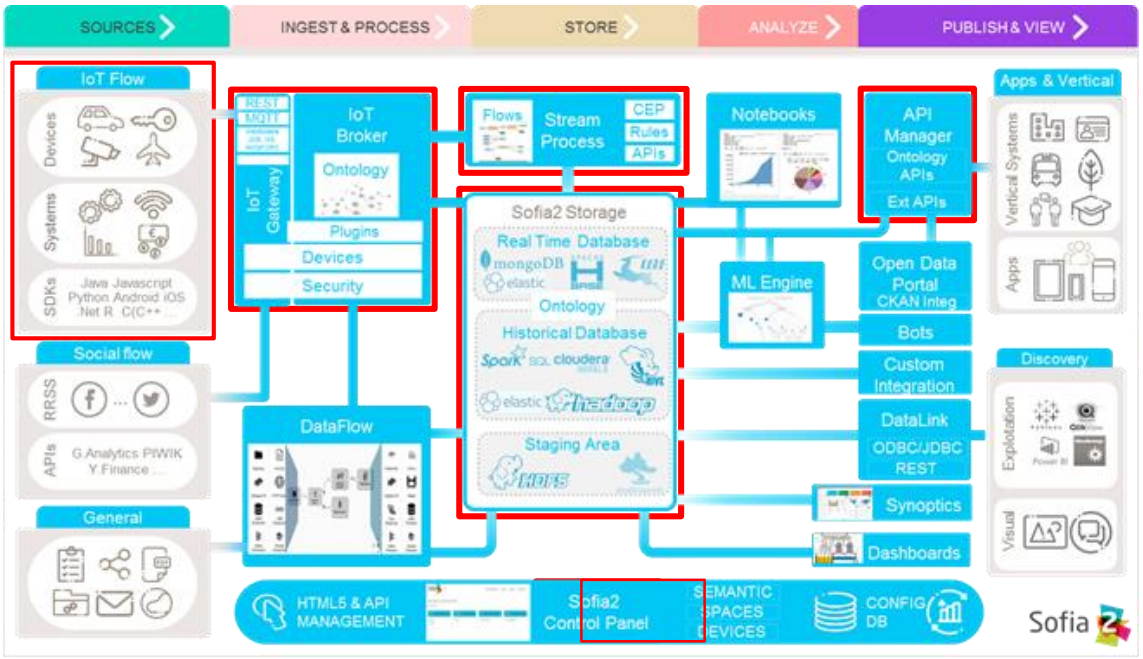

Ilustración 36. Arquitectura de módulos funcionales que componen la plataforma IoT Sofia2

En la Ilustración 36 se han marcado aquellos componentes que forman parte de esta categoría de módulos. Comenzando desde la izquierda, encontramos dentro de la agrupación de IoT Flow, el módulo ***SDKs***, en el que se halla el set herramientas cuya finalidad es facilitar el desarrollo de clientes (emisores y receptores de información) en diferentes lenguajes, como pueden ser: Java, Javascript, Android, IOS, Python, Node.js, Arduino, C, .NET, etc..., y sobre una diversa variedad de protocolos, tal como MQTT, MQTTS, REST, Websockets, WS, etc...

A continuación, se encuentra el módulo ***IoT Gateway***, el cual ofrece una capa de abstracción del protocolo de comunicación, que implementa el protocolo SSAP (Smart Space Access Protocol), sobre diferentes protocolos (MQTT, MQTTS, Websockets, WS, REST) y facilita la incorporación de nuevos protocolos gracias al despliegue de nuevos Plugins. Gracias a esta abstracción, se consigue que la información gestionada por las subsiguientes capas de la plataforma sea completamente agnóstica del protocolo tecnológico usado para el envío del dato. De los protocolos comentados previamente, en esta plataforma, tanto REST como WebSockets se emplean para clientes Javascript, smartphones, etc.., MQTT para comunicaciones bidireccionales y dispositivos básicos y, por último, Web Services, JMS y AMQP para aplicaciones empresariales.

El módulo ***IoT Broker*** es aquel que se encarga de recibir, procesar y almacenar toda la información de las aplicaciones, sensores y dispositivos conectados, actuando como Bus de Interoperabilidad. En esta capa se valida la corrección sintáctica y semántica del dato recibido gracias a la definición previa de la estructura del dato esperado (ontología), identificando de qué dato trata, y aplicando la seguridad correspondiente al mismo. Mediante el despliegue de plugins (mecanismos de extensión de la plataforma) se puede ampliar o adaptar la funcionalidad por defecto de este componente de una manera sencilla. El Motor de Plugins permite dotar de máxima flexibilidad a la Plataforma.

El módulo ***Stream Process*** a su vez se subdivide en dos módulos *Sofia2-Rules* (el Motor de Reglas) y *Sofia-CEP* (el Motor CEP). *Sofia2-Rules* permite ampliar el funcionamiento de Sofia2 permitiendo definir reglas que se ejecutan ante ciertas condiciones (inserción de un nuevo dato o cada cierto tiempo). Estas reglas dan la capacidad de definir, en base a *scripting*, acciones que ejecuta la plataforma, gracias a las cuales, se pueden gestionar y tratar los datos de la misma.

El Motor CEP permite definir reglas en las que interviene el tiempo (por ejemplo, que no ha llegado una cierta medida en 1 día). Los eventos generados por *Sofia-CEP* pueden servir como entrada al motor de Reglas, o ser suscritos por los clientes.

El módulo de almacenamiento de la información de la plataforma (***Sofia2 Storage***) se compone de 3 repositorios:

- *Base de Datos Tiempo Real (BDTR)*: Almacena la información recibida en tiempo real, como instancias de ontologías, siendo, por lo tanto, el primer repositorio en el que se almacena la información recibida de sensores y dispositivos integrados con la plataforma en un contexto IoT típico. Capacidades:
  - Acceso ágil a la información.
  - Herramienta de consulta SQL integrada en el panel de control Sofia2 incluso si la base de datos es NO-SQL.
  - Origen de datos para Analítica de Datos en Tiempo Real.
  - Integración con el motor de Reglas, Machine Learning y capas de integración.
  - Escalabilidad horizontal.
  - Control sintáctico de la información insertada de acuerdo a las ontologías definidas.
- *Base de datos Histórica (BDH)*: Almacena la información histórica para su posterior explotación analítica. Características:
  - Almacenamiento temporal de información heterogénea.
  - Herramienta de consulta SQL integrada en el panel de control Sofia2.
  - Origen de datos para Analítica de Datos Históricos
  - Integración con el motor de Reglas, Machine Learning y capas de integración.
  - Escalabilidad horizontal.
  - Actúa como el corazón del Data Lake de la plataforma, almacenando información heterogénea con capacidad de procesamiento
- *Area de Staging (HDFS)*: Almacena información en diferentes estados (estructurada, semi-estructurada y no estructurada) temporalmente, para facilitar procesos complejos de transformación, ingestión y exposición de datos que requieran la persistencia temporal de estados intermedios del proceso. Características:
  - Almacenamiento temporal de información heterogénea.
  - Usado para dar soporte a procesos analíticos y de transformación de dato complejos.
  - Integración con el motor de Reglas y Machine Learning.
  - Escalabilidad horizontal.

En el apartado de Publish & View (Publicación y vista/acceso a los datos de la plataforma) encontramos el módulo ***Sofia2 API Manager***. Este módulo permite publicar la información gestionada por la plataforma como APIs REST y a su vez permite la búsqueda de estas APIs, la

suscripción por parte de clientes y la gestión del versionado y ciclo de vida de cada una de ellas. Además, este API Manager permite disponibilizar Servicios REST externos a la Plataforma, lo que permite ofrecer un punto único de acceso a APIS internas y externas de la Plataforma. En resumidas cuentas, permite acceder a la información recolectada y gestionada por la plataforma.

Para finalizar con el apartado de módulos, el último de estos es ***Sofia2 Control Panel***, el cual ofrece la funcionalidad de gestionar todos los conceptos que maneja la Plataforma a través de una web de administración/configuración.

Seguridad

Una vez finalizada la descripción de los diferentes módulos y sus funcionalidades, se procederá a informar acerca de cómo la plataforma Sofia2 lidia con la seguridad. La plataforma ofrece diferentes mecanismos de seguridad que permiten garantizar la privacidad de los datos, el control de acceso a los mismos y el envío de información cifrada a través de sus comunicaciones. De entre estos mecanismos, destacan la *comunicación segura y* confidencial en cualquier tipo de conexión, ya sea entre clientes y plataforma o en cualquiera de los puntos de acceso disponibles a través de SSL y/o HTTPS, la *privacidad de los datos*, asegurando el acceso exclusivo de aquellos usuarios que disponen de las credenciales necesarias para el acceso a la información correspondiente, es decir, l*a autorización en el acceso a los datos* permitiendo controlar a grano fino (por ejemplo quien puede insertar qué información y quien puede consultar qué información), la *autenticación* de los clientes de Sofia2 en la comunicación con esta a través de diversos mecanismos, como por ejemplo usuario+ password, token o certificado y la g*estión de usuarios y roles*. Todos estos mecanismos pueden llegar a ser configurados y extendidos a través de los mecanismos de extensión de la plataforma, es decir el Motor de Plugins.

En cuanto a la escalabilidad, Sofia2 garantiza la robustez, la escalabilidad y la alta disponibilidad a diversos niveles:

- Volumen de almacenamiento.
- Procesado de datos.
- Velocidad de proceso de la información.
- Capacidad de respuesta en tiempo real.

Esto se consigue gracias a que la plataforma Sofia2 se construye siguiendo una arquitectura que soporta y garantiza la robustez y escalabilidad horizontal. En adición, relacionado con esta escalabilidad horizontal está el hecho de que la plataforma esté concebida y pensada para correr sobre *hardware commodity* de modo que para aumentar la capacidad de procesamiento baste con incluir máquinas al clúster.

Despliegue

La plataforma oferta tres opciones a la hora de desplegar la plataforma:

- *Cloud Labs o PoC*: La plataforma se despliega en una cloud pública (AWS, Azure, Google Cloud, etc…) y es accesible vía Internet. Esta opción está dirigida a experimentación y pruebas de concepto.
- *On Premise*: La plataforma es desplegada en las instalaciones del cliente (CPD (Centro de Procesamiento de Datos) o Cloud Privada).
- Cloud (SaaS) La plataforma es desplegada en la nube (AWS, Azure, Google Cloud, Flex-IT (Indra), etc…) y puede ser operada por Indra (desarrollador de esta plataforma).

4.2.15.3. Precios

Los planes de precios están ligados al modelo de despliegue por el que se desea optar. En la siguiente Ilustración 37 se puede observar cuales son estos planes y que ofrecen cada uno.

MODELO COMERCIAL

	Cloud Labs o PoC		On Premise	Cloud (SaaS)
Modalidad de Licencia	**Gratis (http://sofia.com)**	**Fijo (por mes)**	**Module x Core (Anual + mantenimiento)**	**Mensajes (miles) & Almacenamiento (TB)**
Soporte	Correo Electrónico & web	8x5 incluido	8x5 incluido 24x7 extendido	8x5 incluido 24x7 extendido
Nuevas versiones, updates, fixes,…	Incluido	Incluido	Incluido	Incluido
SSPP	Elección	Min 50 FTEs	Elección	Elección
Alta Disponibilidad	No HA	No HA	HA	HA
SLA	no SLA	no SLA	No incluido	SLA 99% incluido 99.5% extendido
Entorno Compartido	Si	No	No	No
Infraestructura	Compartida	Dedicada (fija)	Dedicada (configurable)	Dedicada (a demanda)
Funcionalidades	Todo menos: - Acceso directo al sistema de almacenamiento - Capacidades de administración - Documentación premium	Todo	Todo	Todo

Ilustración 37. Precios de Sofia2

Cabe destacar, que si se escoge tanto la opción *Cloud Labs* como *Cloud (SaaS)*, es decir, aquellas que permiten desplegar en nubes públicas tales como Amazon Web Services, Azure o Google Cloud, en esta tabla no se muestra el gasto que conlleva el hosting externo escogido.

4.2.16. Thinger.io

4.2.16.1. ¿Qué es?

Ilustración 38. Logo de Thinger.io IoT Platform

Thinger.io es una plataforma de IoT que proporciona una infraestructura en la nube escalable y lista para usar que mediante las herramientas que ofrece, permite crear prototipos, conectar dispositivos a Internet y realizar cualquier detección o actuación remota a través de Internet de una manera muy sencilla.

Esta plataforma es agnóstica al hardware, por lo que es posible conectar cualquier dispositivo con conectividad a Internet, desde dispositivos Arduino, Raspberry Pi, dispositivos Sigfox, soluciones Lora sobre gateways, o dispositivos ARM, por mencionar algunos.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[29].

[29] https://thinger.io/
https://docs.thinger.io/
https://thinger.io/tag/security/
https://docs.thinger.io/server/deployment
https://pricing.thinger.io/#!/cloud
(Última fecha de consulta: 25/05/2022)

4.2.16.2. ¿Qué ofrece?

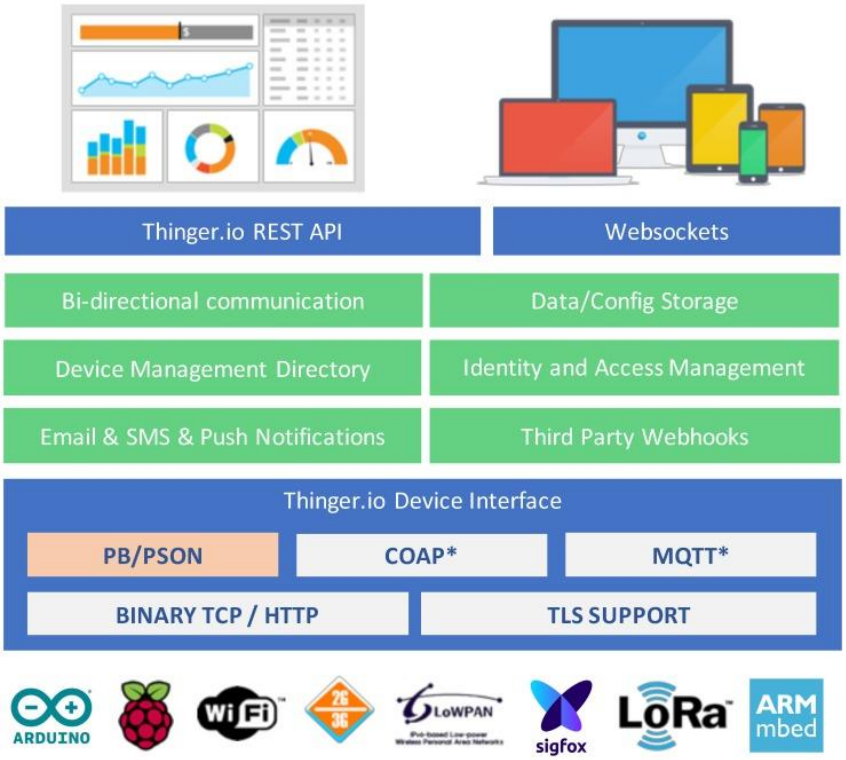

Ilustración 39. Visión general de la plataforma Thinger.io

La visión general de esta plataforma está disponible en la Ilustración 39, como en ella se observa, la plataforma, servidor IoT, proporciona algunas funciones listas para usar, como el registro de dispositivos; la comunicación bidireccional en tiempo real, tanto para la detección como para la actuación; el almacenamiento de datos y configuraciones, de modo que es posible almacenar datos de series temporales; la gestión de identidades y accesos (IAM), para permitir que terceras entidades accedan a la plataforma y a los recursos de los dispositivos a través de las API REST/*Websocket*; los *Webhooks* de terceros, para que los dispositivos puedan llamar fácilmente a otros *Webservices* (servicios web), enviar correos electrónicos, SMS, enviar datos a otras nubes, etc... También proporciona una interfaz web para gestionar todos los recursos y generar *dashboards* (cuadros de mando) para la supervisión remota.

El principal beneficio de utilizar esta plataforma, además del hecho de que es de código abierto, es la posibilidad de obtener una comunicación bidireccional con los dispositivos, en tiempo real, mediante el uso de interfaces REST API estándar. De este modo, es posible desarrollar cualquier aplicación de fusión de datos, ya sea de escritorio, móvil o servicio web, que interactúe con los dispositivos utilizando una interfaz conocida y probada basada en REST.

Para poder llevar a cabo la comunicación bidireccional y por lo tanto acceder a un dispositivo real desde la plataforma, en primer lugar, se ha de crear en la misma una representación de esa entidad, definiendo un modelo, al cual se "enlazara" el dispositivo real. Como se ha comentado previamente, la plataforma Thinger.io está diseñada para soportar casi cualquier

microcontrolador o dispositivo con capacidades de comunicación, a continuación, se mencionarán con mayor profundidad los tipos de dispositivos que son capaces de trabajar en conjunto con la plataforma:

- *Dispositivos compatibles con Arduino* (Arduino MKR WiFi 1010, ESP8266, ESP32, etc.)
- *Dispositivos Linux*, incluyendo la Raspberry Pi, o cualquier otra computadora Linux que ejecute Linux o MacOs.
- *MQTT Device o los clientes de software*, pueden integrarse con el broker MQTT incorporado de Thinger.io.
- *Low-Power Devices or "Edge devices"*: Sigfox, The Things Network o TTN Stack, o cualquier otra infraestructura con gateways.
- Cualquier otro dispositivo o plataforma de terceros que no pueda utilizar el cliente de software de Thinger.io, puede integrarse utilizando su API HTTP REST.

Una de las funciones comentadas previamente fue la del almacenamiento de datos y configuraciones, para conseguir ofrecer esta característica, Thinger.io hace uso de *Data Buckets*. Un *bucket* es un recurso en la nube para almacenar datos de series temporales. Una data *bucket* es un almacenamiento de series temporales donde los dispositivos pueden introducir la información (por ejemplo, la temperatura, la humedad o cualquier otro evento, como detecciones de movimiento, aperturas de puertas de garaje, etc..) cuando sea necesario. Cada conjunto de datos transmitido por un dispositivo se marca automáticamente en la nube en el momento de la recepción, ya que los dispositivos IoT no manejan un *real-time clock* (RTC) (reloj en tiempo real) por sí mismos. Esta información se almacena en la nube en soluciones seguras, eficientes y escalables como por ejemplo MongoDB o DynamoDB de Amazon Web Services. La información almacenada en un *bucket* puede mostrarse en un panel de control dentro de la interfaz de la consola, es decir, en un *dashboard* o exportarse en un almacenamiento escalable (como Amazon S3) para el procesamiento fuera de línea.

Hoy en día es común encontrar que la comunicación entre el dispositivo y el servidor está abierta a ataques (No se manejan conexiones seguras como SSL o TLS). Para solventar este problema, Thinger.io admite conexiones seguras para cualquier dispositivo con la capacidad de comunicarse de manera segura mediante SSL/TLS.

En cuanto al tema del despliegue de esta plataforma, Thinger.io oferta dos modalidades, desplegar un servidor Thinger.io público, es decir, el modelo *Gratuito*, o desplegar el servidor como una instancia privada, modelo *Premiun*, el cual también ofrece una serie de funciones avanzadas. A su vez, el modelo *Premium* oferta dos modelos de hosting: *On-Premise* y *Thinger.io Cloud*.

- *On-Premise*: Las instancias de Thinger.io IoT podrán desplegarse de forma local sobre cualquier tipo de nube o host local, de esta forma el usuario tendrá un control total sobre la infraestructura
- *Thinger.io Cloud*: El despliegue de la instancia privada se realizará sobre la nube accediendo la página web donde se exponen los planes de precios (Ilustración

41). Tras seleccionar uno, simplemente se deberá de seguir el sistema de implementación que ofrece la página.

4.2.16.3. Precios

Los planes *Premium*, *On-Premise* y *Thinger.io Cloud* ofertan diferentes características y capacidades de computación, las cuales se van incrementando/añadiendo conforme aumenta el precio del plan escogido.

	MEDIUM	**LARGE**	**UNLIMITED**
Devices/Assets	Unlimited	Unlimited	Unlimited
Performance	Unlimited	Unlimited	Unlimited
Plugins	3	5	Unlimited
Rebranding	1	5	Unlimited
Custom Domain/TLS	1	5	Unlimited
Developer accounts	5	✓	Unlimited
Guest accounts	Unlimited	Unlimited	Unlimited
Extended support	✓	✓	Available
HA Cluster			Available
Recommended use	Business B2B or B2B2C IoT product	Consultancies with multiple projects	Companies without limits

Ilustración 40. Diferentes licencias ofertadas para el despliegue On-Premise

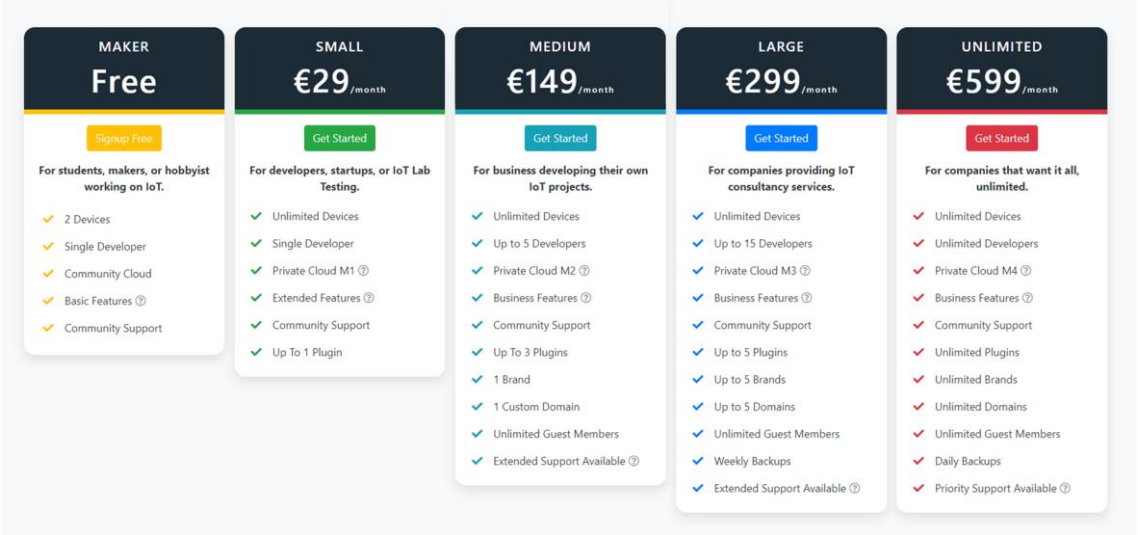

Ilustración 41. Diferentes planes de precios ofertados para el despliegue Thinger.io Cloud

4.2.17. ThingSpeak

4.2.17.1. ¿Qué es?

Ilustración 42. Logo de la plataforma IoT ThingSpeak

ThingSpeak es una plataforma IoT de análisis que permite recolectar, almacenar, visualizar y analizar flujos de datos recolectados por dispositivos IoT en tiempo real en la nube. Tanto el análisis como el procesamiento y la visualización de los datos se puede realizar gracias a la inclusión MATLAB Analytics la cual aporta la capacidad de ejecutar código MATLAB. Toda la documentación consultada para realizar este análisis se encuentra a pie de página[30].

4.2.17.2. ¿Qué ofrece?

El principal aliciente por el que se puede un usuario decantar por esta plataforma es debido a que evita la construcción de un entorno de análisis y visualización de datos, y por la sencillez que ofrece a la hora de realizar la implementación.

Permite que cualquier dispositivo capaz de comunicarse con las APIs tanto REST como MQTT proporcionadas sea capaz de comunicarse con la plataforma. Ejemplos de dispositivos: Arduino, Espressif ESP32 & ESP8266, Raspberry Pi, BeagleBone Black. Un requerimiento añadido del sistema sería el de disponer de la conexión de banda ancha necesaria para el uso interactivo con el sitio web.

En lo referente a cómo funciona la plataforma, ThingSpeak trabaja sobre el concepto de "*Canales*". Un canal es donde se almacenan los datos recibidos de dispositivos y aplicaciones de terceros, los cuales se componen de los siguientes campos:

- Hasta ocho campos ([0-8] campos) para almacenar todo tipo de datos (numérico o alfanumérico).
- Tres campos para almacenar información referente a la latitud, longitud y elevación.
- Un campo denominado estado, el cual se utiliza para describir la información almacenada en el canal.

[30] https://thingspeak.com/pages/learn_more
https://es.mathworks.com/help/thingspeak/
https://thingspeak.com/
https://es.mathworks.com/help/thingspeak/channels-and-charts-api.html?s_tid=CRUX_lftnav
https://thingspeak.com/prices
(Última fecha de consulta: 22/05/2022)

Una vez creado el canal, los dispositivos pueden enviar mensajes con los atributos definidos en el canal. Si el mensaje tiene el formato correcto, la plataforma lo almacenará y para posteriormente, poder analizar (gracias a MATLAB *Analysis*) y presentar de manera gráfica (gracias a MATLAB *Visualizations*) esta información.

En cuanto a la pregunta ¿Es seguro el tráfico de datos?, ThingSpeak soporta TLS, que permite el cifrado de la información transferida entre dos puntos, el dispositivo y ThingSpeak. Una vez los datos estén subidos a ThingSpeak se puede acceder a ellos a través de cualquier navegador (recomiendan Google Chrome, pero también es posible desde Microsoft Edge, Mozilla Firefox y Safari)

4.2.17.3. Precios

ThingSpeak está disponible como servicio gratuito para pequeños proyectos no comerciales (<3 millones de mensajes/año o ~8200 mensajes/día). Para proyectos de mayor envergadura o aplicaciones comerciales, se ofrecen cuatro tipos de licencias anuales diferentes:

- *Gratuito*: Uso relacionado con la educación, que incluye mensajes, canales y tasa de actualización limitados. 0,00 €/año
- *Estudiante*: La opción de licencia para los estudiantes, de bajo coste la cual incluye actualizaciones más rápidas en comparación con la licencia gratuita. 55,00€/año
- *Hogar*: Sólo para uso personal. No para uso gubernamental, académico, comercial o de otro tipo de organización. 75,00€/año
- *Académico*: Licencia para investigación académica y enseñanza. 250,00€/año
- *Estándar*: Uso comercial. 600,00€/año

Tipo de licencia	Gratuito	Estudiante	Hogar	Académico	Estándar
Escalable	No	Sí	Sí	Sí	Sí
Número de mensajes	3M / Año	33 M/año	33 M/año	33 M/año	33 M/año
Límite de tiempo entre mensajes	1 cada 15 segundos	1 cada segundo	1 cada segundo	1 cada segundo	1 cada segundo
Número de canales	4	10	10	250	250
Tiempo máximo de computo	20 s	20 s	20 s	60 s	60 s
Uso de canales privados	Limitado a 3	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Soporte técnico	Community Support	Community Support	Community Support	Standard MathWorks Support	Standard MathWorks Support
Tamaño máximo de la imagen	Función de imagen no disponible	5MB	5MB	5MB	5MB
Mensajes utilizados por imagen	No	100	100	100	100

Tabla I. Tabla comparativa de tipos de licencias de ThingSpeak

4.2.18. Zetta

4.2.18.1. ¿Qué es?

Ilustración 43. Logo de la plataforma API-first para Internet de las cosas (IoT) Zetta

Zetta es una plataforma de IoT de código abierto basada en Node.js para crear servidores de IoT que se ejecutan a través de servidores geo-distribuidos y en la nube. Pese a carecer de una visualización de datos vívida, Zetta combina API REST, WebSockets y *reactive programming* (paradigma enfocado en el trabajo con flujos de datos finitos o infinitos de manera asíncrona, permitiendo que estos datos se propaguen generando cambios en la aplicación, es decir, "reaccionan" a los datos ejecutando una serie de eventos) para ensamblar muchos dispositivos en aplicaciones de datos intensivos y en tiempo real.

Toda la documentación consultada para realizar este análisis se encuentra a pie de página[31].

4.2.18.2. ¿Qué ofrece?

La característica más notable que ofrece Zetta es la de poder ejecutarse en todas partes, es decir, es capaz de ejecutarse la nube, como en un ordenador como incluso en placas de desarrollo modestas (Raspberry Pis, BwangleBones, Arduino y Spark Core, etc…). Al comunicarse con microcontroladores, Zetta puede proporcionar a cada dispositivo una API REST tanto a nivel local como en la nube, lo que en consecuencia deriva en la posibilidad de montar sistemas distribuido de dispositivos que se comunican y reaccionan a través de APIs.

[31] https://www.zettajs.org/
https://github.com/zettajs/zetta/wiki/Overview
(Última fecha de consulta: 10/06/2022)

La Ilustración 44 muestra la arquitectura del servidor Zetta el cual se encuentra en el nivel más alto de abstracción el cual se ejecutará en un concentrador de hardware, es decir un microcontrolador.

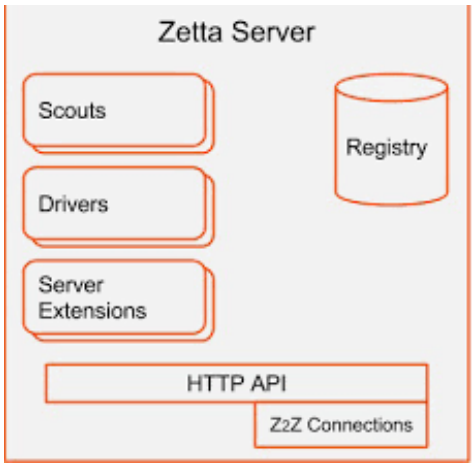

Ilustración 44. Arquitectura del servidor Zetta

Su función es la de coordinar las interacciones entre todos los componentes contenidos en su arquitectura (*Drivers, Scouts, Server Extensions y Registry*) para comunicarse con los dispositivos y generar la API HTTP con las que un consumidor de API puede interactuar.

Los "*Scouts"* (exploradores) sirven como un mecanismo de descubrimiento para dispositivos a través de la red. Buscarán dispositivos en un protocolo particular e informarán a Zetta cuando los encuentren. Los *"Drivers"* son representaciones de máquinas de estado de los dispositivos y son los principales responsables de modelar dispositivos e interactuar con el dispositivo en el nivel físico. Estos modelos de dispositivos se utilizan luego para generar API HTTP y JavaScript para su uso en Zetta. Como su nombre indica, las *"Server Extensions"* (Extensión del servidor), son utilidades que se pueden añadir para ampliar la funcionalidad del servidor, como por ejemplo definición de APIs adicionales o para incrementar la seguridad del mismo. Por último, *"Registry"* es la capa de persistencia de la plataforma. Es una pequeña base de datos que reside en el contexto del servidor y contiene información sobre los dispositivos conectados al propio servidor.

En cuanto al despliegue de Zetta, el esquema del mismo se puede observar en la Ilustración 45. En primer lugar, un servidor Zetta se ejecuta en un microcontrolador, el cual a partir de ese momento actuara como *hub* (nodo). A continuación, el hub se conecta a los dispositivos, y estos se comunican vía HTTP a los protocolos utilizados en el despliegue. Paralelamente, otro servidor Zetta se ejecuta en la nube. Este servidor utiliza exactamente los mismos paquetes Node.js que el servidor que se ejecuta en el hub. Una vez ambos están levantados se conectan. Tras todo esto, Zetta expone una API en el *endpoint* (punto final) de la nube, para que sea consumido por agentes externos.

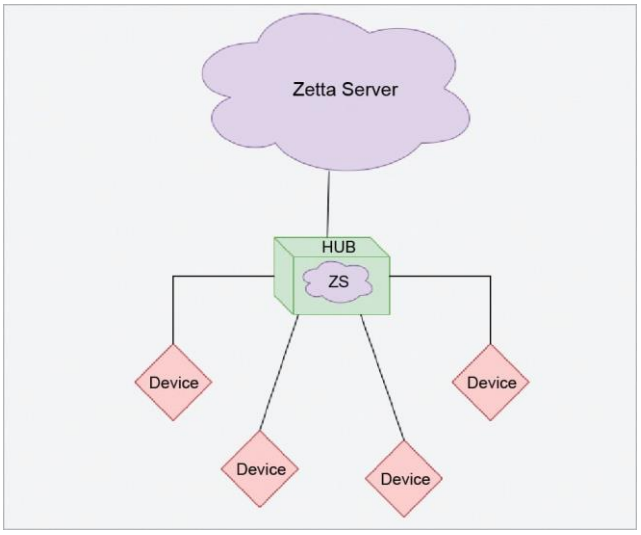

Ilustración 45. Representación del despliegue de Zetta

4.2.18.3. Precios

IoT Zetta es una plataforma gratuita de código abierto, que cuenta con complementos opcionales de pago, como son la asistencia o la formación.

5. Evaluación de plataformas IoT

En el panorama tecnológico actual, existe una amplia variedad de plataformas de Internet de las Cosas (IoT), cada una de las cuales ofrece diversas capacidades y características. Esta abundancia hace que el proceso de selección de una solución adecuada sea una tarea difícil, en la que comprender los atributos específicos de estas plataformas IoT se convierte en una necesidad fundamental.

En este punto plantearemos una serie de características técnicas básicas y avanzadas que pueden ofrecer las plataformas IoT explicando el papel crucial que juegan dentro del ecosistema IoT, para posteriormente, en un conjunto de tablas, se mostrarán de manera resumida estas características.

Durante la realización del análisis previo pudimos observar en las fuentes de documentación un patrón a la hora de describirlas, basado en puntos muy característicos como pueden ser:

- SDK & Lenguajes Soportados para Desarrollar
- Protocolos de Comunicación Compatibles
- Almacenamiento de Datos
- Escalabilidad
- Seguridad
- Facilidad de Uso
- Precio

A su vez, para añadir valor al estudio, hemos leído numerosos artículos para tratar de encontrar otros requisitos/características/funciones que pudiésemos haber pasado por alto. De entre todos ellos queremos destacar los artículos de investigación [1] y [2] los cuales tratan de comparar de manera exhaustiva plataformas en la nube. De ellos hemos obtenido numerosas funciones que acabaron formando parte de las tablas comparativas siendo estas las siguientes:

- Certificaciones y cumplimiento de la normativa
- Análisis de Datos
- Gestión de Dispositivos
- Desarrollo de Aplicaciones
- Hardware Compatible
- Formatos de Serialización Compatibles
- ¿La plataforma permite enviar comandos a los dispositivos?
- Gestión de Usuarios y Roles
- Requisitos No Funcionales
- Dominios o Casos de Uso Respaldados

5.1. Análisis de características clave

La gran cantidad de plataformas de Internet de las Cosas (IoT) disponibles hoy en día presenta una tarea de evaluación compleja para la selección de una solución apropiada. En este contexto, la comprensión de características específicas de las plataformas IoT es esencial ya que estas características juntas desempeñan un papel crucial en la configuración del rendimiento y la funcionalidad de la plataforma. Por tanto, una comprensión y evaluación exhaustivas de estas características conducirán a una elección más informada y eficaz de la plataforma IoT.

Como se ha comentado previamente, se han escogido un total de 17 características que servirán de base para la posterior comparación de las plataformas con el fin de escoger cual es la más apropiada para unos determinados casos de uso. A continuación, se enumerarán las susodichas características y expondrán los argumentos en los que se basa la selección de cada una de ellas.

1) Certificaciones y cumplimiento de la normativa

Esto se refiere a las normas y regulaciones que cada plataforma cumple, y a las certificaciones que ha obtenido para demostrar su cumplimiento. En el mundo del IoT, el cumplimiento de las normas de interoperabilidad y seguridad, entre otras, garantiza una colaboración fluida entre distintos dispositivos y sistemas, al tiempo que reduce los riesgos normativos.

2) SDK & Lenguajes Soportados para Desarrollar

La flexibilidad de utilizar diferentes lenguajes de programación y disponer de un sólido kit de desarrollo de software (SDK) influyen significativamente en la gama de dispositivos y aplicaciones que se pueden crear y utilizar. Un conjunto diverso de lenguajes permite flexibilidad en el desarrollo, adaptándose a las distintas preferencias de los desarrolladores y a los requisitos de los dispositivos, fomentando así la diversidad y la inclusión en las soluciones IoT.

3) Protocolos de Comunicación Compatibles

Los protocolos de comunicación dictan cómo se transmiten los datos entre la infinidad de dispositivos IoT y la nube. Una plataforma que admita múltiples protocolos garantiza la compatibilidad con una amplia gama de dispositivos y aplicaciones. Protocolos como MQTT, CoAP y XMPP se utilizan habitualmente en IoT debido a su eficiencia dispositivos con pocas capacidades y adecuación para distintos casos de uso en función de factores como la eficiencia energética, el ancho de banda y las necesidades de comunicación en tiempo real.

4) Almacenamiento de Datos

Los dispositivos IoT generan grandes cantidades de datos, tanto para el análisis en tiempo real como para el post-proceso, lo que requiere soluciones de almacenamiento robustas y eficientes. La capacidad de almacenamiento de datos de la plataforma, ya sea en su propia infraestructura o en centros de datos externos, junto con su enfoque de la localización de datos, influye significativamente en la gestión de grandes cantidades de datos (ya que puede adaptarse a las necesidades de procesamiento inmediato y/o al análisis histórico a largo plazo), el rendimiento y el coste de una solución IoT.

5) Escalabilidad

Dado que las soluciones de IoT a menudo suelen tener como objetivo los despliegues extensivos, la capacidad de la plataforma para gestionar el crecimiento del número de dispositivos o usuarios a lo largo del tiempo se convierte en una característica crucial. Esta escalabilidad, incluyendo la escalabilidad horizontal (soporte a más dispositivos) y vertical (manejo de más tareas), es esencial para mantener el rendimiento del sistema en medio de aumentos rápidos y sustanciales en los dispositivos y usuarios conectados, típicos de los entornos de IoT.

6) Seguridad

La naturaleza sensible de los datos en las redes IoT requiere una plataforma que cuente con sólidas medidas de seguridad para garantizar la privacidad, integridad y disponibilidad de los datos. Estas medidas deben impedir el acceso no autorizado e incluir seguridad en la capa de transporte, seguridad a nivel de mensaje y cifrado de datos tanto en tránsito como en reposo. Otras características son la autenticación y autorización de usuarios (que se tratará más adelante como un punto independiente, 15) Requisitos No Funcionales, ya que consideramos que es una funcionalidad que puede llegar a ser muy relevante), la conectividad segura de dispositivos y las actualizaciones sin demora de seguridad.

7) Facilidad de Uso

La facilidad de uso de una plataforma de IoT, un factor primordial para el desarrollo y la implantación rápidos se refiere a la facilidad con la que los desarrolladores y los usuarios interactúan con la plataforma. Esto incluye la intuitividad, facilidad de manejo, sencillez de su interfaz de usuario, la calidad de sus herramientas y la riqueza y calidad de su documentación. Una plataforma fácil de usar promueve una adopción más rápida, mejora la productividad de los desarrolladores, reduce el tiempo de comercialización y minimiza la curva de aprendizaje para los desarrolladores. Además, la disponibilidad de servicios de asistencia contribuye a su usabilidad general.

8) Precio

Las implicaciones financieras del despliegue de una plataforma IoT son significativas y van más allá del precio de compra o de la cuota de suscripción. El gasto global incluye costes asociados como el almacenamiento de datos, la comunicación, la conectividad de dispositivos y los servicios adicionales. Por lo tanto, es fundamental comprender el modelo integral de fijación de precios y evaluar la relación coste-beneficio en relación con el valor aportado por la plataforma.

9) Análisis de Datos

Las soluciones IoT, al estar centradas en los datos, generan una gran cantidad de datos que necesitan un procesamiento y análisis competentes para extraer información valiosa. Por lo tanto, las capacidades de análisis de datos integrales de una plataforma IoT son fundamentales. Esto no sólo incluye el análisis en tiempo real para la toma inmediata de decisiones y el procesamiento por lotes para el análisis retrospectivo, sino también herramientas avanzadas como el aprendizaje automático, la inteligencia artificial (IA) y la visualización de datos. El aprendizaje automático y la IA pueden ayudar a identificar patrones, predecir tendencias y tomar decisiones basadas en grandes cantidades de datos, mientras que la visualización de datos simplifica la comprensión de patrones de datos complejos presentándolos en un formato más intuitivo y comprensible. En conjunto, estas capacidades transforman los datos brutos en información práctica, mejorando el uso eficaz de los datos de IoT.

10) Gestión de Dispositivos

En un ecosistema IoT, las funciones de gestión de dispositivos son fundamentales y abarcan aspectos como el aprovisionamiento de dispositivos, la configuración, las actualizaciones de firmware, la supervisión, el diagnóstico y la resolución de problemas. Estas capacidades son esenciales para mantener la salud y el rendimiento de los dispositivos, permitiendo controlar y conocer los dispositivos IoT de la red.

11) Soporte para Desarrollo de Aplicaciones

El soporte de una plataforma IoT para el desarrollo de aplicaciones facilita significativamente el proceso de creación de aplicaciones que interactúen con dispositivos IoT, realicen tareas y se integren con otras soluciones backend y frontend, ahorrando así esfuerzos de desarrollo. Este soporte suele incluir la provisión de las API para acceder a las funciones de la plataforma, los SDK para facilitar la integración, las herramientas de depuración y la compatibilidad con populares entornos de desarrollo, así como herramientas para el desarrollo de interfaces de usuario.

12) Hardware Compatible

La capacidad de una plataforma IoT para admitir una amplia gama de plataformas de hardware, incluidos diversos dispositivos y sensores IoT, garantiza la interoperabilidad, permite la diversidad en el uso de dispositivos y posibilita configuraciones de sistemas más versátiles

dentro del ecosistema IoT. Este amplio soporte de hardware, que debe dar cabida a diversas opciones de conectividad y protocolos de dispositivos, determina la variedad de dispositivos utilizables dentro del sistema IoT y permite ampliar los ámbitos de aplicación.

13) Formatos de Serialización Compatibles

Los formatos de serialización como JSON, XML o formatos avanzados como MessagePack y Protobuf desempeñan un papel significativo en la representación, codificación e intercambio de datos a través de la red. Dada la limitación de recursos de muchos dispositivos IoT, la plataforma debe admitir formatos de serialización de datos eficientes que empaqueten los datos de forma compacta para su transmisión, lo que influye en el volumen de tráfico de red, la potencia de procesamiento necesaria y la compatibilidad con diversos dispositivos.

14) ¿La plataforma permite enviar comandos a los dispositivos?

Esta característica responde a la necesidad de comunicación bidireccional en las redes IoT, que permite la supervisión, el control, la interacción y la configuración de dispositivos en tiempo real. La capacidad de una plataforma IoT para enviar comandos a los dispositivos no sólo permite una interacción dinámica con el sistema IoT, sino que también permite respuestas dinámicas a condiciones cambiantes, desempeñando un papel clave en la gestión eficaz de un sistema IoT.

15) Gestión de Usuarios y Roles

Las funciones de gestión de usuarios y roles son cruciales en una plataforma IoT para mantener la seguridad, garantizar la integridad del sistema y permitir un control preciso. Entre ellas se incluyen la *multi-tenancy* (multi-inquilinos) (significaría que la plataforma puede admitir varios usuarios o grupos de usuarios (inquilinos (*tenants*)), cada uno con su propio conjunto de reglas, configuraciones y datos, al tiempo que comparte los mismos recursos del sistema, como memoria, procesador y almacenamiento), la autenticación de usuarios, la creación y gestión de diferentes roles de usuario con privilegios de acceso específicos y un nivel granular de control de acceso. Tales mecanismos permiten una distribución variada de los permisos, gestionan las cuentas de usuario y proporcionan un control de acceso seguro, garantizando que cada usuario sólo pueda acceder y controlar los dispositivos y datos de su competencia, mejorando así la privacidad.

16) Requisitos No Funcionales

Estos requisitos son fundamentales para garantizar el funcionamiento coherente y fiable del sistema IoT. Son atributos de calidad que describen cómo debe funcionar el sistema, más que sus funcionalidades, e incluyen aspectos como el tiempo de actividad de la plataforma (disponibilidad (*availability*)), la velocidad a la que procesa las solicitudes (rendimiento (*performance*)) y cómo se protege frente a la pérdida de datos (capacidades de copia de seguridad (*data backup*) y

recuperación de datos (*restoration capabilities*)). Estos atributos determinan la fiabilidad y eficacia con que la plataforma puede dar soporte a un sistema IoT en implantaciones reales.

17) Dominios o Casos de Uso Respaldados

Identificar los dominios o casos de uso que admite una plataforma es crucial para determinar su idoneidad para la aplicación prevista. Algunas plataformas pueden estar adaptadas a áreas específicas, como los hogares inteligentes o el IoT industrial, mientras que otras pueden ser más versátiles y admitir una gran variedad de casos de uso, desde dispositivos portátiles hasta ciudades inteligentes. Esta amplitud de aplicabilidad influye en la facilidad de implantación (ya que una plataforma más versátil que admita una amplia gama de casos de uso tendrá probablemente herramientas y características más genéricas, lo que la más hará adaptable a una gran variedad de proyectos), la flexibilidad de la plataforma y su eficacia en escenarios específicos. Cuantos más dominios admita una plataforma, más adaptable será a las distintas aplicaciones de IoT.

En resumen, cada una de estas características es crítica por sí misma, y en conjunto dan forma a la funcionalidad y eficacia generales de una plataforma IoT. Es fundamental tener en cuenta estas características en el contexto de los requisitos específicos del proyecto, ya que una comprensión exhaustiva de las mismas forma parte integral del proceso de selección. La elección correcta puede influir significativamente en el éxito de la implantación y la escalabilidad de los sistemas IoT.

5.2. Catálogo de características ofrecidas por las plataformas IoT

Tras una exploración meticulosa de diecisiete características que perfilan las competencias y la valía intrínseca de las plataformas IoT, la siguiente sección de este trabajo presenta una serie de tablas. Dichas tablas representan un análisis comparativo minucioso, ilustrando cómo las diversas plataformas IoT abordan cada una de estas características. Se ha escogido este enfoque con el fin de proporcionar una perspectiva holística y multidimensional del actual ecosistema de las plataformas IoT.

El objetivo de este análisis crítico es asistir a los lectores en la compleja tarea de seleccionar la plataforma que mejor se adapte a sus necesidades específicas, otorgándoles una herramienta de evaluación pragmática y valiosa. Este desglose en profundidad tiene la capacidad de revelar el impacto real y el valor de estas plataformas en el marco del IoT.

5.2.1. Los pilares para la selección de una plataforma en la nube

La primera tabla (Tabla II) señala a los proveedores de las diferentes plataformas, y las dos características que, en mi opinión, pueden decantar la decisión de no escoger una de ellas. Estas dos serían, en primer lugar, si el servicio ofrecido por la plataforma (generalmente de tipo PaaS) es de pago, y, en segundo lugar, si está posee algún tipo de certificación que asegure que cumple con estándares, normas y/o regulaciones de seguridad de la industria.

Plataforma	Proveedor	Precio	Certificaciones y cumplimiento de la normativa:
Altair SmartWorks IoT	Altair Engineering	L	ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, IEC 62443-3-1:2020, IEC 62366, SOC 2, PCI DSS, HIPAA.
AWS IoT Core	Amazon	L	ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, SOC 2, PCI DSS, HIPAA, GDPR, FedRAMP, FIPS 140-2 Level 3. Reconocido por CSA y NIST.
Carriots	Altair Engineering	G	Certificado por la EU y AWS. Miembro de la IoT Alliance, Cumple con GDPR.
DeviceHive	DataArt	G	Bajo licencia Apache 2.0.
Google Cloud IoT Core	Google	L	ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, SOC 2, PCI DSS, HIPAA, GDPR, FedRAMP. Reconocido por CSA.
IBM Watson IoT	IBM	L	ISO/IEC 27001, SOC 2, PCI DSS, HIPAA, GDPR, FedRAMP. Miembro de IoTSF, IIC, World Economic Forum.
IoTSens	IoTSens	N	Certificado UNE 178104. Reconocido por el gobierno español como plataforma referente en el ámbito del IoT.
Kaa	KaaIoT Technologies	L	ISO/IEC 27001, SOC 2, PCI DSS. Miembro certificado de IoTSF. Cumple con el NIST Cybersecurity Framework.
Macchina.IO EDGE	Applied Informatics	D	ISO/IEC 27001, PCI DSS, HIPAA.
Mainflux	Mainflux Labs	D	ISO/IEC 27001, SOC 2, GDPR, HIPAA. Reconocido por la EU. Miembro de IIC & OIC.
Microsoft Azure IoT Central	Microsoft	L	Azure IoT Central, cumple las políticas y normativas de Microsoft, así como una serie de normas del sector, entre las que se incluyen ISO/IEC 27001, SOC 2, PCI DSS, HIPAA, GDPR, FedRAMP, FIPS 140-2. Certificado por CSA, NIST, IoTSF, Gartner, Forrester, International Data Corporation (IDC), Common Criteria for Information Technology Security Evaluation (CC).
Microsoft Azure IoT Hub	Microsoft	L	ISO/IEC 27001, HIPAA, FedRAMP, SOC 2, PCI DSS y FIPS 140-2. Está reconocido por organizaciones como: Gartner, Forrester, IDC, IoTSF, IIC, NIST, CSA
OpenRemote	OpenRemote BV	D	ISO/IEC 27001, GDPR. Reconocido y certificado por Linux Foundation como uno de los 10 mejores proyectos Open Source y por la Open Source Initiative (OSI) como proyecto totalmente de código abierto.
Predix Platform	GE (General Electric)	P	ISO/IEC 27001, IEC 62443, HIPAA, SOC 2, PCI DSS. Reconocido por IIC, OPC, CSA, TUV SUD y British Standards Institution (BSI).
Sentilo	la Ciudad de Barcelona.	G	OIC, AllSeen Alliance (AllJoyn), OASIS Data Model for Sensor Networks (DMSN) & OASIS Sensor Markup Language (SenML). Bajo licencia Apache 2.0
Sofia2	Indra Company	L	ISO/IEC 27001, SOC 2, PCI DSS
Thinger.io	Thinger.io	L	Certificado por AWS. Cumple con GDPR. Reconocido por IoTSF & CSA
ThingSpeak	MATLAB & Simulink (MathWorks)	L	Certificado por Internet of Things Consortium, LoRa Alliance y CSA. Cumple con GDPR. Reconocido como un líder en el Mercado Global de Plataformas IIoT (Global IIoT Platform Market) por Frost & Sullivan.
Zetta	Apigee (Google Cloud)	G	Cumple con IEEE 802.15.4, Zigbee, Z-Wave, OAuth 2.0, OpenID Connect y Transport Layer Security 1.2.

Tabla II. Lista de Plataformas IoT junto con sus proveedores, modelos de negocio y certificaciones

G	Gratuito para uso personal y comercial	L	Limitado. Ofrece un nivel básico gratuito con ciertas limitaciones, las funcionalidades completas son de pago
D	Depende. Gratuito para uso personal, de pago para uso comercial	P	De pago
N	No ofrece información sobre planes de precios		

Tabla III. Leyenda: Significado de los modelos de negocio de la Tabla II.

En la Tabla II, se aprecia una repetición de ciertos organismos certificadores y normativas que definen las prácticas óptimas para la administración de la seguridad de la información. A continuación, proporcionaré el significado de los términos abreviados que no han sido aclarados en la Tabla II y haré un énfasis particular en aquellos organismos de certificación y normativas que se presentan con mayor frecuencia, explicándolos brevemente.

Abreviatura	**Significado**
CSA	*Cloud Security Alliance*
EU	*European Union*
FedRAMP	*Federal Risk and Authorization Management Program*
FIPS	*Federal Information Processing Standards*
GDPR	*General Data Protection Regulation*
HIPAA	*Health Insurance Portability and Accountability Act*
IEEE	*Institute of Electrical and Electronics Engineers*
IIC	*Industrial Internet Consortium*
IoTSF	*IoT Security Foundation*
ISO/IEC	*International Organization for Standardization / International Electrotechnical Commission*
NIST	*National Institute of Standards and Technology.*
OIC	*Open Interconnect Consortium*
PCI DSS	*Payment Card Industry Data Security Standard*
SOC	*Service Organization Controls*

Tabla IV. Índice de las abreviaturas presentes en la Tabla II

ISO/IEC 27001: Es una norma internacional para la gestión de la seguridad de la información. Proporciona un marco para establecer, implantar, operar, supervisar, revisar, mantener y mejorar un Sistema de Gestión de la Seguridad de la Información (SGSI) (ISMS, por sus siglas en inglés). El SGSI es un enfoque sistemático para gestionar la información sensible de la empresa con el fin de garantizar su seguridad. Esta norma no impone controles de seguridad específicos, pero proporciona una lista de objetivos de control y sugiere una serie de controles alternativos que las organizaciones podrían considerar utilizar.

SOC 2: Los Controles de Organizaciones de Servicios (*Service Organization Controls* (SOC)) son un marco de auditoría definido por el Instituto Americano de CPA (AICPA). El tipo de

certificación 2 (SOC 2) la otorgan auditores externos, que evalúan el grado de cumplimiento por parte de una empresa de uno o varios de los cinco principios de confianza (seguridad, disponibilidad, integridad del procesamiento, confidencialidad y privacidad) basándose en los sistemas y los procesos que la organización utiliza para procesar los datos de los usuarios. Mientras que la SOC 2 se centra más en el cumplimiento de los cinco principios de confianza, SOC 1 se centra en informar sobre los informes financieros. También existe el tipo 3 de esta certificación, el cual informar sobre los mismos controles que SOC 2, pero de una forma que resulte comprensible para el público en general

PCI DSS: El Estándar de Seguridad de los Datos de la Industria de Tarjetas de Crédito (*Payment Card Industry Data Security Standard* (PCI DSS)) es una normativa patentada que impone controles estrictos a las organizaciones que procesan, almacenan o transmiten información de tarjetas de crédito. Establecida por el Consejo de Estándares de Seguridad de la Industria de Tarjetas de Pago, la PCI DSS tiene como objetivo mitigar el fraude con tarjetas de crédito y mejorar la seguridad de los datos de los titulares de tarjetas en todo el mundo.

HIPAA: La Ley de Portabilidad y Responsabilidad de los Seguros Médicos (*Health Insurance Portability and Accountability Act* (HIPAA)) es una ley federal de Estados Unidos promulgada para salvaguardar la información sanitaria sensible de los pacientes y evitar que se divulgue sin su consentimiento o conocimiento. Las organizaciones que manejan información sanitaria protegida (PHI) deben garantizar el cumplimiento riguroso de las medidas de seguridad físicas, de red y de proceso.

GDPR: El Reglamento General de Protección de Datos (*General Data Protection Regulation* (GDPR)) es una normativa de la Unión Europea (UE) diseñada para proteger la privacidad y los datos personales de los ciudadanos de la UE. Impone normas estrictas a las entidades que alojan y procesan datos personales, garantizando los derechos fundamentales de protección de datos, independientemente de la ubicación geográfica de los datos.

FedRAMP: El Programa Federal de Gestión de Riesgos y Autorizaciones (*Federal Risk and Authorization Management Program* (FedRAMP)) es un programa gubernamental estadounidense que proporciona una metodología estandarizada para evaluar, autorizar y supervisar continuamente la seguridad de los productos y servicios en la nube. Su objetivo es promover la adopción de servicios seguros en la nube en todo el territorio estadounidense, proporcionando un enfoque estandarizado para la seguridad y la evaluación de riesgos

De momento, hemos profundizado en el papel fundamental que desempeñan las certificaciones y las normas de conformidad a la hora de optar por una plataforma de Internet de las Cosas (IoT). Estos indicadores, reconocidos a nivel internacional, actúan como poderosos indicadores del compromiso de una plataforma IoT con protocolos de seguridad estrictos y la protección de los datos de los usuarios.

Al navegar por el complejo ámbito de IoT, estas normas reguladoras constituyen una valiosa referencia que influye significativamente en los clientes potenciales en su proceso de elección de plataformas de IoT. Lo hacen estableciendo un punto de partida con respecto al cual pueden evaluarse las capacidades de seguridad y protección de datos de las plataformas IoT.

En consecuencia, no se puede exagerar su importancia como factores determinantes en la selección de una plataforma de IoT.

5.2.2. Perspectiva holística y multidimensional del actual ecosistema de plataformas

En este subpunto se presentan las tablas que desempeñan un papel protagonista en el análisis comparativo las diversas plataformas IoT mostrando cómo estas abordan cada una de las ya mencionadas características críticas.

Cabe destacar que esta comparación se fundamenta principalmente en información de dominio público perteneciente a estas plataformas. La fuente de datos primordial para este estudio comprende los sitios web de los proveedores y las documentaciones oficiales de las plataformas bajo evaluación. Complementariamente, se ha indagado también en artículos técnicos y empresariales, así como a informes de investigación para la recopilación de datos. Cualquier posible error en los datos de alguna plataforma, si lo hubiera, es involuntario y se lamenta.

Se han clasificado las características en 6 aspectos cruciales que debe poseer una plataforma IoT en la nube, en función de cómo estás características se ajustan a dichos conceptos. Será en el punto siguiente **6. Estudio de las plataformas IoT para dominios específicos** donde se explicará la relevancia de estos aspectos en la selección de una plataforma en el ámbito del IIoT (*Industrial IoT*).

Sin más preámbulos, pasamos a mostrar el desglose de características de estas plataformas en el marco del IoT.

1. Gestión de datos:

1.1. Integración e Ingesta de datos:

Plataforma	Formatos de Serialización Compatibles										
	JSON	XML	CSV	Protobuf	Binario	YAML	Avro	CBOR	MessagePack	Thrift	Otros formatos
Altair SmartWorks IoT	✓	✓	✓	✓		✓	✓				INI, formato de serialización Personalizado
AWS IoT Core	✓	✓		✓	✓		✓	✓	✓	✓	Personalizado, Cap'n Proto
Carriots	✓	✓	✓	✓	✓	✓	✓	✓		✓	Personalizado, Cap'n Proto
DeviceHive	✓	✓	✓	✓	✓		✓	✓	✓	✓	Personalizado, Cap'n Proto
Google Cloud IoT Core	✓	✓		✓	✓	✓	✓	✓	✓	✓	Cloud Pub/Sub messages, MQTT, TLV8, JWT, Personalizado
IBM Watson IoT	✓	✓	✓	✓	✓		✓	✓	✓	✓	Personalizado, Cap'n Proto
IoTSens	✓	✓	✓	✓		✓	✓	✓	✓		Cap'n Proto, FlatBuffers
Kaa	✓	✓		✓		✓	✓	✓	✓	✓	Personalizado
Macchina.IO EDGE	✓	✓	✓	✓	✓		✓	✓			
Mainflux	✓	✓	✓	✓		✓	✓	✓	✓	✓	Pickle
Microsoft Azure IoT Central & Hub	✓	✓	✓	✓	✓	✓	✓		✓	✓	Personalizado, Parquet
OpenRemote	✓	✓	✓	✓		✓	✓	✓	✓	✓	Personalizado, Cap'n Proto
Predix Platform	✓	✓	✓	✓	✓		✓	✓	✓	✓	Flatbuffers, Cap'n Proto, UBJSON
Sentilo	✓	✓	✓	✓	✓	✓	✓		✓	✓	
Sofia2	✓	✓	✓	✓	✓	✓	✓		✓	✓	Flatbuffers, Cap'n Proto, ASN.1
Thinger.io	✓	✓	✓	✓							
ThingSpeak	✓	✓	✓								
Zetta	✓	✓	✓	✓	✓	✓	✓		✓		

Característica I. Formatos de Serialización Compatibles

Análisis de los factores clave que se extraen al observar la tabla:

Compatibilidad de la Plataforma con los Formatos: Algunas plataformas son compatibles con una amplia gama de formatos, mientras que otras tienen un conjunto más limitado de formatos compatibles.

- Amplia compatibilidad: Las plataformas AWS IoT Core, Carriots, DeviceHive, Google Cloud IoT Core, IBM Watson IoT, Mainflux, Microsoft Azure IoT Central & Hub, OpenRemote, y Sofia2 son compatibles con la mayoría de los formatos de serialización mencionados en la tabla.

- Compatibilidad Limitada: Las plataformas Thinger.io, ThingSpeak son compatibles solo con un subconjunto limitado de los formatos de serialización mencionados.

Formatos más comunes: JSON y XML parecen ser los formatos más comunes y son soportados por todas las plataformas listadas.

Formatos Específicos: Además de los formatos de serialización estándar que se mencionan en la tabla, varias plataformas también soportan formatos de serialización adicionales o específicos, que se enumeran en la columna "Otros formatos". Algunos ejemplos de estos son:

- Formato personalizado: Algunas plataformas como AWS IoT Core, Altair SmartWorks IoT, Carriots, DeviceHive, IBM Watson IoT, Kaa, Microsoft Azure IoT Central & Hub y OpenRemote ofrecen la opción de personalizar formatos de serialización. Esto puede ser útil si tienes requerimientos muy específicos que no pueden ser cubiertos por los formatos estándar.

- Cap'n Proto: Este es un formato de serialización binaria de alto rendimiento. Algunas plataformas que lo soportan son AWS IoT Core, Carriots, DeviceHive, IBM Watson IoT, IoTSens, OpenRemote, Predix Platform y Sofia2.

- TLV8, JWT: Son soportados por Google Cloud IoT Core. TLV8 es otro formato de serialización, y JWT (JSON Web Token) es una manera compacta y segura de transmitir información entre partes como un objeto JSON.

- Pickle: Este es un formato específico soportado por Mainflux. Se trata de un módulo de Python que implementa protocolos binarios para serializar y deserializar una estructura de objetos Python.

- Parquet: Microsoft Azure IoT Central & Hub soporta este formato de almacenamiento de columnas de código abierto optimizado para su uso con sistemas de procesamiento de consultas analíticas.

- FlatBuffers, UBJSON, ASN.1: Son soportados por varias plataformas. FlatBuffers es un sistema de serialización eficiente similar a Cap'n Proto y Protocol Buffers. UBJSON (*Universal Binary JSON*) es un formato binario que intenta superar las limitaciones de JSON y BSON. ASN.1 (*Abstract Syntax Notation One*) es una norma que define una forma de representar, codificar, transmitir y decodificar datos.

Algunas plataformas pueden ser preferidas sobre otras debido a su soporte para formatos de serialización de alto rendimiento, específicos del dominio, personalizados, o inusuales.

Plataforma	Protocolos de Comunicación Compatibles													
	MQTT	AMQP	REST	HTTP	HTTPS	CoAP	MQTT/AMQP/HTTP sobre WebSockets	LoRaWAN	WebSockets	ModBus	OPC UA	XMPP	LwM2M	Otros Protocolos
Altair SmartWorks IoT	✓	✓	✓	✓	✓	✓					✓	✓		DDS
Amazon Web Service IoT Core	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	AWS IoT Device SDKs, SNMP
Carriots	✓	✓	✓	✓		✓						✓		cURL, hURL
DeviceHive	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	
Google Cloud IoT Core	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	Google Cloud Pub/Sub, gRPC, DDS
IBM Watson IoT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	IBM MQ, IBM IoT Message Gateway, BACnet, KAFKA, JMS
IoTSens	✓	✓	✓			✓	✓	✓	✓	✓		✓	✓	NB-IoT
Kaa	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	Kaa Protocol (1/KP), STOMP, Admite protocolo de trasmisión personalizados
Macchina.IO EDGE	✓		✓	✓	✓	✓	✓		✓	✓	✓			SOAP, JSON-RPC, UPnP, CAN, CANopen, SNMP
Mainflux	✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	MQTT-SN, STOMP
Microsoft Azure IoT Central	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	Serial
Microsoft Azure IoT Hub	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	STOMP, MQTT-SN
OpenRemote	✓	✓	✓	✓	✓	✓		✓	✓				✓	KNX, Serial, SNMP, Wake-On-Lan, X10, Velbus, Telnet, AMX, Denon/Marantz, Domintell, DSC, Honeywell Z-Wave, Lutron, Russound, Samsung TV
Predix Platform	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	gRPC, Serial, SNMP
Sentilo	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	Serial, JMS
Sofia2	✓	✓	✓	✓	✓	✓			✓	✓	✓		✓	Serial, JMS
Thinger.io	✓		✓	✓	✓	✓				✓	✓		✓	PSON, Serial, LPWAN
ThingSpeak	✓		✓	✓	✓	✓								Microsoft Azure Event Hubs, Google Cloud Pub/Sub
Zetta	✓		✓	✓	✓	✓		✓	✓	✓	✓		✓	Serial, DDS, Agnóstico al protocolo de red

Característica II. Protocolos de Comunicación Compatibles

Factores clave a destacar:

Protocolos Comunes: La mayoría de las plataformas son compatibles con los protocolos de comunicación más comunes: MQTT, AMQP, REST, HTTP, HTTPS y CoAP. Estos protocolos son fundamentales en la mayoría de las soluciones IoT.

Soporte para Websockets: Varias plataformas IoT muestran compatibilidad con MQTT, AMQP y HTTP sobre WebSockets, tecnología que permite conexiones interactivas/en tiempo real

bidireccionales entre el cliente y el servido. Esto significa que estas plataformas pueden manejar la comunicación en tiempo real con dispositivos IoT, lo que las hace adecuadas para casos de uso que requieren la transmisión de datos en tiempo real.

Soporte de Protocolos Específicos: Algunas plataformas admiten protocolos más específicos o menos comunes, indicando una especialización o adaptabilidad para ciertos casos de uso. Por ejemplo:

- LoRaWAN: Soportado por IoTSens y Microsoft Azure IoT Central & Hub, entre otros.
- XMPP y LwM2M: Son soportados por un número limitado de plataformas, entre las que se encuentran IBM Watson IoT y Microsoft Azure IoT Hub.
- ModBus y OPC UA: Son soportados por Google Cloud IoT Core, IBM Watson IoT y Microsoft Azure IoT Central & Hub.
- DDS: Utilizado por Altair SmartWorks IoT, Google Cloud IoT Core y Zetta.

Protocolos Propietarios o Especiales: Algunas plataformas ofrecen soporte para protocolos personalizados o específicos de la empresa, que pueden ser indicativos de una especialización en ciertos casos de uso o de un ecosistema propio de la empresa.

- AWS IoT Device SDKs: Utilizado por Amazon Web Services IoT Core.
- IBM MQ, IBM IoT Message Gateway: Utilizado por IBM Watson IoT.
- Google Cloud Pub/Sub, gRPC: Utilizado por Google Cloud IoT Core.
- Kaa Protocol (1/KP), STOMP: Utilizado por Kaa.
- Serial: Soportado por varias plataformas, como Microsoft Azure IoT Central, Predix Platform, Sentilo y Sofia2.

1.2. Almacenamiento de datos:

Plataforma	Almacenamiento de Datos
Altair SmartWorks IoT	Admite el almacenamiento en la nube, en las propias instalaciones (local), híbrido y en el perímetro. Compatible con BBDD NoSQL: AnythingDB y RealtimeDB.
AWS IoT Core	AWS IoT Core almacena los datos de los dispositivos en AWS Cloud, que ofrece diversas opciones de almacenamiento de datos, como Amazon Simple Storage Service (S3), Amazon DynamoDB, Amazon Redshift y Amazon RDS.
Carriots	Carriots pone a tu disposición una gran base de datos NoSQL (Amazon S3) que puede almacenar cientos de terabytes de datos con al menos dos servidores redundantes independientes y una gran capacidad transaccional

DeviceHive	Almacenamiento en la nube, almacenamiento local y almacenamiento híbrido. Ofrece diversas opciones de almacenamiento de datos, en BBDD Relacionales: MySQL, PostgreSQL; y en NoSQL: Cassandra, MongoDB, Elasticsearch,
Google Cloud IoT Core	Datos almacenados y procesados mediante Google Cloud Storage, Bigtable, Datastore, Cloud Spanner, BigQuery y Cloud Pub/Sub. Aprovecha múltiples opciones de almacenamiento para un almacenamiento eficiente de los datos. Google Cloud Storage destaca por ser altamente escalable y fiable.
IBM Watson IoT	Ofrece almacenamiento de datos no estructurados (mediante el servicio: IBM Cloud Object Storage data buckets) en local, en híbrido y en la nube. Compatible con el uso de bases de datos relacionales (PostgresSQL, IBM Db2 Warehouse on Cloud) y NoSQL. Se integra con plataformas de almacenamiento de datos de terceros
IoTSens	Compatible con almacenamiento en la nube, local e híbrido.
Kaa	Preintegrada con bases de datos listas para la producción, como Cassandra, MongoDB, InfluxDB y otras. Permite almacenamiento tanto en local como en la nube
Macchina.IO EDGE	Admite opciones de almacenamiento local, en la nube e híbrido, junto con almacenamiento de bases de datos como InfluxDB (DB diseñada para almacenar los datos de series temporales generados por cada dispositivo).
Mainflux	CassandraDB, MongoDB, InfluxDB, PostgreSQL
M. Azure IoT Central	Almacena los datos en la nube de Azure mediante Azure Blob Storage, lo que permite acceder a ellos desde cualquier lugar.
M. Azure IoT Hub	Ofrece opciones de almacenamiento de datos como Azure Storage (Azure Blob Storage y Azure Data Lake Storage entre otros), Azure SQL Database, Table Storage, Cosmos DB, almacenamiento en la nube, almacenamiento local, almacenamiento híbrido, políticas de retención de datos y purga de datos. También se integra con plataformas de almacenamiento de terceros. Usa enrutamiento de mensajes para enviar datos de telemetría desde los servicios de Azure de dispositivos IoT a Azure Storage.
OpenRemote	Admite proveedores de almacenamiento en la nube (Amazon S3, Google Cloud Storage, Microsoft Azure), almacenamiento local y almacenamiento híbrido.
Predix Platform	Ofrece distintas alternativas para el almacenamiento de datos: PostgreSQL, Cassandra, Amazon S3, Microsoft Azure, Google Cloud Platform, Google Cloud Storage, Microsoft Azure Blob Storage, etc....
Sentilo	Compatible con MongoDB, MySQL, Cassandra, PostgreSQL, Elasticsearch y Memory BD.
Sofia2	La plataforma plantea el uso de tres repositorios distintos uno encargado de almacenar la información recibida en tiempo real (Base de Datos Tiempo Real (BDTR)), otro donde se almacena la información histórica para su posterior explotación analítica (Base de datos Histórica (BDH)) y, por último, uno donde almacenar información en diferentes estados (estructurada, semi-estructurada y no estructurada) temporalmente (Repositorio de Staging (HDFS))
Thinger.io	Almacena la información en la nube en soluciones seguras, eficientes y escalables como por ejemplo MongoDB, DynamoDB o AmazonS3. También emplea Databucket para almacenar datos de series temporales.
ThingSpeak	ThingSpeak trabaja sobre el concepto de "Canales". Un canal es donde se almacenan los datos recibidos de dispositivos y aplicaciones de terceros
Zetta	Admite servidores locales, en la nube, almacenamiento en dispositivos periféricos y bases de datos en memoria, relacionales y NoSQL, incluidas MongoDB, MySQL y PostgreSQL.

Característica III. Almacenamiento de Datos

Síntesis de los factores clave relacionados con el almacenamiento de datos en diferentes plataformas IoT:

Diversidad de Ubicación de Almacenamiento: La mayoría de las plataformas ofrecen una amplia gama de opciones de ubicación de almacenamiento, que incluyen almacenamiento en

la nube, local, híbrido y en el perímetro. Esto proporciona flexibilidad dependiendo de las necesidades específicas de seguridad y acceso a datos.

Soporte para Diversas Bases de Datos: Las plataformas generalmente soportan una variedad de bases de datos, tanto relacionales como NoSQL. Por ejemplo, MySQL, PostgreSQL, Cassandra, MongoDB e InfluxDB son algunas de las bases de datos compatibles mencionadas con frecuencia. El almacenamiento de datos no estructurados es útil para manejar datos diversos y no normalizados que son comunes en los entornos de IoT

Capacidad de Almacenamiento Masivo: Algunas plataformas, como AWS IoT Core y Carriots, proporcionan una capacidad de almacenamiento masivo al utilizar soluciones como Amazon S3.

Almacenamiento especializado de datos IoT: Algunas plataformas, como Macchina.IO EDGE y Thinger.io, utilizan bases de datos diseñadas específicamente para datos IoT, ofreciendo opciones específicas para el almacenamiento de series temporales, lo cual es útil para el manejo de datos de sensores y telemetría a lo largo del tiempo.

Enfoques Avanzados de Almacenamiento: Algunas plataformas proponen enfoques innovadores para el almacenamiento de datos. Por ejemplo, Sofia2 emplea diferentes repositorios para distintos tipos de datos y aplicaciones, mientras que ThingSpeak se basa en el concepto de "Canales" para el almacenamiento de datos.

1.3. Análisis, Procesamiento y Visualización de datos:

Plataforma	Análisis de Datos
Altair SmartWorks IoT	Proporciona herramientas de análisis de datos: *machine learning*, análisis predictivo, detección de anomalías, flujo de datos en tiempo real, análisis de datos históricos, visualización de datos y análisis estadístico.
AWS IoT Core	AWS IoT Core proporciona diversas herramientas y servicios de análisis de datos, como Amazon Kinesis Analytics, Amazon QuickSight, Amazon Athena, streaming de datos en tiempo real, gestión por lotes (*batch*), *machine learning* y se integra con Amazon Redshift y Amazon S3.
Carriots	Proporciona herramientas de análisis de datos: visualización de datos en tiempo real, análisis de datos históricos, *machine learning* e inteligencia artificial.
DeviceHive	Apache Spark, Elasticsearch, Cassandra y Kafka para análisis en tiempo real y sin conexión, visualización de datos, gestión por lotes (*batch*) y *machine learning*.
Google Cloud IoT Core	Se integra con servicios como Google Cloud Pub/Sub, Dataflow, BigQuery, Cloud Storage, Cloud Bigtable, Cloud Datastore y Cloud Dataproc para el análisis avanzado, almacenamiento y procesamiento en tiempo real de flujo de datos. También admite *machine learning*.
IBM Watson IoT	Ofrece *machine learning*, análisis predictivo, detección de anomalías, herramientas de visualización de datos (cuadros de mando, gráficos) e inteligencia artificial. Se integra con plataformas de análisis de datos de terceros.
IoTSens	Proporciona herramientas y funciones de análisis de datos como: visualización de datos, minería de datos, *machine learning* e inteligencia artificial.

Kaa	Proporciona análisis en tiempo real, análisis históricos, análisis *batch* y *machine learning*.
Macchina.IO EDGE	Proporciona un motor de análisis de datos integrado que permite: analizar datos de sensores y generar informes, visualizar datos y emplear *machine learning* e inteligencia artificial. Puede integrarse con plataformas de análisis de terceros.
Mainflux	Visualización de datos y series temporales, análisis históricos y en tiempo real, *machine learning*, inteligencia artificial, motor de análisis integrado y permite la integración con herramientas de *Bussines Intelligence* (*BI*, inteligencia empresaria) de terceros.
M. Azure IoT Central	Proporciona múltiples capacidades de análisis de datos, como *machine learning*, inteligencia artificial, análisis en tiempo real, análisis histórico, análisis predictivo, análisis de *streaming* en tiempo real, visualización de datos en *dashboards* en tiempo real, almacenamiento de datos históricos y análisis *batch*.
M. Azure IoT Hub	Ofrece funciones de análisis de datos, como análisis en tiempo real, análisis de *streaming* en tiempo real, análisis *batch*, *machine learning*, visualización de datos, inteligencia artificial, Power BI, Azure Data Studio, y se integra con plataformas de análisis de datos de terceros.
OpenRemote	Proporciona herramientas de análisis de datos como visualización de datos mediante *dashboards*, generación de informes, alertas, minería de datos, algoritmos de *machine learning* y admite la integración de herramientas de análisis de datos de terceros.
Predix Platform	Proporciona herramientas y servicios de análisis de datos: análisis de *streaming* en tiempo real, análisis *batch*, *machine learning*, análisis predictivo, detección de anomalías e inteligencia artificial orientada al ámbito empresarial.
Sentilo	Funciones integradas de análisis de datos, como agregación, visualización y correlación de datos, *machine learning* y detección de anomalías.
Sofia2	Proporciona funciones tales como *machine learning* y análisis en tiempo real, histórico y predictivo.
Thinger.io	Permite el análisis de datos históricos y en tiempo real, incluyendo funciones de *machine learning*.
ThingSpeak	Proporciona visualización de datos, estadísticas, gráficos, cuadros de mando y alertas. También admite funciones de *machine learning* e inteligencia artificial.
Zetta	Proporciona visualización de datos, minería de datos, *machine learning* y un motor de análisis de datos integrado.

Característica IV. Análisis de Datos

Los puntos clave extraidos de la tabla son:

Machine Learning, Inteligencia Artificial y Análisis Predictivo: Prácticamente todas las plataformas implementan herramientas para el análisis de datos utilizando machine learning, inteligencia artificial y análisis predictivo, lo que permite no solo analizar los datos existentes sino también prever tendencias y comportamientos futuros.

Análisis en Tiempo Real e Histórico: La mayoría de las plataformas soportan análisis de datos en tiempo real (procesamiento de flujo de datos conforme llegan) y análisis histórico (análisis de datos acumulados con el tiempo). El análisis en tiempo real es especialmente útil en situaciones donde los eventos ocurren rápidamente y se requiere una respuesta inmediata, en este aspecto destacan AWS IoT Core y Google Cloud IoT Core al integrarse con Amazon Kinesis Analytics y Google Cloud Pub/Sub respectivamente. El análisis histórico permite estudiar patrones y tendencias a lo largo del tiempo, plataformas como Altair SmartWorks IoT y Carriots proporcionan este tipo de análisis.

Visualización de Datos: Muchas plataformas también proporcionan herramientas de visualización de datos, lo que facilita la interpretación de los datos. Estas pueden incluir cuadros de mando y gráficos. Ejemplos de plataformas que incluyen estas herramientas: DeviceHive y OpenRemote.

Detección de Anomalías: Algunas plataformas incluyen herramientas para la detección de anomalías (identificación de datos que se desvían significativamente de los patrones esperados), lo que puede ser crucial para identificar y tratar problemas potenciales de manera temprana. Ejemplos de plataformas que cuentan con esta capacidad: Altair SmartWorks IoT y IBM Watson IoT.

Integración con Plataformas de Terceros: Varias plataformas permiten la integración con otras herramientas y plataformas de análisis de datos de terceros, lo que proporciona a los usuarios una mayor flexibilidad y la capacidad de utilizar la herramienta que mejor se adapte a sus necesidades. Ejemplos de plataformas que cuentan con esta capacidad: IBM Watson IoT, Macchina.IO EDGE y M. Azure IoT Hub.

Gestión por Lotes (Batch): Algunas plataformas permiten el procesamiento de datos en lotes, que es útil para operaciones que no requieren una respuesta en tiempo real. Ejemplos de plataformas que cuentan con esta funcionalidad: AWS IoT Core y DeviceHive.

Servicios Específicos de Análisis de Plataformas Propietarias: Algunas plataformas, como AWS IoT Core, Google Cloud IoT Core y Microsoft Azure IoT, tienen sus propios servicios de análisis de datos, que brindan herramientas especializadas para tratar con los datos generados en sus respectivas plataformas, como Amazon Kinesis Analytics, Amazon QuickSight, Google Cloud Pub/Sub, Google Dataflow, Azure Power BI, etc.

Plataforma	Facilidad de Uso
Altair SmartWorks IoT	Diseñado para ser fácil de usar, con una interfaz sencilla e intuitiva de arrastrar y soltar, variedad de tutoriales y documentación.
AWS IoT Core	AWS IoT Core es un servicio fácil de configurar y utilizar que proporciona diversos recursos para ayudar a los desarrolladores a comenzar, como tutoriales, documentación y código de muestra. Ofrece un módulo gratuito para el aprendizaje y cuenta con una API sencilla e intuitiva respaldada por una completa documentación.
Carriots	Interfaz sencilla e intuitiva con un panel de control fácil de usar y un editor de aplicaciones de tipo "arrastrar y soltar".
DeviceHive	Fácil de usar, gracias a una interfaz web y un panel de control intuitivos. Ofrece documentación y tutoriales tanto para desarrolladores como para usuarios finales. Fácil de instalar y configurar.
Google Cloud IoT Core	Proporciona una interfaz web fácil de usar, CLI (Interfaz de Línea de Comandos), API, documentación y tutoriales. Diseñado para que resulte fácil para los desarrolladores, incluso sin experiencia previa en IoT.
IBM Watson IoT	Ofrece un sencillo e intuitivo panel de control y API basado en web, así como numerosas funciones para facilitar la conexión de dispositivos, la recopilación de datos y la creación de aplicaciones.

	Proporciona una IU sencilla y bien definida donde se puede añadir y gestionar los dispositivos simple y fácilmente, controlar el acceso al servicio IoT y supervisar el uso.
IoTSens	Diseñado para ser fácil de usar, con una interfaz sencilla y una amplia documentación y recursos de apoyo.
Kaa	Fácil de usar y amigable para desarrolladores.
Macchina.IO EDGE	Ofrece una *GUI* (interfaz gráfica de usuario) fácil de usar, *CLI* (interfaz de línea de comandos) y documentación completa. Puede ser configurado por usuarios no técnicos.
Mainflux	Interfaz de usuario intuitiva y fácil de usar, API bien documentada. Proporciona herramientas y documentación para que los usuarios puedan empezar a utilizarla.
M. Azure IoT Central	Azure IoT Central es una plataforma fácil de usar tanto para desarrolladores como para usuarios no técnicos. Simplifica la conexión de dispositivos, la gestión de datos y el desarrollo de aplicaciones con funciones como un editor de tipo "arrastrar y soltar" (ej. un editor web de *dashboards*), un motor de reglas y un simulador integrados y una interfaz gráfica de usuario (*GUI*).
M. Azure IoT Hub	Puede ser utilizado tanto por principiantes como por expertos. Proporciona una plataforma fácil de usar con herramientas, recursos, tutoriales, documentación y código de muestra para ayudar a los desarrolladores a gestionar los dispositivos y datos IoT.
OpenRemote	Plataforma que ofrece una interfaz web (*GUI*) y una interfaz de línea de comandos (*CLI*). Adecuada tanto para desarrolladores como para no desarrolladores.
Predix Platform	Es una plataforma fácil de usar, que ofrece una interfaz sencilla, un entorno de desarrollo, un mercado de aplicaciones ya creadas, una comunidad de desarrolladores y diversas herramientas y recursos para ayudar a desarrolladores y usuarios finales a empezar a trabajar.
Sentilo	Fácil de usar y gestionar, con una interfaz web (*GUI*) sencilla y una API bien documentada.
Sofia2	Interfaz sencilla y fácil de usar. Cuenta con una documentación detallada y fácil de seguir lo que la hace adecuada tanto para desarrolladores como para no desarrolladores.
Thinger.io	Plataforma fácil de usar, tanto para desarrolladores como para no desarrolladores. Es muy sencillo trabajar con ella gracias a la gran documentación paso a paso que posee.
ThingSpeak	Fácil de usar para desarrolladores y no desarrolladores gracias a su sencilla interfaz.
Zetta	Plataforma fácil de usar. Fácil de configurar y utilizar tanto para principiantes como para usuarios experimentados.

Característica V. Facilidad de Uso

Puntos clave de la tabla en términos de la facilidad de uso de cada plataforma:

Interfaz intuitiva y sencilla: Muchas plataformas tienen una interfaz fácil de usar que permite a los usuarios interactuar con ellas de manera sencilla. Por ejemplo, Altair SmartWorks IoT, Carriots, Sofia2, AWS IoT Core y Google Cloud IoT Core tienen interfaces intuitivas diseñadas para ser fáciles de usar, incluso algunas de ellas cuentan con interfaces de “arrastrar y soltar” que simplifican la creación y gestión de aplicaciones.

Documentación y recursos de apoyo: Todas las plataformas parecen ofrecer una amplia gama de documentación, tutoriales y recursos para ayudar a los usuarios a empezar a trabajar con ellas. AWS IoT Core, DeviceHive, Sofia2 y IoTSens son ejemplos de plataformas con una amplia documentación.

Accesibilidad para desarrolladores y usuarios no técnicos: Muchas de las plataformas, como Google Cloud IoT Core, Macchina.IO EDGE, Thinger.io y Azure IoT Central, son accesibles tanto para desarrolladores como para usuarios no técnicos, lo que indica una amplia usabilidad.

Interfaz de Línea de Comandos (CLI): Algunas plataformas ofrecen una Interfaz de Línea de Comandos, lo que permite a los usuarios más técnicos tener un mayor control y flexibilidad. Ejemplos de plataformas que ofrecen esta característica son Google Cloud IoT Core y Macchina.IO EDGE.

Interfaz gráfica de usuario (GUI): Muchas plataformas, como Macchina.IO EDGE, Azure IoT Central y OpenRemote, ofrecen una interfaz gráfica de usuario (GUI) que puede proporcionar una forma más visual y fácil de interactuar con la plataforma.

2. Seguridad:

Plataforma	Seguridad
Altair SmartWorks IoT	Características de seguridad: encriptación, autenticación, autorización, control de acceso basado en roles, encriptación de datos, y detección de intrusiones.
AWS IoT Core	Ofrece diversas funciones de seguridad, como autenticación de dispositivos, cifrado de extremo a extremo (todo el tráfico hacia y desde AWS IoT se envía de forma segura a través de *Transport Layer Security* (*TLS*)), cifrado de datos y control de acceso (*RBAC*) entre otras medidas.
Carriots	Características: cifrado de datos, autenticación de usuarios y control de acceso basado en roles. Utiliza medidas de seguridad estándar del sector para proteger los datos.
DeviceHive	Incorpora funciones de seguridad como autenticación de usuarios, cifrado de datos y control de acceso. Admite TLS, SSL, OAuth 2.0 y 2FA.
Google Cloud IoT Core	Ofrece sólidas funciones de seguridad, como autenticación de dispositivos, cifrado de datos de extremo a extremo, mecanismos de control de acceso y control de acceso basado en roles. Se integra con Google Cloud Identity and Access Management (*IAM*) para conceder permisos granulares de usuarios y dispositivos, el cual ofrece supervisión 24/7, detección de intrusiones. Este servicio está construido sobre la infraestructura de seguridad de clase mundial de Google.
IBM Watson IoT	Está diseñado con la seguridad como prioridad. Incluye cifrado, autenticación de dos factores, autorización, control de acceso basado en roles y otras funciones de seguridad para proteger datos y dispositivos. Admite protocolos de seguridad como TLS, IPsec.
IoTSens	Proporciona encriptación de datos, autenticación y autorización.
Kaa	Ofrece seguridad de extremo a extremo con funciones como cifrado de datos, control de acceso basado en roles (*RBAC*) y autenticación de dispositivos.
Macchina.IO EDGE	Proporciona un alto nivel de seguridad, incluido el cifrado TLS/SSL, el control de acceso basado en roles (*RBAC*) y medidas de prevención de pérdida de datos.
Mainflux	Proporciona control de acceso basado en roles (*RBAC*), cifrado de datos, autenticación, autorización, conexiones PostgreSQL seguras, transporte seguro mediante certificados y seguridad de extremo a extremo mediante 2 modalidades de comunicación encriptada: - *Manager service* (proporciona autenticación comprobando la validez de los JSON Web Tokens. - *NGINX* (proxy inverso) (forma un cortafuegos, cerrando todas las rutas no públicas para el acceso externo).

M. Azure IoT Central	Azure IoT Central es una plataforma segura que ofrece una serie de funciones de seguridad, como, por ejemplo: autenticación de dispositivos, cifrado de datos, control de acceso basado en roles (*RBAC*), auditoría, protección de datos y seguridad de datos.
M. Azure IoT Hub	Ofrece funciones de seguridad como autenticación, autorización, autenticación de dispositivos, cifrado de dispositivos, cifrado de datos y control de acceso basado en roles (*RBAC*).
OpenRemote	Utiliza medidas de seguridad para proteger los datos, como el cifrado, la autenticación y la autorización. La autenticación y autorización en el stack OpenRemote se realiza mediante Keycloak OpenID Connect Provider y utiliza OAuth 2.0. Gestión de identidades: Un activo es por defecto privado, sólo puede ser accedido por el superusuario o usuarios regulares de su dominio
Predix Platform	Diseñado con funciones de seguridad integradas: autenticación, autorización, control de acceso basado en roles (*RBAC*), detección de intrusiones y transferencia cifrada de datos de extremo a nube.
Sentilo	Admite diversas funciones de seguridad, como autenticación, autorización, cifrado de datos y control de acceso basado en roles (*RBAC*). La plataforma validará cualquier solicitud recibida por el sistema siguiendo la terminología *AAA, Authentication, Authorization & Accounting/Traceability* (Autenticación, Autorización y Trazabilidad).
Sofia2	Proporciona autenticación, autorización, control de acceso basado en roles (*RBAC*), copia de seguridad de los datos, seguridad en las comunicaciones, detección de intrusiones e integridad y cifrado de datos.
Thinger.io	Proporciona funciones como cifrado, autenticación y autorización.
ThingSpeak	Utiliza cifrado de datos (SSL/TLS), autenticación de usuarios y control de acceso basado en roles (*RBAC*).
Zetta	Proporciona autenticación, autorización y cifrado y sigue estándares de seguridad como OAuth 2.0, OpenID Connect y TLS 1.2.

Característica VI. Seguridad

A continuación, presentare los factores clave extraídos con respecto a las características de seguridad en las diferentes plataformas:

Encriptación de datos: Prácticamente todas las plataformas, como Altair SmartWorks IoT, AWS IoT Core y Carriots, ofrecen encriptación de datos para garantizar la seguridad de la información.

Autenticación: Se observa que la autenticación (proceso de verificar la identidad de un usuario, dispositivo o sistema y el cual asegura que el usuario es quien dice ser) es una característica común en todas las plataformas listadas. Algunas plataformas, como DeviceHive, incluso soportan autenticación de dos factores.

Autorización: Es el proceso que viene después de la autenticación. Una vez que un usuario ha sido autenticado, la autorización determina qué permisos tiene ese usuario (qué puede y no puede hacer, qué datos puede y no puede ver). Este es otro aspecto común de la seguridad en la mayoría de las plataformas, como Altair SmartWorks IoT, AWS IoT Core y Carriots.

Control de acceso basado en roles (RBAC): Este es un factor común en muchas plataformas, incluyendo Altair SmartWorks IoT, AWS IoT Core, Carriots y Google Cloud IoT Core, permitiendo un control granular sobre quién puede acceder a qué datos y funciones.

Integración con sistemas de gestión de la identidad existentes: Google Cloud IoT Core, por ejemplo, se integra con Google Cloud Identity and Access Management (IAM), mientras que OpenRemote utiliza Keycloak OpenID Connect Provider para gestionar la identidad y el acceso. (Punto elaborado en profundidad en la explicación de la siguiente tabla, Característica VII. Gestión de Usuarios y Roles).

Detección de intrusiones: Algunas plataformas, como Altair SmartWorks IoT, Predix Platform y Sofia2, tienen capacidades de detección de intrusiones para detectar y responder a cualquier actividad sospechosa.

Cifrado de extremo a extremo: AWS IoT Core y Google Cloud IoT Core son ejemplos de plataformas que garantizan la seguridad de los datos en tránsito mediante el cifrado de extremo a extremo.

Uso de protocolos de seguridad avanzados: Algunas plataformas utilizan estándares de seguridad reconocidos, como *Transport Layer Security* (TLS), *Secure Socket Layers* (SSL), OAuth 2.0, *Two Factor Authentication* (2FA) y OpenID Connect, como se observa en DeviceHive, IBM Watson IoT o Zetta, por ejemplo.

Plataforma	Gestión de Usuarios y Roles
Altair SmartWorks IoT	El sistema de gestión de usuarios y funciones permite a los Administradores crear y gestionar usuarios y funciones. Proporciona un sistema de control de acceso basado en roles (*role-based access control* (*RBAC*)) que permite definir permisos para usuarios y grupos.
AWS IoT Core	Es compatible con el control de acceso basado en roles (*RBAC*), es decir, permite crear y gestionar usuarios y roles, y ofrece funciones como la creación de usuarios, la creación de roles, la gestión de permisos, los privilegios de los roles y la asignación de roles.
Carriots	Permite crear y gestionar diferentes roles de usuario (ej: administrador, desarrollador, usuario). Permite controlar quién tiene acceso a los datos y dispositivos.
DeviceHive	Sistema de gestión de usuarios que permite la creación y gestión de cuentas de usuario, grupos y roles (permite crear diferentes roles de usuario con diferentes permisos).
Google Cloud IoT Core	Admite la gestión de usuarios y funciones. Se integra con Google Cloud IAM para ofrecer permisos granulares de usuarios y roles, lo que incluye autenticación de usuarios, autorización de usuarios, control de acceso basado en roles (*RBAC*), creación de usuarios, asignación de roles y control de acceso para gestionar quién tiene acceso a los datos y dispositivos IoT.
IBM Watson IoT	Proporciona un completo sistema de control de acceso basado en roles (*RBAC*) para gestionar a los usuarios y sus permisos a través del panel de control vía navegador web de la plataforma.
IoTSens	Proporciona un sistema de gestión de usuarios y funciones para crear y gestionar cuentas y roles de usuario.
Kaa	Gestiona usuarios y roles con control de acceso basado en roles (*RBAC*)
Macchina.IO EDGE	Proporciona un sistema de gestión de usuarios para crear y gestionar usuarios y roles.
Mainflux	Admite políticas de control de acceso basadas en roles (*RBAC*) y control de acceso de granularidad fina (*ABAC, attribute-based access control*).

M. Azure IoT Central	Proporciona un sistema de control de acceso basado en roles (RBAC) que permite definir permisos para usuarios, grupos y perfiles. Admite funciones de gestión de usuarios y roles, como la creación de usuarios, la asignación de roles, el control de acceso y la capacidad de gestionar permisos de usuario para controlar el acceso a datos y aplicaciones de IoT.
M. Azure IoT Hub	Admite funciones completas de gestión de usuarios y roles. Esto incluye gestión de usuarios, de roles y de permisos, autenticación de usuarios, control de acceso basado en roles (*RBAC*) y aprovisionamiento de usuarios.
OpenRemote	La autenticación y autorización en el stack OpenRemote se realiza mediante Keycloak OpenID Connect Provider. El cual proporciona un sistema de gestión de usuarios y roles que permite crear usuarios, asignarles roles y permisos.
Predix Platform	La plataforma Predix proporciona un sistema de control de acceso basado en roles (*RBAC*) para la gestión de usuarios y sus permisos, permitiendo a las organizaciones controlar el acceso a la plataforma y definir roles de usuario específicos.
Sentilo	Admite la gestión de usuarios y roles (*RBAC*), con capacidad para crear y gestionar usuarios, grupos, roles y permisos.
Sofia2	Gestiona a los usuarios y roles mediante características como su control de acceso basado en roles (*RBAC*) encargado de la gestión de usuarios, roles y permisos.
Thinger.io	Permite la creación y gestión de usuarios, funciones y grupos.
ThingSpeak	Admite la gestión de usuarios y roles y el control de acceso basado en roles (*RBAC*).
Zetta	Zetta permite la gestión de usuarios y roles. A los usuarios se les pueden asignar roles con diferentes permisos.

Característica VII. Gestión de Usuarios y Roles

los factores clave que podemos extraer en términos de gestión de usuarios y roles en las diferentes plataformas IoT son:

Control de acceso basado en roles (RBAC): La mayoría de las plataformas tienen un sistema de RBAC. Esto permite a los administradores asignar roles específicos a los usuarios y luego determinar qué acceso a los datos y dispositivos tienen esos roles. Esto es crucial para administrar de manera efectiva quién tiene permiso para hacer qué en la plataforma. En adición, Mainflux también ofrece Control de acceso de granularidad fina (ABAC) además de RBAC, lo que permite un control de acceso aún más detallado basado en atributos específicos, como la ubicación del usuario, el tipo de dispositivo que están utilizando, el horario, etc.

Gestión de usuarios: Casi todas las plataformas permiten a los administradores crear y gestionar cuentas de usuario. Esto es esencial para controlar quién tiene acceso a la plataforma y administrar ese acceso a lo largo del tiempo.

Gestión de roles: Al igual que con la gestión de usuarios, la mayoría de las plataformas permiten a los administradores crear y gestionar roles. Esto puede incluir la creación de nuevos roles, la modificación de roles existentes y la asignación de roles a usuarios.

Gestión de grupos: Algunas plataformas, como Altair SmartWorks IoT y Thinger.io, también permiten la creación y gestión de grupos de usuarios. Esto puede ser útil para gestionar el acceso y los permisos en un nivel más amplio.

Integración con sistemas de gestión de la identidad existentes: Algunas plataformas IoT pueden integrarse con sistemas de gestión de identidad ya existentes. Estos sistemas proporcionan un marco para la autenticación y la autorización, a menudo proporcionando funcionalidades adicionales como la gestión de contraseñas, la autenticación de dos factores, el inicio de sesión único, etc. En la tabla encontramos como ejemplo a Google Cloud IoT Core que se integra con Google Cloud Identity and Access Management (IAM) y OpenRemote que utiliza Keycloak OpenID Connect Provider.

3. Gestión de dispositivos:

Plataforma	Gestión de Dispositivos	¿La plataforma permite enviar comandos a los dispositivos?
Altair SmartWorks IoT	Funciones de gestión de dispositivos: registro de dispositivos, actualizaciones de firmware, control remoto, aprovisionamiento de dispositivos y supervisión remota.	✓
AWS IoT Core	Ofrece diversas funciones de gestión de dispositivos, como el registro de dispositivos, las actualizaciones de firmware, la gestión remota de dispositivos, el aprovisionamiento de dispositivos, la monitorización de dispositivos y Device Shadow para almacenar el estado de los dispositivos, lo que facilita la conexión, la gestión y el escalado de flotas de dispositivos.	✓
Carriots	Funciones como aprovisionamiento de dispositivos (inscripción, configuración y autenticación en la red de la organización), actualizaciones de firmware y supervisión remota.	✓
DeviceHive	Ofrece registro, aprovisionamiento, supervisión, control remoto y actualizaciones de firmware de dispositivos.	✓
Google Cloud IoT Core	Ofrece funciones completas de gestión de dispositivos, incluidos el aprovisionamiento, el registro, la supervisión, las actualizaciones de firmware y la gestión del ciclo de vida de los dispositivos IoT.	✓
IBM Watson IoT	Ofrece funciones integrales, como registro de dispositivos, actualizaciones de firmware y monitorización y control remotos. Los usuarios pueden gestionar los dispositivos desde una ubicación geográfica remota.	✓
IoTSens	Ofrece aprovisionamiento de dispositivos, supervisión y actualizaciones de firmware.	✓
Kaa	Ofrece actualizaciones de firmware y configuración, supervisión, registro y gestión remota de dispositivos.	✓
Macchina.IO EDGE	Proporciona registro, configuración, gestión remota y supervisión de dispositivos y actualizaciones de firmware.	✓
Mainflux	Proporciona aprovisionamiento y supervisión de dispositivos, actualizaciones de firmware y gestión remota CRUD (Crear, Leer, Actualizar, Eliminar).	✓
M. Azure IoT Central	Facilita diversas funciones de gestión de dispositivos, como el aprovisionamiento, el diagnóstico, la supervisión remota y las actualizaciones de firmware de dispositivos.	✓
M. Azure IoT Hub	Ofrece funciones de gestión de dispositivos como: aprovisionamiento, registro, supervisión del estado y gestión centralizada de dispositivos, además de actualizaciones de firmware.	✓
OpenRemote	Proporciona funciones de gestión de dispositivos como seguimiento de activos, actualizaciones de firmware, solución remota de problemas y herramientas personalizadas de gestión de dispositivos, además de control remoto, detección, aprovisionamiento y supervisión de dispositivos.	✓
Predix Platform	Proporciona aprovisionamiento de dispositivos, actualizaciones de firmware, supervisión remota, detección de dispositivos, etc…	✓

Sentilo	Admite el registro, configuración, supervisión, aprovisionamiento, actualización de firmware y control remoto de dispositivos.	✓
Sofia2	Ofrece actualizaciones de *firmware* y registro, aprovisionamiento, supervisión remota y configuración de dispositivos.	✓
Thinger.io	Incluye funciones como el registro, la configuración, la supervisión, el aprovisionamiento y la actualización de dispositivos.	✓
ThingSpeak	Incluye registro de dispositivos, supervisión de estado, actualizaciones de firmware, configuración y control remoto.	✓
Zetta	Ofrece funciones como detección de dispositivos, aprovisionamiento, supervisión, actualizaciones de firmware y configuración de ajustes.	✓

Característica VIII. Gestión de Dispositivos y ¿La plataforma permite enviar comandos a los dispositivos?

Resumen conciso de los factores clave detectados en términos de gestión de dispositivos en las plataformas de IoT:

Envió de comandos a los dispositivos: Todas las plataformas listadas permiten enviar comandos a los dispositivos, permitiendo así realizar ajustes o cambios en los dispositivos de forma remota.

Registro de dispositivos: Todas las plataformas admiten el registro de dispositivos, funcionalidad que permite a las plataformas de IoT agregar y registrar nuevos dispositivos en el sistema, lo que es un paso fundamental para la gestión y supervisión de los dispositivos. Algunas plataformas como Predix Platform y Zetta, ofrecen la capacidad de detectar dispositivos automáticamente, lo que puede facilitar la tarea de registro y aprovisionamiento de nuevos dispositivos que podrían ser agregados a la red.

Aprovisionamiento de dispositivos: La mayoría de las plataformas, como AWS IoT Core y Carriots, disponen de funciones de aprovisionamiento que facilitan la inscripción, configuración y autenticación de los dispositivos, preparándolos para su integración en la red de la organización. Este proceso de aprovisionamiento es solo una parte del ciclo de vida más amplio de los dispositivos IoT. En este sentido, algunas plataformas, como Google Cloud IoT Core, van más allá, proporcionando una gestión completa del ciclo de vida de estos dispositivos, abarcando desde su inicial aprovisionamiento hasta su eventual retirada del sistema.

Actualizaciones de firmware: Casi todas las plataformas proporcionan la funcionalidad para realizar actualizaciones de firmware de los dispositivos desde la plataforma, lo que es característica crucial para mantener los dispositivos seguros y actualizados.

Supervisión y control remotos: Estas son características comunes en la mayoría de las plataformas que permiten a los usuarios rastrear el estado y monitorear el rendimiento de los dispositivos en tiempo real, y controlar los dispositivos desde ubicaciones remotas favoreciendo la solución de problemas y el ajuste de configuraciones.

Herramientas de diagnóstico y seguimiento de activos: Son características adicionales proporcionadas por algunas plataformas, como M. Azure IoT Central y OpenRemote, para ayudar en la identificación de problemas y el seguimiento de la ubicación de los dispositivos.

4. Habilitación y gestión de aplicaciones:

Plataforma	SDK & Lenguajes Soportados para Desarrollar										
	Java	Python	C/C++	JavaScript/Node.js	C#	Go	.NET	Android	iOS	Arduino	Otros SDK/ Lenguajes
Altair SmartWorks IoT	✓	✓	✓	✓				✓			Sigfox
Amazon Web Service IoT Core	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Carriots	✓	✓	✓	✓							Groovy
DeviceHive	✓	✓	✓	✓	✓	✓	✓				Ruby
Google Cloud IoT Core	✓	✓	✓	✓	✓	✓	✓				Ruby
IBM Watson IoT	✓	✓	✓	✓	✓		✓	✓	✓	✓	Ruby
IoTSens	✓	✓	✓	✓	✓		✓	✓	✓	✓	Sigfox, Ruby, Groovy
Kaa	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Macchina.IO EDGE	✓	✓	✓	✓				✓	✓	✓	Macchina.io REMOTE SDK
Mainflux	✓	✓	✓	✓	✓	✓					
Microsoft Azure IoT Central & Hub	✓	✓	✓	✓	✓		✓	✓	✓		UWP
OpenRemote	✓	✓	✓	✓	✓		✓				Ruby, Groovy
Predix Platform	✓	✓	✓	✓	✓		✓	✓	✓	✓	
Sentilo	✓	✓	✓	✓							Sigfox
Sofia2	✓	✓	✓	✓	✓		✓	✓	✓	✓	
Thinger.io	✓	✓	✓	✓						✓	Sigfox, (Visual Studio Code con) PlatformIO
ThingSpeak	✓	✓	✓	✓						✓	MATLAB
Zetta	✓	✓	✓	✓							

Característica IX. SDK & Lenguajes Soportados para Desarrollar

A continuación, presentare los factores clave extraídos con respecto a los SDK y Lenguajes de programación soportados para desarrollar en las diferentes plataformas:

Soporte universal de lenguajes: Todas las plataformas listadas soportan Java, Python, C/C++, y JavaScript/Node.js. Esto indica una amplia compatibilidad y flexibilidad para los desarrolladores en términos de lenguajes de programación que pueden utilizar. Por ejemplo, un desarrollador que prefiera Python podría optar por cualquier plataforma de las listadas.

Soporte para una gama amplia de lenguajes: Plataformas como AWS IoT Core, IBM Watson IoT, y Kaa soportan una gama amplia de lenguajes, desde los ampliamente usados como C# y Go, hasta los específicos para desarrollo de aplicaciones móviles como Android e iOS, y para

hardware como Arduino. Esto sugiere que estas plataformas serían adecuadas para un proyecto que requiera, por ejemplo, una aplicación en Android y un dispositivo Arduino.

Plataformas con SDK exclusivos: Algunas plataformas ofrecen SDK exclusivos, como Macchina.IO EDGE con su Macchina.io REMOTE SDK, Microsoft Azure IoT Central & Hub con su *Universal Windows Platform* (UWP) o Thinger.io, que además de soportar desarrollo para hardware Arduino y Sigfox, permite el uso de Visual Studio Code con PlatformIO. Por otra parte, la única plataforma que cuenta con soporte para desarrollo en MATLAB es ThingSpeak.

Lenguajes adicionales: Aunque hay un conjunto de lenguajes comúnmente soportados, algunas plataformas también brindan soporte para lenguajes menos comunes o más específicos, como:

- Ruby en DeviceHive, Google Cloud IoT Core, IBM Watson IoT, IoTSens y OpenRemote.
- Groovy en Carriots, IoTSens y OpenRemote.
- Sigfox en Altair SmartWorks IoT, IoTSens, Sentilo y Thinger.io.

Soporte para desarrollo de aplicaciones móviles: Varias plataformas, como IBM Watson IoT, IoTSens, Kaa, Microsoft Azure IoT Central & Hub, Predix Platform, Sofia2, y Macchina.IO EDGE, soportan tanto Android como iOS, facilitando el desarrollo de aplicaciones móviles. Por ejemplo, un desarrollador que busque crear una aplicación IoT para iOS y Android podría considerar IBM Watson IoT como una opción viable.

Soporte para C#: La mayoría de las plataformas soportan C#, existen algunas excepciones notablemente como Altair SmartWorks IoT, Carriots, Macchina.IO EDGE, Sentilo, Thinger.io, ThingSpeak y Zetta no lo hacen. Esto podría influir en la decisión de los desarrolladores que prefieren utilizar C#, pudiendo optar por plataformas como Amazon Web Service IoT Core o IBM Watson IoT.

Soporte para .NET: Muchas plataformas admiten .NET, lo que podría ser especialmente relevante para los desarrolladores que trabajan con tecnologías de Microsoft o que prefieren este marco de trabajo.

Uso de Go: Sólo algunas plataformas, como AWS IoT Core, DeviceHive, Google Cloud IoT Core, Kaa y Mainflux, soportan Go. Esto puede ser un punto clave para los desarrolladores que prefieran utilizar Go por su eficiencia y sintaxis sencilla, pudiendo seleccionar la plataforma Google Cloud IoT Core que además ofrece la ventaja de integrarse con otros servicios de Google.

Plataforma	Soporte para Desarrollo de Aplicaciones
Altair SmartWorks IoT	Funciones de desarrollo de aplicaciones: desarrollo mediante un constructor de “arrastrar y soltar”, scripting visual, generación de código, editor de código, testeo y un simulador.
AWS IoT Core	Proporciona una variedad de funciones para el desarrollo de aplicaciones, como Device Shadow, un motor de reglas, integración con AWS Lambda, Amazon API Gateway, Amazon Cognito y Alexa Voice Service.
Carriots	Funciones de desarrollo de aplicaciones: Constructor de tipo "arrastrar y soltar", API REST, IDE web, biblioteca de componentes pre-construidos y mercado de aplicaciones de terceros entre otras funciones.
DeviceHive	Amplia gama de APIs, herramientas, recursos y bibliotecas para el desarrollo rápido de aplicaciones IoT.
Google Cloud IoT Core	Proporciona una serie de herramientas y servicios para el desarrollo de aplicaciones, como Cloud Functions, Cloud Run, Device Shadow, comandos de dispositivos, eventos de dispositivos, integración de dispositivos, análisis de datos, visualización de datos, mensajería de dispositivo a nube (y de nube a dispositivo) y servicios como Google Cloud SDK, Google Cloud Platform Console y Google Cloud APIs. Permite a los desarrolladores crear y desplegar aplicaciones IoT que procesen datos en tiempo real y activen acciones basadas en la telemetría del dispositivo.
IBM Watson IoT	Proporciona herramientas y recursos, como API, SDK, modelos, un IDE basado en la nube, una biblioteca de código de muestra y herramientas integradas para ayudar al desarrollo de aplicaciones IoT.
IoTSens	Ofrece un entorno de desarrollo, una biblioteca de componentes prefabricados y un espacio de compra de aplicaciones de terceros.
Kaa	Permite el desarrollo rápido de aplicaciones y es compatible con aplicaciones hechas a medida. Cuenta con un constructor de interfaz de usuario de "arrastrar y soltar".
Macchina.IO EDGE	Admite el desarrollo rápido de aplicaciones con componentes y servicios preconstruidos, APIs, SDKs. Su elaborada documentación también es de gran ayuda.
Mainflux	Admite una amplia gama de aplicaciones IoT, creador de interfaces de usuario, motor de reglas, sistema de notificaciones, herramientas de desarrollo de aplicaciones y modelos/plantillas.
M. Azure IoT Central	Ofrece un amplio conjunto de funciones de desarrollo de aplicaciones, como un mercado integrado, un portal para desarrolladores, diversos SDK y lenguajes, plantillas de aplicaciones, una interfaz de "arrastrar y soltar", un motor de reglas visual integrado, un editor de código integrado, la posibilidad de crear aplicaciones personalizadas e integrarse con aplicaciones existentes mediante conectores de aplicaciones.
M. Azure IoT Hub	Proporciona funciones de desarrollo de aplicaciones, incluido un agente (*broker*) de mensajes integrado, una API REST, Azure Functions, Azure Stream Analytics, Azure Machine Learning, comandos, eventos y gemelos (*device twins*) de dispositivos, junto con plantillas y SDKs de aplicaciones.
OpenRemote	Proporciona funciones de desarrollo de aplicaciones como: un creador de reglas de tipo "arrastrar y soltar", una API REST, funciones para la creación, implantación y supervisión de aplicaciones, automatización basada en reglas y cuadros de mando personalizados. Admite herramientas de desarrollo de aplicaciones personalizadas y de terceros.
Predix Platform	Ofrece herramientas y servicios como: un entorno de desarrollo, repositorio de código, entorno de testeo y apoya el desarrollo rápido de aplicaciones con componentes preconstruidos.
Sentilo	Ofrece varios frameworks de desarrollo de aplicaciones, incluidos Node.js, Spring Boot y Django. Admite modelos de datos personalizados, reglas, alertas, plantillas y herramientas para el desarrollo y puesta en marcha.
Sofia2	Desarrollo rápido de aplicaciones IoT gracias a su generador de aplicaciones con interfaz de "arrastrar y soltar" y su compatibilidad con SDKs para lenguajes populares.
Thinger.io	Ofrece herramientas para el desarrollo de aplicaciones personalizadas, como, por ejemplo, un creador de cuadros de mando de tipo "arrastrar y soltar", una API REST o un broker MQTT.
ThingSpeak	Ofrece un creador de aplicaciones de tipo "arrastrar y soltar", una API REST, una API MATLAB y funciones de creación, despliegue y gestión de aplicaciones.
Zetta	Proporciona una API enriquecida, un IDE integrado, integración de dispositivos, visualización de datos y un paquete de herramientas y librerías.

Característica X. Soporte para Desarrollo de Aplicaciones

De acuerdo con la tabla, estos son los puntos clave:

Constructores de aplicaciones "arrastrar y soltar": Muchas plataformas como Altair SmartWorks IoT, Carriots, Kaa, Microsoft Azure IoT Central, OpenRemote, Sofia2 y ThingSpeak proporcionan funcionalidades de creación un constructor de interfaces de usuario de tipo “arrastrar y soltar”. Este enfoque simplifica la creación de interfaces de usuario y acelera el desarrollo de aplicaciones y permite a los usuarios sin experiencia en programación crear interfaces de forma rápida y sencilla.

Integración con otros servicios: Algunas plataformas como AWS IoT Core, Google Cloud IoT Core, Microsoft Azure IoT Central y Hub ofrecen integración con otros servicios de su respectiva suite de productos. Por ejemplo, AWS IoT Core se integra con AWS Lambda, Amazon API Gateway, Amazon Cognito y Alexa Voice Service, mientras que Google Cloud IoT Core ofrece integración con Google Cloud Functions y Google Cloud Run.

Herramientas de desarrollo rápidas: Plataformas como DeviceHive, Kaa, Macchina.IO EDGE y Sofia2 ofrecen un conjunto de herramientas, recursos y bibliotecas para el desarrollo rápido de aplicaciones IoT, lo que puede agilizar el proceso de desarrollo.

Componentes y servicios preconstruidos y Mercado de aplicaciones de terceros: Plataformas como Carriots, IoTSens, Predix y Macchina.IO EDGE ofrecen componentes preconstruidos y/o un mercado de aplicaciones de terceros. Esto permite a los desarrolladores acelerar el desarrollo y ampliar las posibles funcionalidades de sus aplicaciones utilizando componentes ya existentes. Por ejemplo, un desarrollador en IoTSens puede usar una aplicación preexistente para el monitoreo de la calidad del aire desde el mercado de aplicaciones de terceros.

Motor de reglas: AWS IoT Core, Mainflux, Microsoft Azure IoT Central, OpenRemote y Sentilo proporcionan un motor de reglas, lo que permite a los desarrolladores crear acciones condicionales sin necesidad de programar.

Herramientas de análisis y visualización de datos: Algunas plataformas, como Google Cloud IoT Core y Zetta, proporcionan herramientas para el análisis y visualización de datos. Estas herramientas pueden ser útiles para entender e interpretar los patrones y tendencias en los datos recopilados por los dispositivos IoT. Por ejemplo, un desarrollador en Google Cloud IoT Core puede usar Google Cloud API’s para analizar y visualizar los datos de los dispositivos IoT en tiempo real.

5. Capacidades del sistema:

5.1. Extensibilidad/Escalabilidad:

Plataforma	Escalabilidad
Altair SmartWorks IoT	Soporta un gran número de dispositivos y datos. Puede satisfacer las necesidades de cualquier organización, desde pequeñas empresas hasta grandes corporaciones.
AWS IoT Core	AWS IoT Core es un servicio altamente escalable que puede soportar miles de millones de dispositivos conectados y enrrutar billones de mensajes.
Carriots	Escalado automático para gestionar de 1 a millones de dispositivos (no hay limitación de cuánto pueden escalar tus proyectos tanto en volumen como en rapidez)
DeviceHive	Sistema basado en microservicios, construido con alta escalabilidad y disponibilidad. Plataforma que no solo puede escuchar cientos de dispositivos simultáneamente, sino también escalar a la cantidad requerida de instancias para garantizar la seguridad, disponibilidad de los datos y para gestionar el aumento de los niveles de producción. Gestionado y orquestado por Kubernetes.
Google Cloud IoT Core	Diseñado para gestionar millones de dispositivos y escalar horizontalmente para adaptarse a despliegues IoT en constante crecimiento. Puede manejar grandes volúmenes de ingestión de datos, procesamiento y gestión de dispositivos de forma eficiente.
IBM Watson IoT	Escalable horizontalmente. Capaz de soportar despliegues de IoT de pequeño a gran tamaño y gestionar millones de dispositivos y miles de millones de mensajes al día. Watson IoT es un servicio en la nube y, como tal, a medida que aumente la demanda de cualquier servicio, se escalará la capacidad para satisfacer esa demanda. Los usuarios no tienen que preocuparse por eso, se hace automáticamente.
IoTSens	Plataforma horizontal y transversal capaz de evolucionar y ser ampliable gracias a la organización de cada una de sus capas.
Kaa	Con la arquitectura de microservicios IoT preparada por Kubernetes puede escalar infinitamente y manejar millones de dispositivos.
Macchina.IO EDGE	Alta escalabilidad, hasta diez mil dispositivos por instancia de servidor REMOTE de macchina.io (se pueden agrupar varios servidores (escalabilidad horizontal) para aumentar la capacidad hasta millones de dispositivos).
Mainflux	Escalable horizontalmente. Admite millones de dispositivos y es fácilmente escalable. Compatible con la implantación en la nube y en las instalaciones. Permite despliegue mediante Contenedores Docker.
M. Azure IoT Central	Es una plataforma altamente escalable, capaz de soportar millones de dispositivos, miles de millones de mensajes al día y manejar grandes volúmenes de datos. Se puede escalar tanto horizontal como verticalmente.
M. Azure IoT Hub	Altamente escalable. Admite millones de despliegues de IoT, desde pequeños a grandes, incluyendo la partición de dispositivos y el escalado automático. Se puede escalar horizontal o verticalmente.
OpenRemote	Plataforma escalable que puede utilizarse para gestionar un gran número de dispositivos, datos y usuarios. Admite escalado horizontal y vertical.
Predix Platform	Diseñado para ser escalable y poder gestionar millones de dispositivos, lo que lo hace adecuado para implantaciones de IoT a gran escala. A su vez, Predix ha demostrado que la ingesta de datos, el procesamiento analítico y la gestión de operaciones son capaces de adaptarse y escalar para satisfacer las exigentes cargas de trabajo de los líderes industriales.
Sentilo	Escalable horizontalmente para admitir un gran número de dispositivos y sensores. Admite desde servidores individuales a clústeres de hasta 10.000 nodos.
Sofia2	Ofrece escalabilidad tanto horizontal como vertical.
Thinger.io	Altamente escalable. Puede albergar un gran número de dispositivos y usuarios.

ThingSpeak	Plataforma escalable para proyectos de gran envergadura. Admite millones de dispositivos y puede gestionar grandes volúmenes de datos (hasta miles de millones de datos).
Zetta	Plataforma escalable. Puede gestionar un gran número de dispositivos, usuarios y datos.

Característica XI. Escalabilidad

A continuación, presento el análisis esquematizado y detallado basado en los datos de la tabla sobre la escalabilidad de diferentes plataformas IoT:

Escalabilidad masiva: Algunas plataformas, como AWS IoT Core, Google Cloud IoT Core y Microsoft Azure IoT Central & Hub, destacan por su capacidad para soportar millones de dispositivos y gestionar miles de millones de mensajes al día. Esto es crucial para las grandes corporaciones y proyectos IoT a gran escala. Por ejemplo, una empresa de transporte internacional que utiliza IoT para realizar un seguimiento de su flota global de vehículos necesitaría una plataforma como AWS IoT Core.

Escalabilidad horizontal y vertical: Varias plataformas, incluyendo IBM Watson IoT, OpenRemote y Sofia2, proporcionan escalabilidad tanto horizontal como vertical. Esto significa que estas plataformas pueden manejar un aumento en el número de dispositivos (escalabilidad horizontal) y también pueden aumentar su capacidad para manejar más datos por dispositivo (escalabilidad vertical).

Escalado automático: Plataformas como Carriots, M. Azure IoT Hub y IBM Watson IoT pueden aumentar automáticamente su capacidad para manejar más dispositivos o datos cuando sea necesario. Esto es útil para las empresas que experimentan fluctuaciones en la demanda de sus servicios IoT.

Microservicios y Kubernetes: Algunas plataformas como DeviceHive y Kaa utilizan una arquitectura basada en microservicios y son orquestadas por Kubernetes, lo que les permite escalar prácticamente de manera ilimitada.

Capacidad de agrupación: Plataformas como Macchina.IO EDGE permiten agrupar varios servidores para aumentar la capacidad y soportar más dispositivos. Esto es útil para las empresas que necesitan gestionar enorme cantidad de datos, agrupando servidores para manejar millones de dispositivos.

En general, todas las plataformas listadas ofrecen alguna forma de escalabilidad, lo cual es un aspecto clave para cualquier solución IoT, ya que el número de dispositivos y la cantidad de datos generados puede aumentar significativamente con el tiempo.

5.2. Interoperabilidad:

Plataforma	Hardware Compatible
Altair SmartWorks IoT	Admite dispositivos de hardware: sensores, actuadores, *gateways* y dispositivos de periferia (*edge*).
AWS IoT Core	Admite diversos dispositivos de hardware, como Raspberry Pi, Arduino, Intel Edison, dispositivos Amazon Sidewalk, sensores, dispositivos actuadores y gateways.
Carriots	Compatible con Raspberry Pi, Arduino y ESP8266.
DeviceHive	Amplia gama de hardware IoT, incluidos Arduino, ESP8266, Raspberry Pi, BeagleBone, sensores, actuadores y gateways.
Google Cloud IoT Core	Compatible con una amplia gama de dispositivos y plataformas de hardware, como Raspberry Pi, Arduino, Intel Edison, Android Things, dispositivos con firmware Cloud IoT Core, dispositivos de terceros con conectividad compatible, sensores, actuadores, *gateways* y otros.
IBM Watson IoT	Admite una amplia gama de módulos hardware, incluidos sensores, actuadores, *gateways*, dispositivos de periferia (*edge*) y cualquier dispositivo que pueda conectarse a Internet incluyéndose también, hardware de terceros.
IoTSens	Admite una amplia gama de dispositivos, incluidos sensores, actuadores y *gateways*.
Kaa	Compatible con una amplia gama de dispositivos de hardware como sensores, actuadores y *gateways*. Ej. de algunos de ellos: Raspberry Pi, Arduino o BeagleBone Black entre otros.
Macchina.IO EDGE	Admite una amplia gama de hardware, como: Raspberry Pi, BeagleBone, Intel Edison y otros dispositivos basados en Linux.
Mainflux	Admite una gran variedad de hardware IoT, incluidos sensores, actuadores y *gateways* de entre los que destacan: Raspberry Pi, Arduino, Intel Edison, AWS IoT Greengrass.
M. Azure IoT Central	Admite diversos dispositivos de hardware, como sensores, actuadores y *gateways*.
M. Azure IoT Hub	Compatible con hardware tales como sensores, actuadores, gateways, Raspberry Pi, Arduino, Intel Edison, Azure Stack, Azure IoT Edge. Asimismo, se puede integrar con hardware de terceros.
OpenRemote	Admite diversos dispositivos de hardware, como sensores, actuadores, gateways, además de dispositivos hardware personalizados.
Predix Platform	Compatible con una amplia gama de dispositivos hardware, como sensores, actuadores y gateways, y admite diferentes módulos hardware, como Raspberry Pi, Intel Edison y Arduino entre otros.
Sentilo	Compatible modelos hardware como Raspberry Pi, Arduino, Intel Edison, y dispositivos *gateways* en general.
Sofia2	Soporta una amplia gama de dispositivos IoT, tales como Raspberry Pi, Arduino o Intel Edison y numerosos sensores y actuadores del mercado.
Thinger.io	Compatible con Arduino, Raspberry Pi y dispositivos BLE, ESP8266.
ThingSpeak	Arduino, Raspberry Pi, BeagleBone Black, así como diversos sensores y actuadores.
Zetta	Compatible con Raspberry Pi, BeagleBone Black, Intel Edison y Arduino entre otros.

Característica XII. Hardware Compatible

Resumen conciso de los factores clave detectados en términos de interoperabilidad con diferentes dispositivos hardware en las plataformas IoT:

Compatibilidad generalizada con dispositivos: Casi todas las plataformas listadas admiten una amplia gama de dispositivos IoT, lo que incluye sensores, actuadores, gateways y dispositivos de periferia (edge) entre otros. Este punto se ve evidenciado en la compatibilidad de IBM Watson IoT, que admite los módulos hardware mencionados previamente y cualquier dispositivo que pueda conectarse a Internet, incluyendo hardware de terceros.

Soporte de dispositivos hardware comunes: Muchas plataformas son compatibles con hardware IoT común y específico, generalmente de terceros, como Raspberry Pi, Arduino, Intel Edison y BeagleBone Black. Por ejemplo, AWS IoT Core admite Raspberry Pi, Arduino, Intel Edison, dispositivos Amazon Sidewalk, entre otros.

Soporte para dispositivos propios o personalizados: Algunas plataformas, como OpenRemote, también admiten dispositivos de hardware personalizados, lo que ofrece aún más flexibilidad.

Plataforma	Dominios o Casos de Uso Respaldados															
	Ciudades Inteligentes	Hogares Inteligentes/Domótica	Vehículos Inteligentes	Edificios inteligentes	Automatización Industrial	Sanidad	Energía	Logística	Fabricación	Comercio	Transporte/Distribución	Seguimiento de Activos	Agricultura	Seguridad	Gestión de Flotas	Monitorización Medioambiental
Altair SmartWorks IoT	✓				✓	✓	✓	✓	✓	✓	✓					
AWS IoT Core	✓		✓		✓	✓		✓	✓	✓	✓	✓		✓		
Carriots	✓	✓			✓	✓		✓	✓		✓	✓	✓			
DeviceHive	✓	✓			✓	✓		✓			✓	✓	✓	✓		
Google Cloud IoT Core	✓		✓		✓	✓					✓	✓	✓			
IBM Watson IoT	✓					✓	✓		✓		✓					
IoTSens	✓	✓			✓	✓										
Kaa	✓				✓	✓		✓			✓	✓			✓	
Macchina.IO EDGE	✓				✓	✓		✓	✓		✓					
Mainflux	✓				✓	✓	✓	✓	✓		✓		✓			
M. Azure IoT Central	✓				✓	✓	✓	✓	✓		✓	✓				
M. Azure IoT Hub	✓				✓	✓			✓		✓					
OpenRemote	✓				✓	✓	✓		✓		✓					
Predix Platform						✓	✓		✓		✓					

Plataforma	Dominios o Casos de Uso Respaldados															
	Ciudades Inteligentes	Hogares Inteligentes/Domótica	Vehículos Inteligentes	Edificios inteligentes	Automatización Industrial	Sanidad	Energía	Logística	Fabricación	Comercio	Transporte/Distribución	Seguimiento de Activos	Agricultura	Seguridad	Gestión de Flotas	Monitorización Medioambiental
Sentilo	✓			✓	✓	✓		✓			✓					
Sofia2	✓				✓	✓		✓	✓		✓					
Thinger.io					✓	✓			✓							✓
ThingSpeak	✓				✓	✓	✓	✓	✓		✓		✓			✓
Zetta	✓	✓			✓	✓							✓			

Característica XIII. Dominios o Casos de Uso Respaldados

Análisis de la tabla función del número de casos de uso que cada plataforma abarca:

Abarcan la mayor cantidad de casos de uso: Estas son las plataformas que cubren una gran variedad de dominios o casos de uso, es decir, 9 o más.

- AWS IoT Core, Carriots, DeviceHive y ThingSpeak.

Abarcan aproximadamente la mitad de los casos de uso: Estas plataformas respaldan aproximadamente la mitad de los dominios o casos de uso, es decir, 5 a 8.

- Altair SmartWorks IoT, Google Cloud IoT Core, IBM Watson IoT, Kaa, Macchina.IO EDGE, Mainflux, M. Azure IoT Central & Hub, OpenRemote, Sentilo, Sofia2 y Zetta.

Abarcan menos de la mitad de los casos de uso: Estas son las plataformas que cubren menos dominios o casos de uso, es decir, 4 o menos.

- IoTSens, Predix Platform y Thinger.io

6. Requisitos No Funcionales:

Plataforma	Requisitos No Funcionales (Ej.: Disponibilidad, Rendimiento, Copia de seguridad de datos, etc...)
Altair SmartWorks IoT	Tiene un Acuerdo de Nivel de Servicio (SLA) de alta disponibilidad del 99,9%. Diseñado para alta disponibilidad y rendimiento, soportando aplicaciones de misión crítica.

	Proporciona funciones de copia de seguridad de datos, recuperación, equilibrio de carga y almacenamiento en caché, con copias de seguridad automáticas a través de AnythingDB.
AWS IoT Core	Proporciona un servicio de alta disponibilidad con un SLA de tiempo de actividad del 99,9% y ofrece diversas características de desempeño como colas de mensajes y equilibrio de carga, y opciones de backup de datos, como Amazon S3 y Amazon Glacier.
Carriots	SLA de alta disponibilidad del 99,9%. Puede gestionar hasta 1 millón de mensajes por segundo. Funciones: copia de seguridad diaria de los datos, recuperación en caso de catástrofe y replicación de datos.
DeviceHive	SLA de alta disponibilidad del 99,9%. Admite funciones de alto rendimiento como el equilibrio de carga y el almacenamiento en caché. Copia de seguridad de datos en varias localizaciones de almacenamiento: Amazon S3, Google Cloud Storage, Microsoft Azure, disco local y almacenamiento en la nube. Se realizan copias de seguridad de los datos con regularidad y se pueden restaurar en caso de desastre.
Google Cloud IoT Core	Garantiza un SLA de alta disponibilidad, fiabilidad y rendimiento para despliegues IoT. Aprovecha la infraestructura global de Google para proporcionar una plataforma escalable y robusta. Puede gestionar millones de mensajes por segundo y proporciona un alto nivel de rendimiento. Se realizan copias de seguridad diarias de los datos, que pueden restaurarse en cualquier momento. Otras funciones son el equilibrio de carga y la geo-replicación.
IBM Watson IoT	Ofrece alta disponibilidad, rendimiento y copia de seguridad de datos con un *SLA* de tiempo de actividad del 99,9%. Dispone de funciones de almacenamiento en caché y balanceo de carga. Las copias de seguridad pueden realizarse de manera periódica, guardando las mismas en varios proveedores de almacenamiento en la nube.
IoTSens	Garantiza alta disponibilidad, funciones de rendimiento como balanceo de carga y almacenamiento en caché, y servicios de copia de seguridad y recuperación de datos.
Kaa	Garantiza un acuerdo de nivel de servicio (*SLA*) de alta disponibilidad, baja latencia y un tiempo de actividad del 99,9%. También ofrece copia de seguridad de datos.
Macchina.IO EDGE	Proporciona alta disponibilidad, con un SLA de tiempo de actividad del 99,9%. Puede gestionar grandes volúmenes de datos y realizar copias de seguridad. Se puede desplegar en un clúster para garantizar una alta disponibilidad.
Mainflux	Ofrece alta disponibilidad y rendimiento con un SLA del 99,9%, 10.000 mensajes por segundo y copias de seguridad diarias.
M. Azure IoT Central	Ofrece un SLA de alta disponibilidad y alto rendimiento del 99,9%, balanceo de carga, almacenamiento en caché y permite copias de seguridad de datos en Azure Blob Storage cada 15 minutos.
M. Azure IoT Hub	Azure IoT Hub cumple requisitos no funcionales como alta disponibilidad (con un SLA del 99,9%), rendimiento, balanceo de carga, almacenamiento en caché, copia de seguridad de datos en Azure Blob Storage y otros proveedores de almacenamiento en la nube, georreplicación y recuperación ante desastres. La plataforma puede escalarse para satisfacer las necesidades de aplicaciones exigentes, y se realizan copias de seguridad de los datos de forma automática y periódica para garantizar la disponibilidad en caso de fallo.
OpenRemote	SLA de alta disponibilidad del 99,9% con un rendimiento máximo de 1 millón de mensajes por segundo. Ofrece opciones de backup de datos, con copias de seguridad diarias en Amazon S3.
Predix Platform	Diseñado para ofrecer una alta disponibilidad, con un SLA del 99,9%. Ofrece funciones que garantizan un alto rendimiento, una gran fiabilidad, copias de seguridad y cifrado de datos.
Sentilo	Ofrece una alta disponibilidad y rendimiento con un SLA del 99,9% de tiempo de operatividad. A su vez, también cuenta con: manejo de grandes volúmenes de datos y millones de mensajes por segundo y opciones de copia de seguridad (MongoDB, Cassandra, Elasticsearch) para proteger los datos en caso de fallo del sistema.
Sofia2	Garantiza una alta disponibilidad (SLA del 99,9%), alto rendimiento y copia de seguridad de datos tanto en la nube como en las instalaciones propias del cliente.
Thinger.io	Garantiza una alta disponibilidad (SLA del 99,9% de tiempo operativo) y proporciona funciones de alto rendimiento como balanceo de carga y almacenamiento en caché, y varios servicios de copia de seguridad de datos y recuperación en caso de catástrofe.
ThingSpeak	Garantiza una alta disponibilidad (SLA del 99,9%). También permite gestionar grandes volúmenes de datos y ofrece servicios diarios de copia de seguridad y recuperación de datos.

Zetta	Garantiza un SLA de alta disponibilidad. Proporciona almacenamiento en caché con balanceo de carga y diversas opciones de copia de seguridad de datos en diferentes ubicaciones (servidores locales, almacenamiento en la nube y dispositivos periféricos).

Característica XIV. Requisitos No Funcionales

Tras revisar estos detalles, se puede observar que todas las plataformas ofrecen un alto nivel de disponibilidad y funcionalidades para la copia de seguridad de datos, lo que demuestra la importancia de estos factores en el campo del IoT. Sin embargo, las funcionalidades específicas de balanceo de carga, almacenamiento en caché y capacidad para manejar altos niveles de tráfico pueden variar de una plataforma a otra.

1. Alta Disponibilidad: Casi todas las plataformas garantizan una alta disponibilidad, reflejada en un Acuerdo de Nivel de Servicio (SLA) del 99,9%. Esto significa que se espera que estas plataformas estén operativas el 99,9% del tiempo, minimizando las interrupciones de servicio.

2.Copia de seguridad de datos: Todas las plataformas ofrecen alguna forma de copia de seguridad de datos, lo que es esencial para prevenir la pérdida de datos en caso de fallo del sistema. Algunas plataformas también proporcionan la capacidad de restaurar los datos en caso de desastre y/o la opción de geo-replicación de los datos, que permite almacenar copias en diferentes ubicaciones geográficas

3. Balanceo de carga y almacenamiento en caché: Muchas de las plataformas ofrecen funciones de balanceo de carga y almacenamiento en caché, lo que puede mejorar el rendimiento y la escalabilidad al distribuir la carga de trabajo entre varios recursos y almacenar temporalmente datos de uso frecuente para un acceso más rápido.

4. Rendimiento: Algunas de las plataformas pueden gestionar grandes volúmenes de datos y un alto tráfico de mensajes por segundo. Esto es crucial para las aplicaciones de IoT, que a menudo implican la transmisión de grandes cantidades de datos en tiempo real.

Al analizar esta información, es evidente que estas plataformas se centran en ofrecer un servicio robusto y confiable que pueda soportar las demandas de las aplicaciones de IoT. La elección de una plataforma específica dependerá en gran medida de las necesidades de cada proyecto y de sus requisitos técnicos. Para proyectos que requieran una alta capacidad de procesamiento de datos y tráfico, plataformas como Google Cloud IoT Core, Carriots o Mainflux pueden ser más adecuadas. Para proyectos donde la restauración de datos y la recuperación de desastres sean factores críticos, plataformas como DeviceHive, IBM Watson IoT o Azure IoT Hub podrían ser más apropiadas.

En la sección **10. Apéndice** se podrá encontrar la información mostrada en este punto, en vez de siendo clasificada por *Características*, organizada por *Plataformas*.

6. Estudio de las plataformas IoT para dominios específicos

6.1. Plataformas IoT en la nube: Usuario vs. Industrial (IIoT)

El Internet de las Cosas (IoT) se ha convertido en una fuerza transformadora en varios sectores, lo que ha dado lugar a avances tanto en la comodidad del consumidor como en la eficiencia industrial. Estos avances están impulsados por plataformas de IoT en la nube, que pueden clasificarse a grandes rasgos en plataformas IoT a nivel Usuario (de consumo) y plataformas IoT a nivel Industrial (IIoT). Este ensayo ofrece una comparación exhaustiva entre estos dos tipos de plataformas en cuanto a su público objetivo, enfoque y casos de uso específicos. Además, profundiza en la creciente importancia de la IIoT en los entornos empresariales contemporáneos y explora el papel fundamental de los enfoques híbridos borde-nube en la implementación efectiva de la IIoT.

Las plataformas IoT de consumo se dirigen principalmente a usuarios finales individuales y se centran en aplicaciones que proporcionan comodidad en el día a día, bienestar personal y automatización inteligente del hogar. Aparatos de seguimiento de la actividad física como medidores de pasos, dispositivos de control de la salud como glucómetros, y dispositivos domésticos inteligentes como termostatos y sistemas de iluminación, son ejemplos de los casos de uso de las plataformas IoT de consumo. Estas hacen hincapié en interfaces fáciles de usar y en una integración transparente, facilitando el uso a usuarios sin conocimientos técnicos. Además, dan prioridad a aspectos como la privacidad de los datos, la experiencia del usuario y la compatibilidad entre dispositivos.

Por otro lado, las plataformas IoT a nivel Industrial (IIoT) se dirigen a ingenieros, profesionales industriales y empresas que gestionan y mantienen sistemas industriales complejos. La eficiencia operativa, el mantenimiento predictivo y la optimización de procesos y otros entornos industriales forman el núcleo del enfoque de IIoT. Están diseñadas para recopilar y manejar grandes volúmenes de datos con el fin de mejorar el rendimiento, predecir posibles fallos y facilitar una mejor toma de decisiones, gestionar intrincados procesos industriales y satisfacer los requisitos específicos de diversos sectores, como la energía, el transporte y la fabricación.

En los últimos años, se ha producido un repunte en la adopción de plataformas IIoT por parte de las empresas industriales, un cambio impulsado por un movimiento más allá del paso de la experimentación tecnológica a la persecución de objetivos empresariales basados en IoT. Como resultado de la pandemia COVID-19, se ha producido un aumento de las operaciones que minimizan la participación humana, impulsando en consecuencia la adopción de plataformas

IIoT. Estas empresas se centran ahora fundamentalmente en el ahorro de costes, el aumento de la producción, la automatización, la optimización de los procesos industriales, los requisitos de sostenibilidad y la seguridad de los empleados.

A la luz de estos cambios, las empresas industriales están adaptando sus iniciativas de negocio para centrarse en la automatización, la monitorización remota y la sostenibilidad, lo que demuestra una evolución significativa en el mercado de plataformas IIoT. Las empresas también se están aventurando en nuevas iniciativas de ingresos como los productos inteligentes y oportunidades de negocio innovadoras, como los "productos como servicio", acelerando de forma efectiva la transformación de sus modelos de negocio.

Una consideración clave para adoptar con éxito las plataformas IIoT es elegir un modelo de implantación eficaz. Según un estudio de Gartner Inc. en el que participaron aproximadamente 1900 empresas para el “Cuadrante Mágico de plataformas IIoT globales” [10], el enfoque híbrido borde-nube emerge como el modelo de despliegue más común, con un 40% de los proyectos, en gran medida, debido a que un enfoque híbrido otorga una gran flexibilidad ya que permite una mezcla de procesamiento de borde (*edge*) y procesamiento basado en la nube, proporcionando un equilibrio de latencia, seguridad de datos y potencia de procesamiento. A este enfoque le siguen de cerca los despliegues solo en la nube, con un 36 %, mientras que los despliegues solo en el borde representan el 24 % restante.

En conclusión, las diferencias fundamentales entre las plataformas IoT de consumo y las IIoT residen en su público objetivo, su enfoque y sus casos de uso. Mientras que las plataformas IoT de consumo están orientadas a mejorar el bienestar personal y la comodidad de los usuarios finales, las plataformas IIoT se dirigen a los profesionales de la industria, haciendo hincapié en la eficiencia operativa, el mantenimiento predictivo y la optimización de procesos. Con el cambio cada vez mayor hacia objetivos empresariales basados en IoT y la creciente preferencia por modelos de despliegue híbridos borde-nube, las plataformas IIoT están llamadas a desempeñar un papel cada vez más vital en el futuro de las operaciones industriales.

6.2. Características clave que debe poseer una plataforma en la nube IIoT

El IIoT ha presagiado una nueva era de automatización industrial, mantenimiento predictivo y fábricas inteligentes. La clave de estos avances es la selección e implementación de la plataforma en la nube IIoT adecuada, que sirve de columna vertebral para todo el ecosistema IIoT. Ésta proporciona un marco unificado en el que los datos se recopilan, procesan, analizan y explotan para impulsar la toma de decisiones informadas y las perspectivas empresariales. Sin embargo, para garantizar el éxito de la implantación de un proyecto de IIoT es necesario considerar detenidamente varios aspectos cruciales a la hora de seleccionar la plataforma en la nube IIoT adecuada. A continuación, se ofrece una explicación más detallada de aquellos aspectos que consideramos de mayor relevancia en el ámbito industrial:

1. **Integración e Ingesta de datos**

La ingesta y la integración de datos son procesos críticos en el ecosistema IIoT. La ingesta de datos se refiere a la recopilación, el procesamiento y el almacenamiento de la gran cantidad de datos diversos generados por los dispositivos IoT. Este proceso requiere compatibilidad con diversos protocolos de comunicación y tipos de datos (estructurados, no estructurados y semiestructurados), así como la capacidad de gestionar la velocidad y el volumen de generación de datos para garantizar un procesamiento oportuno y eficiente.

Por otro lado, la integración en IoT implica combinar software, herramientas y tecnologías para garantizar una interacción sin fisuras entre la plataforma IoT y otros sistemas empresariales, incluidos los despliegues en la nube y en las instalaciones. Esto se facilita mediante API, middleware y conectores preconstruidos, que permiten una comunicación y un intercambio de datos eficientes.

En esencia, una ingestión de datos eficiente sienta las bases para un análisis de datos eficaz en IoT, mientras que la integración garantiza un funcionamiento fluido dentro de sistemas empresariales más amplios. El equilibrio entre estos dos procesos es esencial para la solidez y eficiencia de cualquier sistema IoT.

2. **Análisis de datos**

El análisis de datos es fundamental en la IIoT, ya que transforma los datos brutos de los sistemas conectados en información procesable mediante la supervisión en tiempo real, el análisis predictivo, los algoritmos de machine learning, el procesamiento de flujos de eventos, los motores de reglas y la visualización de datos. Este proceso de extracción permite diversas aplicaciones útiles, como la supervisión del uso de activos, la anticipación de las necesidades de mantenimiento, el seguimiento de patrones y la optimización de la utilización de recursos.

El valor intrínseco del IIoT se manifiesta en estos conocimientos basados en datos, proporcionando una ventaja significativa en la toma de decisiones estratégicas, la predicción de resultados futuros y el descubrimiento de información previamente oculta. En consecuencia, las organizaciones que operan en este ámbito reconocen el carácter indispensable de unas sólidas capacidades de análisis de datos para mantener una ventaja competitiva y garantizar la capacidad de respuesta.

3. Visualización de datos y Accesibilidad

La utilización óptima de los datos obtenidos por los sistemas IIoT, depende en gran medida de la estrategia de visualización implementada y de la intuitividad de las interfaces que ofrezca la plataforma. Cuadros de mando personalizables permiten una interpretación significativa de datos complejos, contribuyendo a una toma de decisiones informada.

Adicionalmente, la accesibilidad a usuarios con distintos conocimientos técnicos se garantiza mediante una interfaz fácil de usar. Dicha interfaz suele incluir funciones personalizables como cuadros de mandos, diagramas y gráficos que facilitan la interpretación y visualización de los datos, haciendo que la información sea más práctica. Además, es recomendable escoger aquellas plataformas que proporcionen APIs completas y documentación clara para una integración sin problemas con otros sistemas.

4. Seguridad e Integridad de los datos

Dado que los sistemas IIoT suelen manejar datos sensibles y de misión crítica, no se puede subestimar el riesgo que plantean las posibles ciberamenazas. Utilizar un servicio en la nube de confianza con un equipo de seguridad sólido es una forma viable de garantizar la seguridad de los datos. Los grandes proveedores de servicios en la nube pueden permitirse contratar a expertos en ciberseguridad de primer nivel, lo que se traduce en una mayor seguridad de los datos. Sin embargo, la responsabilidad de la seguridad de los datos va más allá del proveedor de la nube; los ingenieros de IIoT deben incorporar medidas de seguridad en sus diseños, creando un enfoque de seguridad multicapa que proteja el ecosistema de la IIoT de las ciberamenazas en evolución. Esta estrategia de defensa en capas, conocida como "defensa en profundidad", puede mitigar significativamente el riesgo de éxito de un ciberataque. Es fundamental integrar la seguridad en cada capa de la pila IoT, empleando protocolos de seguridad establecidos como autenticación multifactor y control de acceso basado en roles (*RBAC*).

Por otra parte, garantizar la integridad de los datos es de vital importancia en el ecosistema IIoT. La integridad de los datos asegura que los datos almacenados sigan siendo precisos y coherentes durante todo su ciclo de vida, garantizando así la fiabilidad de los datos para los procesos de toma de decisiones.

5. Gestión de dispositivos

La gestión eficaz de los dispositivos es un componente crítico del IIoT y constituye la columna vertebral de cualquier sistema IoT. Las plataformas IoT deben ofrecer sólidas funciones de gestión de dispositivos, que permitan una incorporación fluida de dispositivos con metadatos enriquecidos y una agrupación eficiente basada en características compartidas. Estas plataformas deben proporcionar estrictas medidas de seguridad, incluida la autenticación y autorización de dispositivos específicos, así como una fácil aplicación de políticas de listas negras y blancas.

Asimismo, deben facilitar la gestión remota, permitiendo tareas como actualizaciones de firmware y depuración (*debugging*) en ubicaciones remotas. Esta función de gestión de dispositivos debe incluir software que permita realizar tareas tanto manuales como automatizadas para gestionar dispositivos IoT de forma remota, ya sea en bloque o individualmente, garantizando un enfoque integral y eficiente en la gestión de dispositivos. Por otra parte, las plataformas IoT deben ser capaces de integrarse con las soluciones MRP tradicionales (*Material Requierement Planning*, o Planificación de Requerimientos de Materiales), mejorando la asignación de recursos.

En esencia, la gestión de dispositivos es fundamental para la eficiencia, la seguridad y el rendimiento general de los sistemas IoT, y los futuros avances en este campo marcarán la trayectoria futura del área del IIoT.

6. Habilitación y Gestión de aplicaciones

La habilitación y la gestión de aplicaciones (*Application Enablement and Management* (AEM)) es fundamental para el panorama del IIoT ya que proporciona el software y la infraestructura que permite a las aplicaciones empresariales de cualquier modelo de implantación analizar datos y llevar a cabo funciones empresariales relacionadas con IoT. AEM consta de componentes de software básicos que gestionan el sistema operativo, la entrada y salida estándar o los sistemas de archivos, mejorando así la eficiencia y funcionalidad del entorno IIoT.

AEM soporta el desarrollo de aplicaciones y la gestión del tiempo de ejecución, ofreciendo a las empresas la capacidad de crear, gestionar y desplegar aplicaciones de forma rápida y eficaz. Esto es crucial para mantener un flujo ininterrumpido de datos y operaciones en una estructura IIoT. Otro aspecto clave es que AEM garantiza la escalabilidad y fiabilidad a escala de la nube, lo que permite a las empresas gestionar cargas de trabajo y volúmenes de datos cada vez mayores sin comprometer el rendimiento. También facilita el rápido despliegue y entrega de soluciones IoT, lo que lo convierte en un componente indispensable en el vertiginoso entorno digital actual.

Por último, AEM proporciona plantillas e instancias de gemelos digitales e hilos digitales, lo que facilita una mejor simulación, testeo y optimización de los sistemas IoT. Este enfoque predictivo y preventivo es especialmente importante en el panorama de la IIoT, donde las interrupciones pueden acarrear implicaciones significativas.

7. Escalabilidad

La escalabilidad en el panorama del IIoT es una necesidad crítica. A medida que evolucionan los ecosistemas IIoT, el número de dispositivos y datos puede aumentar considerablemente, lo que acentúa la necesidad de escalabilidad. Una plataforma en la nube IIoT ideal debe gestionar un volumen creciente de datos y dispositivos sin problemas y sin comprometer el rendimiento o la disponibilidad.

Por lo tanto, la selección de una plataforma que pueda escalar eficientemente hacia arriba (o hacia abajo) de acuerdo con los cambios en los requisitos del negocio es una necesidad absoluta. Esta flexibilidad permite a las organizaciones ajustar la escala de su sistema garantizando que el proyecto pueda crecer orgánicamente y adaptarse a los cambiantes requisitos empresariales, evitando el exceso de aprovisionamiento de recursos y reduciendo costes innecesarios, asegurando la viabilidad a largo plazo.

8. Interoperabilidad

La interoperabilidad es esencial en el panorama del Internet Industrial de las Cosas (IIoT), caracterizado por la diversidad de dispositivos, sistemas y protocolos. Una plataforma IIoT eficaz debe admitir múltiples protocolos e interfaces de comunicación para facilitar la integración y asegurar una comunicación eficiente.

Una plataforma IIoT puede ser agnóstica respecto a la nube, lo que deriva en que estas pueden integrar diferentes servicios en la nube, evitando así la dependencia de un solo proveedor y garantizando la diversidad del sistema. Esta flexibilidad permite a las organizaciones elegir los mejores dispositivos y sensores para sus necesidades, independientemente del protocolo o la interfaz utilizada, lo que promueve un ecosistema IIoT diverso y robusto.

Las plataformas deben estar diseñadas para soportar la interoperabilidad desde el principio, permitiendo que los dispositivos IoT se conecten con una amplia gama de otros productos y servicios. Esto conduce a una mayor eficiencia operativa, reducción de costes y mejores resultados empresariales. El éxito futuro de la IIoT dependerá de plataformas que puedan adaptarse a las nuevas tecnologías y estándares, permitiendo una adaptabilidad continua dentro del ecosistema de IoT en constante evolución.

En resumen, el éxito de la implementación de una plataforma en la nube IIoT gira en torno a la cuidadosa consideración de la seguridad e integridad de los datos, la gestión eficiente de grandes volúmenes de datos, la extracción de información valiosa a través del análisis de datos, la visualización eficaz de los datos y la interoperabilidad entre los dispositivos y plataformas conectados Un conocimiento profundo de estos aspectos garantiza el éxito de la implantación de la IIoT, que puede evolucionar con las necesidades empresariales y resistir los retos de un panorama tecnológico en constante cambio, lo que en última instancia conduce a una mayor eficiencia operativa, reducción de costes y mejores resultados empresariales.

6.3. ¿Quién lidera actualmente el entorno IIoT en el mercado?

Para responder a esta pregunta, nos basaremos en el estudio realizado por Gartner Inc. "empresa líder mundial en investigación, asesoramiento y consultoría de tecnología, que busca brindar ideas, consejos y herramientas a empresas, profesionales TIC y comunidad en general, sobre las nuevas tendencias de tecnologías de la información en el mercado" [32]

Gartner realiza una exhaustiva investigación y análisis específicos para las diferentes líneas de servicios en TI dando como resultado un completo informe, acompañado del llamado "Cuadrante Mágico de Gartner". Este cuadrante es una herramienta que mide el nivel de innovación y desarrollo de las empresas de tecnología a nivel mundial el cual se convirtió en una referencia para las empresas en el momento de tomar decisiones en los procesos de transformación digital.

Los resultados se muestran a través de un gráfico de dos ejes y 4 zonas: *Challengers* (Aspirantes), *Niche Players* (Jugadores de Nicho), *Visionaires* (Visionarios) y *Leaders* (Lideres); considerándose Líderes aquellos que destacan por su comprensión visionaria de hacia dónde se dirige el mercado y su capacidad para adaptar sus capacidades para satisfacer las necesidades específicas de la industria, tanto a nivel local como global.

En el ámbito del Internet Industrial de las Cosas (IIoT), los Líderes no solo ofrecen soluciones integrales que se adhieren a los protocolos y normativas industriales necesarios, sino que también aportan valor añadido a sus plataformas mediante la integración de aplicaciones y servicios adicionales. Como resultado, estas plataformas pueden funcionar sin problemas con una diversa gama de activos y sistemas industriales. A su vez, los Líderes garantizan la flexibilidad y accesibilidad de sus soluciones, permitiendo su desarrollo a medida y colaborando con otras entidades tecnológicas para asegurar su integración con otros sistemas. Generan confianza en el mercado a través de la presentación de casos de éxito y se comprometen con la entrega de soluciones integrales para satisfacer las necesidades únicas de sus clientes, manteniéndose a la vanguardia con innovación y perfeccionamiento continuo.

Una vez hemos explicado esto, y observando la Ilustración 46, daremos una breve explicación de por qué es considerada Microsoft Azure IoT como Líder en el mercado global de plataformas IoT Industriales en la más reciente investigación realizada por Gartner Inc. en este ámbito [10] (Publicada el 12 de diciembre de 2022).

[32] https://www.gartner.com/en.

Ilustración 46. Cuadrante Mágico de Plataformas Globales de IoT Industrial

A modo de resumen del análisis particular que realiza Gartner, Microsoft es reconocido como líder en el Cuadrante Mágico con su plataforma Azure IoT, la cual ofrece un amplio espectro de opciones de despliegue en la nube, en el perímetro e híbridas para cubrir casi todos los casos de uso e industrias. No obstante, determinar las combinaciones de productos adecuadas y entender los precios puede resultar complicado. Microsoft utiliza un sólido ecosistema de socios, que abarca integradores y socios tecnológicos, para ampliar su alcance a través de diferentes mercados. En los últimos años, varios actores importantes de la IIoT, como PTC, GE, Siemens, ABB y Schneider (algunos de ellos pertenecientes a los grupos Visionarios o Aspirantes a Lideres), han anunciado públicamente que sus plataformas se han construido sobre la nube de Microsoft.

Entre los puntos fuertes de Microsoft se encuentran su vasto ecosistema de socios, su profundo conocimiento de perfiles de seguridad complejos y su experiencia en capacidades relacionadas, como la integración de datos, el análisis avanzado y la visualización, entre otras.

Sin embargo, existen ciertos retos. La complejidad y amplitud de la oferta de productos de Microsoft puede confundir a los clientes y retrasar la finalización de los proyectos, al mismo tiempo que puede aumentar su coste. La intrincada estructura de precios requiere una evaluación cuidadosa por parte de las empresas para garantizar que la plataforma IIoT se alinea con sus objetivos de negocio. Además, las experiencias de los clientes pueden variar entre regiones, sectores y socios, por lo que es esencial que las empresas establezcan objetivos de proyecto claros y resultados medibles.

En resumidas cuentas, los 5 motivos por los que Microsoft Azure es la mejor opción para implementar soluciones IIoT, y que en consecuencia la convierten en un Líder son:

1. Azure IoT Central como plataforma SaaS: Azure IoT Central es la principal plataforma SaaS de Microsoft para soluciones IoT, proporcionando una base sólida para aplicaciones IIoT.

2. Ecosistema de socios diverso y rico: Como se ha mencionado anteriormente, varios actores importantes de IIoT como PTC, GE, Siemens, ABB y Schneider han anunciado que sus plataformas están construidas en la nube de Microsoft, lo que la realza como ecosistema fuertemente establecido.

3. Énfasis en la seguridad: Microsoft ha realizado importantes inversiones en servicios de seguridad, como Azure Defender for IoT (para la seguridad de puntos finales), Azure Sphere (una unidad de microcontrolador que garantiza la integración segura de borde a nube y de nube a borde) y Azure RTOS (un sistema operativo en tiempo real para IoT y dispositivos de borde). Estos servicios abordan los complejos retos de seguridad a los que a menudo se enfrentan las empresas.

4. Software IoT Edge compatible con DevOps: Azure IoT Edge, un software de código abierto de Microsoft, actúa como puente entre los dispositivos locales y la nube pública, mejorando la productividad de desarrolladores y operadores. Componentes esenciales como Azure Stream Analytics y Azure SQL Database se han migrado al borde, proporcionando procesamiento de flujos y almacenamiento en el borde.

5. Estrecha integración con plataformas de datos, análisis y *machine learning* (ML): Azure ofrece un amplio conjunto de servicios de datos y análisis que están estrechamente integrados con la plataforma Azure IoT. Ofrece la posibilidad de recibir datos telemétricos de sensores, almacenarlos en Azure Cosmos DB o Azure Data Lake y utilizarlos para entrenar modelos de aprendizaje automático en Azure ML. Servicios como Azure Event Bus, Stream Analytics, Data Lake y Azure ML ofrecen análisis completos *end-to-end*.

No obstante, me gustaría concluir recalcando que escoger al Líder no es siempre la mejor opción, sino que se ha de escoger aquella que en el caso de uso en el que nos encontremos, sea capaz de ser la mejor que satisfaga las necesidades particulares que se requieran. Un proveedor más pequeño y centrado puede ofrecer un excelente apoyo y compromiso que se adapte mejor a las necesidades individuales. Además, algunos proveedores pueden proporcionar capacidades específicas, como una mayor seguridad o experiencia en un submercado particular, que podrían ser cruciales para nuestra organización.

7. Aplicación a casos prácticos

7.1. Diseño de escenarios ilustrativos: Un enfoque práctico para la selección óptima de plataformas en contextos IIoT reales

En esta fase del proyecto, procederemos a elaborar una serie de escenarios ilustrativos, cuidadosamente seleccionados, que representarán diversas situaciones o contextos en las que, basándonos en los datos y análisis realizados previamente, argumentaremos en función de los requerimientos específicos de cada situación cuales son las plataformas que mejor se adaptan para conseguir abarcar todas sus necesidades. De esta manera, esta fase proporcionará un marco práctico y contextual para comprender mejor la aplicabilidad y utilidad de las diferentes características en situaciones reales.

Recordatorio de todas las características estudiadas en el Pto 5.2:

- Certificaciones y cumplimiento de la normativa
- SDK & Lenguajes Soportados para Desarrollar
- Protocolos de Comunicación Compatibles
- Almacenamiento de Datos
- Escalabilidad
- Seguridad
- Facilidad de Uso
- Precio
- Análisis de Datos
- Gestión de Dispositivos
- Soporte para Desarrollo de Aplicaciones
- Hardware Compatible
- Formatos de Serialización Compatibles
- ¿La plataforma permite enviar comandos a los dispositivos?
- Gestión de Usuarios y Roles
- Requisitos No Funcionales
- Dominios o Casos de Uso Respaldados

7.1.1. Empresa hostelera

Nos encontramos frente a una empresa hostelera "*El IoTel*", la cual ha adoptado el uso de la tecnología IoT y la ha unificado a su anterior modelo de negocio para tratar de modernizarse.

El uso de la tecnología IoT en este ámbito destaca por aportar numerosos beneficios, tanto en términos de eficiencia operativa como de mejora de la experiencia del cliente. En concreto, en este contexto, los beneficios obtenidos fueron:

- Optimización de la gestión de recursos: La implementación de dispositivos IoT permitió monitorear en tiempo real el consumo de energía, agua y alimentos, identificando áreas de ineficiencia y optimizando su uso, lo que resultó en ahorros significativos.

- Automatización de servicios: Gracias a la tecnología IoT, se pudo simplificar el proceso de *check-in* y *check-out* mediante el uso de llaves electrónicas enviadas a los dispositivos móviles de los huéspedes, mejorando la eficiencia del servicio.

- Personalización de la experiencia del huésped: Mediante la recopilación de datos a través de dispositivos IoT, se personalizó la experiencia del huésped en aspectos como iluminación, temperatura de la habitación y actividades de interés.

- Mejora de la seguridad del recinto: El nivel de seguridad del hotel se elevó gracias a la implementación de cámaras de seguridad conectadas y cerraduras inteligentes mediante la tecnología IoT, proporcionando un ambiente más seguro para huéspedes y personal.

- Eficiencia en el mantenimiento de las instalaciones: Los dispositivos IoT permitieron identificar y abordar posibles problemas de mantenimiento antes de que ocurran fallos costosos, como es el caso de los sensores de temperatura en la cocina del hotel.

- Gestión de inventario optimizada: Los sensores IoT hicieron seguimiento de los niveles de stock en tiempo real, lo que permitió una reposición oportuna de productos y redujo el desperdicio.

Tras modernizarse ahora *El IoTel* busca implementar o contratar los servicios de una plataforma IoT para manejar y enriquecerse todavía más gracias a toda la información que generan los diferentes sensores que han instalado.

De entre todas las características de plataformas IoT propuestas el hotel ha escogido aquellas que considera de mayor interés para sus intereses, es decir, buscan una plataforma que ofrezca la capacidad de manejar eficientemente la comunicación entre los dispositivos, proporcionar análisis de datos en tiempo real, garantizar la seguridad de los datos, gestionar eficientemente los dispositivos y sea escalable para soportar el crecimiento del hotel.

Desglose y explicación del porqué de la selección de cada una de esas características ordenadas de mayor a menor en cuanto a su relevancia:

1. Seguridad: En el contexto de "*El IoTel*", la seguridad es vital ya que se manejarán datos sensibles tanto de la operación del hotel como de los huéspedes. Las plataformas IoT deben contar con sólidas medidas de seguridad para proteger contra el acceso no autorizado, manipulación de datos y ataques cibernéticos.

2. Gestión de dispositivos: El hotel tendrá múltiples dispositivos IoT implementados en toda la propiedad, desde llaves de puertas electrónicas hasta sensores de iluminación y temperatura. Una eficaz gestión de dispositivos es esencial para garantizar que todos estos dispositivos funcionen correctamente, se actualicen oportunamente y se manejen las fallas de manera eficiente.

3. Análisis de datos: Los dispositivos IoT generarán una gran cantidad de datos que, si se analizan correctamente, pueden proporcionar valiosos conocimientos. Por ejemplo, se puede mejorar la eficiencia energética analizando los datos de los sensores de temperatura, o se puede mejorar la experiencia del huésped al identificar y prever sus necesidades a través del análisis de patrones de comportamiento.

4. Protocolos de comunicación soportados: Con la variedad de dispositivos IoT, puede haber diferentes protocolos de comunicación en uso. La plataforma IoT debe ser capaz de soportar estos diferentes protocolos para garantizar una integración y comunicación fluida entre los dispositivos.

5. Almacenamiento de datos: El almacenamiento de datos es una característica importante de una plataforma IoT por varias razones. Facilita la gestión eficiente de la vasta cantidad de datos que los dispositivos IoT generan en tiempo real, lo que a su vez permite respuestas rápidas y automatizaciones. Además, estos datos almacenados son una fuente valiosa para analizar tendencias históricas y hacer proyecciones futuras, mejorando la eficiencia y la planificación del hotel.

Además, la seguridad y privacidad de los datos son vitales, especialmente cuando los datos implican información sensible de los huéspedes. Un correcto almacenamiento garantiza que esta información esté protegida y se maneje de acuerdo con las normativas de protección de datos. Por último, el almacenamiento de datos en un formato compatible permite su fácil integración con otros sistemas del hotel, promoviendo la coherencia y la interoperabilidad en todas las operaciones del hotel.

6. Soporte para Desarrollo de aplicaciones: La capacidad de desarrollar y personalizar aplicaciones a partir de la plataforma IoT permite al hotel mejorar la experiencia del huésped. Por ejemplo, se podrían crear aplicaciones personalizadas para permitir a los huéspedes controlar la iluminación y la temperatura de sus habitaciones desde sus teléfonos. Este punto está muy relacionado con el siguiente.

7. ¿La plataforma permite enviar comandos a las cosas?: Esto es esencial para automatizar procesos en el hotel. Por ejemplo, se podrían automatizar acciones como ajustar la temperatura de la habitación antes de la llegada de un huésped o apagar las luces cuando una habitación no está ocupada.

8. Escalabilidad: A medida que el hotel crezca y se añadan más dispositivos, la plataforma debe ser capaz de escalar para soportar este crecimiento. Debe ser capaz de manejar un creciente volumen de datos y número de dispositivos sin que se vea afectado su rendimiento.

A continuación, se explicará brevemente porqué las siguientes características no se han considerado relevantes a la hora de escoger una plataforma IoT:

1. Gestión de usuarios y roles: Este aspecto es menos crítico ya que en un hotel los usuarios con acceso a la plataforma (por ejemplo, los gerentes y el personal de mantenimiento) puede ser un número relativamente pequeño, por lo que una gestión de roles y usuarios muy granular podría no ser necesaria.

2. SDK y lenguajes soportados para desarrollar: El desarrollo de aplicaciones personalizadas puede no ser una prioridad para el hotel, especialmente si la plataforma IoT ya ofrece funcionalidades que cubren sus necesidades. Además, si el hotel no tiene un equipo de desarrollo interno y depende de proveedores externos para estas tareas, los lenguajes y SDK soportados podrían no ser una consideración crítica.

3. Costes: Aunque el costo siempre es una consideración importante, el hotel puede estar dispuesto a asumir un costo más alto si la plataforma proporciona las funcionalidades que necesita y si el retorno de la inversión, en términos de eficiencia operativa y una mejor experiencia del huésped, justifica el costo.

4. Hardware compatible: Aunque la compatibilidad con el hardware es importante, la mayoría de las plataformas IoT modernas son compatibles con una amplia gama de dispositivos. Por lo tanto, a menos que "*El IoTel*" esté utilizando algún hardware muy específico o poco común, esta característica podría no ser tan relevante.

5. Formatos de serialización soportados: La mayoría de las plataformas IoT modernas soportan una variedad de formatos de serialización comunes, por lo que a menos que el hotel tenga necesidades muy específicas en este sentido, esta característica podría ser menos relevante.

En base a las necesidades del hotel y las características de relevancia destacadas, dos de las plataformas más robustas y versátiles que podrían considerarse son *Amazon Web Services IoT Core* y *Microsoft Azure IoT Hub*.

Amazon Web Services (AWS) IoT Core es una opción conveniente para *El IoTel*, gracias a su capacidad para conectar y gestionar de manera segura una gran cantidad de dispositivos IoT. Su arquitectura en la nube permite un flujo constante de información en tiempo real, crucial para

optimizar las operaciones y la experiencia del huésped. Además, la compatibilidad con protocolos de comunicación ligeros como MQTT mejora la eficiencia de la red, especialmente en un entorno de hotel con conexiones intermitentes. En resumen, AWS IoT Core cumple con las demandas de escalabilidad, seguridad y gestión de grandes volúmenes de datos, lo que lo hace muy atractivo para el hotel.

Por otro lado, *Microsoft Azure IoT Hub* es una plataforma completamente administrada que ofrece confiabilidad, escalabilidad, seguridad y analítica avanzada. Ofrece soluciones *end-to-end*, desde el dispositivo hasta la nube, y su escalabilidad la hace adecuada para aplicaciones de IoT de todos los tamaños. Azure IoT Hub también permite a los desarrolladores codificar la lógica de la aplicación para la integración de los dispositivos IoT en los sistemas *back-end* del hotel.

Ambas opciones ofrecen un alto nivel de seguridad, soporte para una amplia variedad de protocolos de comunicación, y la capacidad de manejar un gran volumen de datos, lo cual es esencial dado el gran número de dispositivos IoT que pueden estar operando en un entorno de hotel. Adicionalmente, ambas plataformas permiten el análisis de datos y la integración con otros sistemas o plataformas de almacenamiento de datos.

La elección entre *AWS IoT Core* y *Azure IoT Hub* puede depender de otros factores, como las preferencias de desarrollo, la compatibilidad con los sistemas existentes del hotel y el costo. Ambas son opciones sólidas con capacidades de administración de dispositivos, análisis de datos y almacenamiento de datos de alto rendimiento que cumplen con las necesidades del hotel "*El IoTel*".

7.1.2. Formula 1

La Fórmula 1 es uno de los deportes guiados por datos más emocionantes, tan competitivo que incluso una décima de segundo de ventaja puede cambiar el resultado de una carrera. Los equipos de F1 se esfuerzan por encontrar esa ventaja utilizando las mejores herramientas de análisis y plataformas de inteligencia artificial y machine learning, capaces de analizar miles de datos por segundo.

Cada coche de F1 posee alrededor de 300 sensores los cuales generan 1,1 millones de datos telemétricos por segundo que son transmitidos desde los coches a los boxes. Sabiendo que hay 20 vehículos corriendo en un circuito en un momento dado, se puede estimar que cada fin de semana de carrera se generan aproximadamente 160TB (160.000.000MB) de datos.

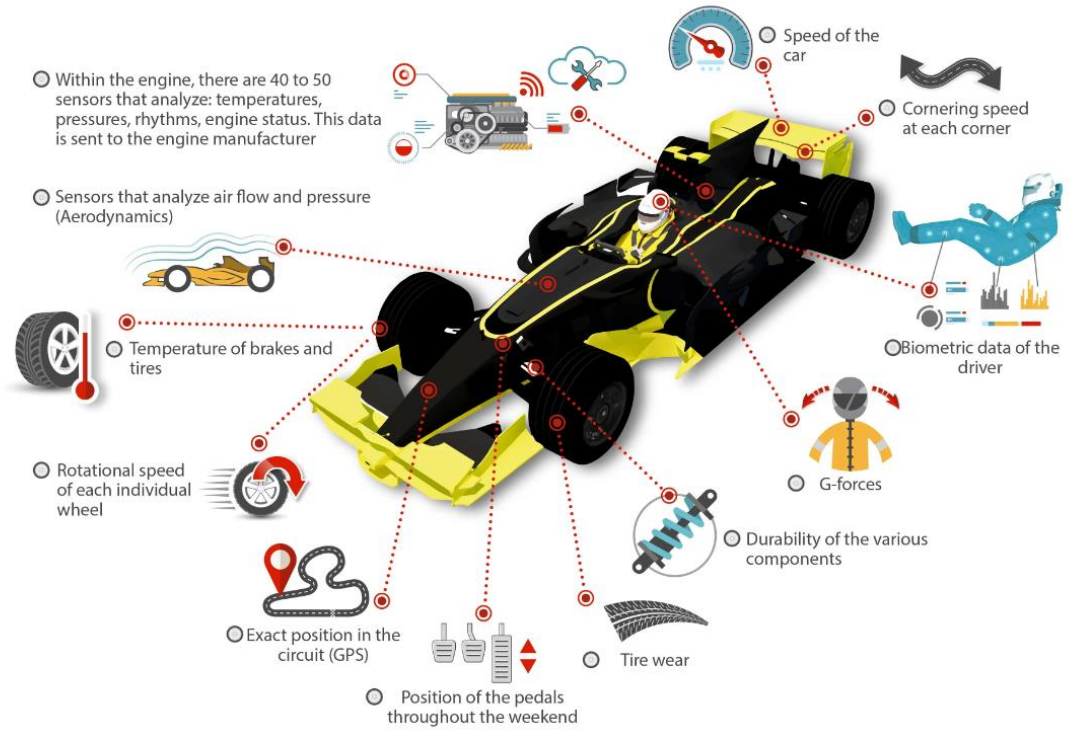

Ilustración 47. Ejemplos de diferentes datos recopilados por los sensores en un coche de F1

Estos datos telemétricos son cruciales para los equipos, sobre todo en las pruebas de pretemporada, entrenamientos y clasificación, por 2 razones principales:

- Comparar el rendimiento: Permite a los pilotos comparar su rendimiento con el de los demás para entender como están respecto a sus compañeros de equipo, y en cierta medida, compararse a sus rivales, cuya información particular de los sensores del coche es privada, pero valores como el tiempo de vuelta o histórico de tiempos no.

- Comprobación de fiabilidad: Ayuda a los equipos a supervisar el coche, comprobar que funciona correctamente y tomar las decisiones más adecuadas en función de los datos recopilados.

En pocas palabras, los datos de telemetría son la mejor forma que tienen los equipos de saber exactamente cómo están funcionando sus coches, cómo están funcionando sus pilotos y analizar las carreras pasadas para mejorar continuamente.

Actualmente, la FIA (Federación Internacional del Automóvil), entidad que dirige la competición, emplea los servicios de AWS para recolectar, analizar y aprovechar los datos telemétricos para tomar decisiones, pero imaginemos que esto no es así, y que la FIA se encuentra en la situación de seleccionar una plataforma para llevar a cabo todas estas gestiones de datos.

La FIA es consciente de que en el mundo de las carreras de Fórmula 1, los datos son los reyes y que cada curva, cada cambio de marcha, cada fracción de segundo puede significar la diferencia entre la victoria y la derrota. Este vasto y diverso flujo de datos es la base de los equipos modernos de Fórmula 1, ya que permite optimizar el rendimiento de los coches, elaborar estrategias en tiempo real y buscar constantemente la ventaja competitiva que puede llevarles a la victoria. Para gestionar este flujo masivo y complejo de datos, se está buscando una plataforma IoT que permita a los equipos gestionar sus dispositivos, proteger su flujo de información y almacenar y analizar datos con un nivel de profundidad y complejidad que puede elevarse exponencialmente.

Sin embargo, no todas las plataformas IoT son iguales, y las necesidades específicas de las carreras de Fórmula 1 exigen una cuidadosa consideración de las características de la plataforma. Estas características son fundamentales para gestionar y utilizar eficazmente los datos, lo que influye en el rendimiento del equipo y en su ventaja competitiva.

Teniendo en cuenta las demandas únicas de las carreras de F1, las características más relevantes de una plataforma IoT ordenadas de mayor a menor en cuanto a su grado de relevancia son:

1. Análisis de datos (y Visualización): La clave para obtener una ventaja competitiva en la F1 reside en la capacidad de interpretar y comprender las ingentes cantidades de datos que se producen. Cada coche genera 1,1 millones de puntos de datos telemétricos por segundo, que deben analizarse en tiempo real para tomar decisiones estratégicas durante la carrera. Una plataforma con un sólido sistema de análisis de datos permite obtener información inmediata sobre el rendimiento del coche y del piloto, las condiciones de la pista y los posibles problemas mecánicos. Además, las capacidades de aprendizaje automático pueden aprovechar los datos históricos para realizar análisis predictivos para la formulación de estrategias.

2. Seguridad: Los datos telemétricos de la Fórmula 1 son increíblemente sensibles. Un acceso no autorizado podría proporcionar a otros equipos información muy valiosa sobre las estrategias y los parámetros de rendimiento de un competidor. Por lo tanto, es esencial que una

plataforma IoT proporcione medidas de seguridad estrictas, incluyendo el cifrado de datos, controles de acceso seguro y monitoreo continuo de posibles amenazas a la seguridad.

3. Almacenamiento de datos: Cada fin de semana de carreras se generan unos 160 terabytes de datos, por lo que una plataforma no sólo debe ofrecer una gran capacidad de almacenamiento, sino también una gestión eficaz de los datos. Esto incluye la capacidad de organizar, recuperar y archivar datos para futuros análisis y mejoras continuas. La velocidad también es esencial, ya que cualquier retraso en la recuperación de datos puede suponer la diferencia entre el primer y el segundo puesto.

4. Escalabilidad: La plataforma debe ser escalable para dar cabida a los enormes volúmenes de datos generados en tiempo real y para hacer frente a posibles aumentos futuros a medida que evolucione la tecnología de sensores y se monitoricen potencialmente más fuentes de información.

5. Protocolos de comunicación soportados: La plataforma IoT debe admitir protocolos de comunicación rápidos, eficientes y fiables para la transmisión en tiempo real de datos telemétricos desde los coches a los boxes. Protocolos como MQTT o WebSockets, diseñados para comunicaciones intermitentes y de alta velocidad, podrían suponer una ventaja significativa en este entorno de alta velocidad.

6. Gestión de dispositivos: La plataforma debe ofrecer sólidas capacidades de gestión de dispositivos, que permitan a los equipos supervisar la salud, el rendimiento y la salida de datos de los sensores. Con más de 300 sensores en cada coche, los equipos necesitan estar seguros de que cada uno funciona correctamente para garantizar la precisión de los datos telemétricos.

7. SDK y lenguajes soportados para desarrollar: La compatibilidad con una serie de SDK y lenguajes de desarrollo puede permitir a los equipos adaptar sus aplicaciones y cuadros de mando a sus necesidades específicas, lo que permite una interpretación más eficiente de los datos y la toma de decisiones.

8. Gestión de usuarios y roles: Esta característica garantiza que los subconjuntos de datos específicos sean accesibles a los usuarios adecuados dentro de un equipo de F1. Esto optimiza la toma de decisiones al proporcionar los datos adecuados a las personas adecuadas, desde los ingenieros que necesitan datos técnicos en tiempo real hasta los especialistas en la estrategia que analizan las tendencias de rendimiento a largo plazo. Aunque importante, esta función tiene menos prioridad que otras, ya que se espera que los sistemas internos del equipo gestionen el control de acceso de forma eficaz.

En esencia, a la hora de seleccionar una plataforma IoT para gestionar los datos telemétricos de la Fórmula 1, la FIA debería dar prioridad a aquellas que ofrezcan un análisis de datos sólido y en tiempo real, un almacenamiento de datos seguro y eficiente, medidas de seguridad estrictas y una gestión eficaz de los dispositivos.

Las características que no son tan cruciales en este contexto incluyen:

1. Desarrollo de aplicaciones: La capacidad de desarrollar aplicaciones puede ofrecer personalización para atender las necesidades específicas de los equipos. Sin embargo, en el contexto de la Fórmula 1, las soluciones prefabricadas que responden a las necesidades específicas de este deporte pueden ser más valiosas.

2. Coste: Aunque el coste siempre es un factor a tener en cuenta, en el mundo de la Fórmula 1, con un presupuesto tan elevado, es probable que se dé más importancia a la funcionalidad, fiabilidad y eficacia de la plataforma que al coste.

3. ¿La plataforma permite enviar comandos a las cosas?: En el contexto de la F1, el envío directo de comandos puede no ser tan crucial, ya que la toma de decisiones en tiempo real depende principalmente de las entradas humanas.

4. Hardware compatible: Ya que cada equipo de F1 desarrolla su propia tecnología de sensores. Estos sensores son altamente especializados, y los equipos cuentan con expertos internos que pueden programarlos a medida para que funcionen con la mayoría de las plataformas IoT. Por lo tanto, la flexibilidad de una plataforma para admitir una amplia variedad de hardware no es una preocupación capital.

5. Formatos de serialización soportados: La mayoría de las plataformas IoT avanzadas ya están preparadas para manejar formatos de serialización universalmente aceptados como JSON, XML y Protobuf. Dado que estos formatos son habituales en la F1, la gran la mayoría de las plataformas serían suficientes, reduciendo la relevancia de esta característica en el proceso de selección.

Estos aspectos, aunque importantes en otros contextos, no aportan beneficios significativos para las exigencias únicas de la gestión de datos de telemetría en la Fórmula 1.

Dada la naturaleza exigente y de alto rendimiento de la Fórmula 1, y teniendo en cuenta los requisitos detallados, como la alta capacidad de cálculo, el procesamiento y análisis masivos de datos y la posible necesidad de integrar herramientas de terceros para complementar y o aumentar las modalidades de análisis, como, por ejemplo, usar Grafana para visualizar de una manera más efectiva todos los datos visualizados, sugeriría que *Google Cloud IoT Core* o *AWS IoT Core* serían excelentes opciones.

Google Cloud IoT Core ofrece capacidades de HPC (*High Performance Computing* (Computación de Alto Rendimiento)) robustas y un ecosistema de análisis de datos, incluyendo BigQuery para análisis a gran escala y Data Studio para la visualización de datos. Su capacidad para integrar con *TensorFlow* permite modelos de *machine learning* avanzados, que podrían ser esenciales en la extracción de conocimiento significativo de los vastos datos de telemetría.

Amazon Web Services IoT Core, por otro lado, también proporciona un potente ecosistema de IoT y HPC con una sólida infraestructura en la nube. AWS ofrece una variedad de servicios de análisis, desde el análisis de streaming en tiempo real hasta el análisis de macrodatos y el aprendizaje automático. A su vez, e es compatible con un gran número de servicios AWS adicionales, que permiten una amplia gama de análisis y métodos de visualización.

Por último, *IBM Watson IoT* podría ser otra alternativa viable, dada su avanzada capacidad de análisis de datos y aprendizaje automático. No obstante, *Google Cloud IoT Core* y *AWS IoT Core* parecen tener una oferta más sólida y flexible en términos de HPC y capacidad de análisis, lo que las convierte en candidatas fuertes para la F1.

En conclusión, a medida que desgranamos las exigentes demandas de la gestión de datos en el contexto de la Fórmula 1, queda claro que la elección de *AWS IoT Core* por parte del equipo de especialistas de la FIA no fue un accidente. Con su robusta capacidad de HPC, versatilidad en el análisis de datos y facilidad de integración con herramientas de terceros, la alinean perfectamente con las necesidades de este entorno. Por lo tanto, este ejercicio no solo ilustra el proceso de elección de la plataforma IoT más adecuada, sino que también valida la decisión de la FIA de confiar en AWS para impulsar el futuro de la Fórmula 1.

7.1.3. Domótica adaptada a un paciente diabético

Este contexto actualmente es ficticio, pero se trata de una promesa a futuro de un proyecto personal que quiero completar. Antes de realizar este TFM, comencé otro que me vi obligado a cancelar debido a mi falta de conocimiento en diferentes leguajes de programación, en los que mi nivel era extremadamente bajo comparado con aquel que exigía este proyecto, y a la incapacidad de desencriptar los datos recibidos por los sensores corporales que reportaban a una aplicación de soporte, y a la inexistencia de APIs, por parte de la aplicación, para acceder a estos datos.

La idea principal de este proyecto era desarrollar a través de la inteligencia ambiental, una solución que facilitase y mejorase mi vida como paciente diabético. El resultado que se esperaba obtener era el de convertir mi casa en una *Smart House* que estuviese constantemente leyendo la información enviada por un sensor que se encuentra conectado a mi brazo, cuya función es la de medir el nivel de glucosa en sangre, y en función de los valores obtenidos, realizar una determinada acción, que podría ser, por ejemplo, hacer sonar una alarma en la Alexa si el nivel de azúcar bajaba de cierto valor.

El proyecto se abandonó cuando tras conseguir leer mediante RFID (Identificación por Radio Frecuencia), gracias a un módulo Raspberry Pi, el flujo de datos enviados por el sensor, fui incapaz de descifrar esa información. Pero imaginemos que eso nunca sucedió, y que Miguel Ángel fue capaz de tener toda la información a partir de la que partir y comenzar a sacar provecho de ella.

En este caso, de todas las características estudiadas, para llevar a cabo este proyecto, es crucial seleccionar una plataforma IoT adecuada que permita interactuar con el sensor de glucosa, procesar los datos, y activar las alertas y acciones necesarias en tiempo real. Algunas características de estas plataformas son más relevantes que otras para este contexto, por lo que en las siguientes líneas las analizaré en detalle para identificar cuáles son las más importantes en este escenario particular:

1. Facilidad de uso: Este es el aspecto más importante para este proyecto personal. Es fundamental que la plataforma sea fácil de usar y entender, especialmente para un individuo que no tiene acceso a un equipo completo de ingenieros y especialistas en TI. La plataforma debe tener una interfaz de usuario intuitiva y una documentación completa y comprensible para ayudarme a entender cómo funciona y cómo se puede utilizar para satisfacer mis necesidades.

2. Costes: Dado que se trata de un proyecto personal y no de una gran empresa, es importante que los costes sean mínimos. Una plataforma gratuita o que ofrezca una versión básica (con limitaciones) gratuita sería ideal. Dado que el sensor de glucosa enviará una cantidad mínima de datos por minuto, es probable que no me vea afectado por las limitaciones que podrían tener algunas opciones gratuitas.

3. Análisis de datos: Aunque no se vayan a realizar análisis de datos a gran escala, sigue siendo una característica primordial. La capacidad de analizar y visualizar en tiempo real los datos

de glucosa puede ser vital para tomar decisiones informadas. Sería de gran ayuda que la plataforma escogida facilitase el desarrollo ofreciendo un motor de reglas con el cual planificar todas las acciones que ha de efectuar en función del valor leído.

4. ¿La plataforma permite enviar comandos a las cosas?: La capacidad de enviar comandos a los dispositivos conectados es clave para implementar acciones basadas en los niveles de glucosa, como activar una alarma en el dispositivo Alexa o notificar en la pantalla de un electrodoméstico inteligente (nevera en este caso) la recomendación de tomar un producto dulce, el cual sabe que está dentro de ella (paso que estamos seguros que se podría conseguir gracias a la innovación que ha experimentado este campo, que a día de hoy permite que existan neveras con gestor de inventario, del cual nos aprovecharíamos para mostrar estos mensajes)

5. Gestión de dispositivos: La plataforma deberá ser capaz de gestionar el sensor de glucosa y cualquier otro dispositivo que decidas añadir en el futuro a tu casa inteligente, como alarmas o electrodomésticos inteligentes.

Tras realizar el análisis individual de las características que ofrecían cada una de las plataformas, se pudo observar que el protocolo de comunicación, MQTT, como el lenguaje de programación, Python, así como el formato de serialización de mensajes, JSON, que se querían emplear eran tan comunes y ampliamente utilizados, que la mayoría de las plataformas los soportaban. Por lo que pese a ser características realmente decisivas a la hora de escoger una plataforma, se vio que de ninguna manera estas llegarías a ser un problema en la elección.

1. Protocolos de comunicación soportados

2. SDK y lenguajes soportados para desarrollar

3. Formatos de serialización soportados

En cuanto a las características que no se consideran relevantes a la hora de seleccionar una plataforma encontramos:

1. Seguridad: Ya que este proyecto es a pequeña escala y personal, y el carácter de los datos no es sensible pese a ser datos de salud, la seguridad de los datos no llega a ser un punto vital es vital.

2. Almacenamiento de datos: No consideramos el almacenamiento de datos relevante, ya que, las funcionalidades que se encargan de llevar un seguimiento a largo plazo y para hacer un análisis retrospectivo de los niveles de glucosa ya las ofrece la aplicación oficial del sensor.

3. Escalabilidad: En este caso, donde el número de dispositivos y la cantidad de datos es relativamente pequeña, la escalabilidad no es un factor determinante en la elección de la plataforma.

4. Gestión de usuarios y roles: Aunque es una característica útil, no es prioritaria en este escenario ya que el único usuario sería yo.

Teniendo todo esto en cuenta, llego la hora de tomar la decisión de que plataforma será la más idónea con la que trabajar. Realmente, esta decisión fue tomada hace tiempo, ya que pese a abandonar el proyecto, nunca se olvidó. Al analizar cada una de las plataformas, siempre se planteaba y respondía a la pregunta "*¿Escogería esta plataforma en un futuro?*", debido a esto, tras analizar todas, la elección ya había sido tomada, *Sofia2*.

La decisión de optar por *Sofia2* no fue arbitraria, fue guiada por características clave que se alineaban con las necesidades específicas del proyecto.

En primer lugar, y para mí el más importante, la facilidad de uso de *Sofia2* destacó desde el principio. Su interfaz intuitiva, junto con una documentación detallada y fácil de seguir, ofreció la accesibilidad necesaria para un proyecto personal. La perspectiva de enfrentarse a una curva de aprendizaje no parecía tan desalentadora con la ayuda de la documentación y las guías proporcionadas.

En segundo lugar, en lo que respecta a los costos, resultó ser una opción asequible. La plataforma ofrece una versión básica gratuita que, aunque tiene limitaciones, es suficientemente robusta para satisfacer las necesidades del proyecto.

Además, partiendo de que la capacidad de análisis de datos en tiempo real es esencial, esta plataforma no sólo cumple con esto, sino que también proporciona una visión significativa de los datos, facilitando la toma de decisiones informadas. Aquí es donde entra en juego la característica a subrayar de *Sofia2*, su motor de reglas. Este permite definir condiciones y acciones personalizadas.

La capacidad de Sofia2 para enviar comandos a los dispositivos es también un aspecto relevante. En combinación con su excepcional manejo en la gestión de dispositivos y junto a los aspectos esenciales previamente comentados, *Sofia2* emerge como la opción ideal para este proyecto de domótica centrado en la atención de la diabetes.

8. Conclusiones

Durante la realización de este Trabajo de Fin de Máster, hemos abordado la creciente relevancia de la Internet de las Cosas (IoT) y su aplicación en diferentes dominios. Hemos explorado y analizado diversas soluciones industriales existentes, resaltando sus características y funcionalidades. Esta labor nos ha permitido organizar y sintetizar un catálogo de características tanto básicas como avanzadas que estas plataformas pueden ofrecer.

El primer objetivo propuesto fue la exploración y análisis de las soluciones ámbito de la IoT, el cual se ha cumplido satisfactoriamente. Hemos elaborado un inventario exhaustivo de soluciones actuales, destacando no sólo la gama de funcionalidades que ofrecen, sino también sus respectivas fortalezas y limitaciones. Esto no sólo nos ha permitido ganar una comprensión profunda del paisaje de las soluciones IoT disponibles, sino que también nos ha capacitado para identificar criterios de selección efectivos para empresas, instituciones u organizaciones que buscan la solución más adecuada para sus necesidades específicas.

En el segundo objetivo, conseguimos organizar y sintetizar la información recopilada en un catálogo de características fácilmente visualizable y entendible. Este catálogo puede ser de gran utilidad para quienes busquen un marco de referencia al seleccionar una plataforma para desarrollar una solución IoT.

El tercer objetivo se centró en el desarrollo de escenarios ilustrativos. Cada escenario permitió evaluar cuáles de los frameworks o plataformas se adaptan mejor a sus requerimientos específicos. Este objetivo aportó un marco práctico y contextual para entender mejor la aplicabilidad y utilidad de las diferentes características en situaciones reales. Finalmente, este objetivo se completó mediante la selección de una opción para el desarrollo de una solución IoT orientada a la conversión de una vivienda en una '*Smart house*' diseñada para personas con diabetes. Este proceso nos permitió aplicar los conocimientos adquiridos y demostrar nuestra habilidad para seleccionar la solución más adecuada basándonos en requisitos específicos.

Durante el desarrollo de este Trabajo de Fin de Máster, nos hemos enfrentado a una serie de retos significativos, entre ellos, la vasta cantidad de soluciones disponibles y la complejidad de las características inherentes a cada una de ellas. Sin embargo, el uso de metodologías de investigación y análisis rigurosos nos permitió superar estos obstáculos y lograr un entendimiento profundo de las tecnologías IoT. Este proyecto no sólo nos permitió adentrarnos en el mundo de la IoT y sus aplicaciones, sino también mejorar nuestro entendimiento de las tecnologías de vanguardia, y adquirir habilidades valiosas en el análisis y evaluación de soluciones tecnológicas.

Cuando se alcanzaba el final de este trabajo se identificaron diversas áreas que podrían beneficiarse de una investigación adicional, así como una serie de posibles mejoras o ampliaciones derivadas de los resultados de los análisis realizados, que, dado el tiempo restante y la naturaleza

compleja de los mismos, no fueron posibles abordar. A continuación, discutiremos en detalle esos asuntos que quedaron sin resolver.

8.1. Trabajos futuros

En lo que respecta a los trabajos futuros, varias cuestiones surgieron durante el desarrollo de este TFM que podrían beneficiarse de una investigación adicional.

Un aspecto que nos hubiera gustado abordar en mayor profundidad es el análisis de diferentes soluciones IoT, las cuales desconocíamos hasta ya entrados en las fases finales del proyecto. En este trabajo se ha referenciado en numerosas ocasiones al "Cuadrante Mágico de Gartner", y fue gracias a él que descubrimos plataformas que en dicho documento son consideradas como plataformas "Visionarias" o "Retadoras", por lo que son de gran relevancia, pero que no aparecen en este TFM. Dichas plataformas son: PTC, Cumulocity IoT (Software AG), Lumada IIoT (Hitachi), MindSphere IIoT (Siemens) y Ability Genix IIoT (ABB).

Por otra parte, consideramos fundamental que el catálogo de características que hemos desarrollado se mantenga y actualice según las nuevas soluciones y tecnologías que surjan en el campo de la IoT. Con el ritmo de desarrollo en este campo, es probable que surjan nuevas funcionalidades y se realicen mejoras en la eficiencia, lo que justificaría futuras actualizaciones y revisiones de nuestro trabajo.

Por último, durante el desarrollo de este proyecto surgieron numerosas ideas que podrían aportar un valor significativo, de entre las que destacaron realizar un "Árbol de decisión", idea que evoluciono a la del desarrollo de una "App formulario" que recomendase una plataforma IoT en función de las respuestas proporcionadas. Finalmente, esta idea fue descartada debido a la dificultad y tiempo que consideramos que podría acarrear realizarla, ya que el nivel de profundidad alcanzado en el análisis, con 17 características diferentes, provocaba que la dificultad aumentase exponencialmente. En futuras revisiones esta es una línea de desarrollo que podría incrementar el valor total de este increíble proyecto

9. Referencias

ARTÍCULOS DE INVESTIGACIÓN

[1] Ganguly, P., "Selecting the right IoT cloud platform," 2016 International Conference on Internet of Things and Applications (IOTA), Pune, India, 2016, pp. 316-320, https://doi.org/10.1109/IOTA.2016.7562744.

[2] Ray, P. P., "A survey of IoT cloud platforms", Future Computing and Informatics Journal, Volume 1, Issues 1–2, 2016, Pages 35-46, ISSN 2314-7288, https://doi.org/10.1016/j.fcij.2017.02.001.

[3] Grünberg, K., Schenck, W., (2018). "A Case Study on Benchmarking IoT Cloud Services". In: Luo, M., Zhang, LJ. (eds) Cloud Computing – CLOUD 2018. CLOUD 2018. Lecture Notes in Computer Science(), vol 10967. Springer, Cham. https://doi.org/10.1007/978-3-319-94295-7_28.

[4] Nakhuva, B. & Champaneria, T., (2015). "Study of Various Internet of Things Platforms". International Journal of Computer Science & Engineering Survey. 6. 61-74. https://doi.org/10.5121/ijcses.2015.6605.

[5] Nel, P. (2021). "Estudio comparativo de plataformas cloud que ofrecen servicios IoT" [Proyecto Dirigido 2, Especialización en Gestión en Redes de Datos, Universidad Santo Tomas]. Autoarchivo de Tesis y Trabajos de Grado de estudiantes de Pregrado, Maestrías, Doctorados y Especialización de la Universidad Santo Tomás https://repository.usta.edu.co/handle/11634/33347

[6] Cardoso, J., Pereira, C., Aguiar A. & Morla R., "Benchmarking IoT middleware platforms," 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Macau, China, 2017, pp. 1-7, https://doi.org/10.1109/WoWMoM.2017.7974339.

[7] Raza, S., Helgason, T., Papadimitratos, P., & Voigt, T. (2017). "SecureSense: End-to-end secure communication architecture for the cloud-connected Internet of Things". Future Generation Computer Systems, Volume 77, Pages 40–51. https://doi.org/10.1016/j.future.2017.06.008.

[8] Martinez, J. (2010). "Comparativa y estudio de plataformas IoT" [Trabajo Final de Grado, Universidad Politècnica de Catalunya]. Repositorio institucional de la Universidad Politècnica de Catalunya https://upcommons.upc.edu/bitstream/handle/2117/113622/TFG-RodrigoMartinezJacobson.pdf?sequence=1&isAllowed=y

[10] Velosa, A. & Goodness, E. (2022) "Magic Quadrant for Global Industrial IoT Platforms". Gartner. https://www.gartner.com/en/documents/4006918

[9] OTRAS FUENTES DE INFORMACIÓN Y ARTÍCULOS DE OPINIÓN

Techtic Solutions, Inc. (n.d.). *10 Best Open Source IoT Frameworks of All Time*. Techtic Solutions. https://www.techtic.com/blog/top-10-open-source-iot-frameworks/

Ilchenko, V. (n.d.). 5 Best Open Source IoT Frameworks. ByteAnt. https://www.byteant.com/blog/5-best-open-source-iot-frameworks/

Geekflare. (2022). 11 plataformas y herramientas de Internet de las cosas (IoT) de código abierto. Geekflare. https://geekflare.com/es/iot-platform-tools/

Pedamkar, P. (2023). IoT Framework. EDUCBA. https://www.educba.com/iot-framework/

10 Best IoT Platforms To Watch Out In 2023. (2023, Junio 24). Software Testing Help. https://www.softwaretestinghelp.com/best-iot-platforms/

Plataformas IoT de Código Abierto. (2023, Junio 27). Automatización Del Internet De Las Cosas. https://alfaiot.com/page/plataformas-iot-codigo-abierto

Editor. (2020, Mayo 20). Making Sense of IoT Platforms: AWS vs Azure vs Google vs IBM vs Cisco. AltexSoft. https://www.altexsoft.com/blog/iot-platforms/

Gracia, L. (2016, Noviembre 2). Tabla comparativa Plataformas IoT: Azure IoT Hub vs AWS IoT vs Watson IoT Foundation vs Sofia2 IoT Platform. Un Poco De Java. https://unpocodejava.com/2016/11/02/tabla-comparativa-plataformas-iot-azure-iot-hub-vs-aws-iot-vs-watson-iot-foundation-vs-sofia2-iot-platform/

Gracia, L. (2020, Noviembre 10). Offering de Servicios Cloud AWS, Google Cloud Platform y Azure (2020 updated). Un Poco De Java. https://unpocodejava.com/2020/08/18/offering-de-servicios-cloud-aws-google-cloud-platform-y-azure-2020-updated/

Thinger.io. (n.d.). SourceForge. https://sourceforge.net/software/product/Thinger.io/alternatives

10. Apéndice

Altair SmartWorks IoT	
Características	**Detalles**
Certificaciones y cumplimiento de la normativa	ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, IEC 62443-3-1:2020, IEC 62366, SOC 2, PCI DSS, HIPAA.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, Android y Sigfox entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, OPC UA y XMPP entre otros
Almacenamiento de Datos	Admite el almacenamiento en la nube, en las propias instalaciones (local), híbrido y en el perímetro. Compatible con BBDD NoSQL: AnythingDB y RealtimeDB.
Escalabilidad	Soporta un gran número de dispositivos y datos. Puede satisfacer las necesidades de cualquier organización, desde pequeñas empresas hasta grandes corporaciones.
Seguridad	Características de seguridad: encriptación, autenticación, autorización, control de acceso basado en roles, encriptación de datos, y detección de intrusiones.
Facilidad de Uso	Diseñado para ser fácil de usar, con una interfaz sencilla e intuitiva de arrastrar y soltar, variedad de tutoriales y documentación.
Precio	Plataforma por suscripción. El coste de la suscripción depende del número de dispositivos y datos que se utilicen. Está disponible en una variedad de planes de precios, incluyendo el nivel gratuito, estándar y de empresa.
Análisis de Datos	Proporciona herramientas de análisis de datos: *machine learning*, análisis predictivo, detección de anomalías, flujo de datos en tiempo real, análisis de datos históricos, visualización de datos y análisis estadístico.
Gestión de Dispositivos	Funciones de gestión de dispositivos: registro de dispositivos, actualizaciones de firmware, control remoto, aprovisionamiento de dispositivos y supervisión remota.
Soporte para Desarrollo de Aplicaciones	Funciones de desarrollo de aplicaciones: desarrollo mediante un constructor de arrastrar y soltar, scripting visual, generación de código, editor de código, testeo y un simulador.
Hardware Compatible	Admite dispositivos de hardware: sensores, actuadores, *gateways* y dispositivos de periferia (*edge*).
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf , YAML, Avro, INI y formatos de serialización Personalizados entre otros
¿La plataforma permite enviar comandos a los dispositivos?	Si

Gestión de Usuarios y Roles	El sistema de gestión de usuarios y funciones permite a los Administradores crear y gestionar usuarios y funciones. Proporciona un sistema de control de acceso basado en roles (*role-based access control* (*RBAC*)) que permite definir permisos para usuarios y grupos.
Requisitos No Funcionales	Tiene un Acuerdo de Nivel de Servicio (SLA) de alta disponibilidad del 99,9%. Diseñado para alta disponibilidad y rendimiento, soportando aplicaciones de misión crítica. Proporciona funciones de copia de seguridad de datos, recuperación, equilibrio de carga y almacenamiento en caché, con copias de seguridad automáticas a través de AnythingDB.
Dominios o Casos de Uso Respaldados	Compatible con una amplia gama de casos de uso de IoT en distintos sectores: Industria, Sanidad, Energía, Logística, Ciudades Inteligentes, Transporte, Comercio, etc...

Apéndice I. Altair SmartWorks IoT

Amazon Web Services IoT Core	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, SOC 2, PCI DSS, HIPAA, GDPR, FedRAMP, FIPS 140-2 Level 3. Reconocido por CSA y NIST.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C#, Go, .NET, Android, iOS y Arduino entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, MQTT/AMQP/HTTP sobre WebSockets, LoraWAN, WebSockets, LwM2M, AWS IoT Device SDKs y SNMP entre otros.
Almacenamiento de Datos	AWS IoT Core almacena los datos de los dispositivos en AWS Cloud, que ofrece diversas opciones de almacenamiento de datos, como Amazon Simple Storage Service (S3), Amazon DynamoDB, Amazon Redshift y Amazon RDS.
Escalabilidad	AWS IoT Core es un servicio altamente escalable que puede soportar miles de millones de dispositivos conectados y enrutar billones de mensajes.
Seguridad	Ofrece diversas funciones de seguridad, como autenticación de dispositivos, cifrado de extremo a extremo (todo el tráfico hacia y desde AWS IoT se envía de forma segura a través de *Transport Layer Security* (*TLS*)), cifrado de datos y control de acceso (*RBAC*) entre otras medidas.
Facilidad de Uso	AWS IoT Core es un servicio fácil de configurar y utilizar que proporciona diversos recursos para ayudar a los desarrolladores a comenzar, como tutoriales, documentación y código de muestra. Ofrece un módulo gratuito para el aprendizaje y cuenta con una API sencilla e intuitiva respaldada por una completa documentación.
Precio	AWS IoT Core es un servicio de pago por uso. Se cobra por el número de mensajes que envía y recibe, la cantidad de datos que almacena y el número de dispositivos que conecta. También se cobran la transferencia de datos, el almacenamiento de mensajes y el registro de dispositivos. La plataforma ofrece una solución rentable con 2,25 millones de minutos de conexión y 500.000 mensajes al mes durante 12 meses en la versión gratuita.
Análisis de Datos	AWS IoT Core proporciona diversas herramientas y servicios de análisis de datos, como Amazon Kinesis Analytics, Amazon QuickSight, Amazon Athena, streaming de datos en tiempo real, gestión por lotes (*batch*), *machine learning* y se integra con Amazon Redshift y Amazon S3.
Gestión de Dispositivos	Ofrece diversas funciones de gestión de dispositivos, como el registro de dispositivos, las actualizaciones de firmware, la gestión remota de dispositivos, el aprovisionamiento de dispositivos, la monitorización de dispositivos y Device Shadow para almacenar el estado de los dispositivos, lo que facilita la conexión, la gestión y el escalado de flotas de dispositivos.
Soporte para Desarrollo de Aplicaciones	Proporciona una variedad de funciones para el desarrollo de aplicaciones, como Device Shadow, un motor de reglas, integración con AWS Lambda, Amazon API Gateway, Amazon Cognito y Alexa Voice Service.
Hardware Compatible	Admite diversos dispositivos de hardware, como Raspberry Pi, Arduino, Intel Edison, dispositivos Amazon Sidewalk, sensores, dispositivos actuadores y gateways.
Formatos de Serialización Compatibles	Soporta: JSON, XML, Protobuf ,Binary, YAML, Avro, CBOR, MessagePack, Thrift, Cap'n Proto y formatos personalizados entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Es compatible con el control de acceso basado en roles (*RBAC*), es decir, permite crear y gestionar usuarios y roles, y ofrece funciones como la creación de usuarios, la creación de roles, la gestión de permisos, los privilegios de los roles y la asignación de roles.
Requisitos No Funcionales	Proporciona un servicio de alta disponibilidad con un SLA de tiempo de actividad del 99,9% y ofrece diversas características de desempeño como colas de mensajes y equilibrio de carga, y opciones de backup de datos, como Amazon S3 y Amazon Glacier.

Dominios o Casos de Uso Respaldados	AWS IoT Core es una plataforma versátil que puede utilizarse para una amplia gama de casos de uso y dominios de IoT, entre los que se incluyen: Datos de automoción, Coches conectados, Productos de seguridad, Automatización Industrial, Ciudades Inteligentes, Rastreo de Activos, Fabricación, Logística y Sanidad.

Apéndice II. Amazon Web Services IoT Core

Carriots	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	Certificado por la EU y AWS. Miembro de la IoT Alliance. Cumple con GDPR.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js y Groovy.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, CoAP, XMPP, cURL y hURL.
Almacenamiento de Datos	Carriots pone a tu disposición una gran base de datos NoSQL (Amazon S3) que puede almacenar cientos de terabytes de datos con al menos dos servidores redundantes independientes y una gran capacidad transaccional
Escalabilidad	Escalado automático para gestionar de 1 a millones de dispositivos (no hay limitación de cuánto pueden escalar tus proyectos tanto en volumen como en rapidez)
Seguridad	Características: cifrado de datos, autenticación de usuarios y control de acceso basado en roles. Utiliza medidas de seguridad estándar del sector para proteger los datos.
Facilidad de Uso	Interfaz sencilla e intuitiva con un panel de control fácil de usar y un editor de aplicaciones de tipo "arrastrar y soltar".
Precio	En su día, previo a la adquisición de esta plataforma, ofrecía sus servicios para la creación de aplicaciones M2M de forma totalmente gratuita.
Análisis de Datos	Proporciona herramientas de análisis de datos: visualización de datos en tiempo real, análisis de datos históricos, *machine learning* e inteligencia artificial.
Gestión de Dispositivos	Funciones como aprovisionamiento de dispositivos (inscripción, configuración y autenticación en la red de la organización), actualizaciones de firmware y supervisión remota.
Soporte para Desarrollo de Aplicaciones	Funciones de desarrollo de aplicaciones: Constructor de tipo "arrastrar y soltar", API REST, IDE web, biblioteca de componentes pre-construidos y mercado de aplicaciones de terceros entre otras funciones.
Hardware Compatible	Compatible con Raspberry Pi, Arduino y ESP8266.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf ,Binary, YAML, Avro, CBOR, Thrift, Cap'n Proto e incluso formatos de serialización personalizados
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Permite crear y gestionar diferentes roles de usuario (ej: administrador, desarrollador, usuario). Permite controlar quién tiene acceso a los datos y dispositivos.
Requisitos No Funcionales	SLA de alta disponibilidad del 99,9%. Puede gestionar hasta 1 millón de mensajes por segundo. Funciones: copia de seguridad diaria de los datos, recuperación en caso de catástrofe y replicación de datos.

Dominios o Casos de Uso Respaldados	Apto para muchos ámbitos: Fabricación, Sanidad, Logística, Distribución/Comercio, Agricultura, Ciudades inteligentes, Seguimiento de activos, Hogar inteligente, Automatización industrial.

Apéndice III. Carriots

DeviceHive	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	Bajo licencia Apache 2.0.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C#, Go, .NET y Ruby entre otros
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, MQTT/AMQP/HTTP WebSockets, LoraWAN, WebSockets y LwM2M entre otros.
Almacenamiento de Datos	Almacenamiento en la nube, almacenamiento local y almacenamiento híbrido. Ofrece diversas opciones de almacenamiento de datos, en BBDD Relacionales: MySQL, PostgreSQL; y en NoSQL: Cassandra, MongoDB, Elasticsearch,
Escalabilidad	Sistema basado en microservicios, construido con alta escalabilidad y disponibilidad. Plataforma que no solo puede escuchar cientos de dispositivos simultáneamente, sino también escalar a la cantidad requerida de instancias para garantizar la seguridad, disponibilidad de los datos y para gestionar el aumento de los niveles de producción. Gestionado y orquestado por Kubernetes.
Seguridad	Incorpora funciones de seguridad como autenticación de usuarios, cifrado de datos y control de acceso. Admite TLS, SSL, OAuth 2.0 y 2FA.
Facilidad de Uso	Fácil de usar, gracias a una interfaz web y un panel de control intuitivos. Ofrece documentación y tutoriales tanto para desarrolladores como para usuarios finales. Fácil de instalar y configurar.
Precio	El conjunto de herramientas de DeviceHive es completamente gratuito. El gasto que conlleva esta solución depende del hosting escogido para el despliegue de la plataforma, así como los servicios opcionales de pago disponibles: asistencia y formación.
Análisis de Datos	Apache Spark, Elasticsearch, Cassandra y Kafka para análisis en tiempo real y sin conexión, visualización de datos, gestión por lotes (*batch*) y *machine learning*.
Gestión de Dispositivos	Ofrece registro, aprovisionamiento, supervisión, control remoto y actualizaciones de firmware de dispositivos.
Soporte para Desarrollo de Aplicaciones	Amplia gama de APIs, herramientas, recursos y bibliotecas para el desarrollo rápido de aplicaciones IoT.
Hardware Compatible	Amplia gama de hardware IoT, incluidos Arduino, ESP8266, Raspberry Pi, BeagleBone, sensores, actuadores y gateways.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf ,Binary, Avro, CBOR, MessagePack, Thrift, Cap'n Proto e incluso formatos de serialización personalizados entre otros
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Sistema de gestión de usuarios que permite la creación y gestión de cuentas de usuario, grupos y roles (permite crear diferentes roles de usuario con diferentes permisos).

Requisitos No Funcionales	SLA de alta disponibilidad del 99,9%. Admite funciones de alto rendimiento como el equilibrio de carga y el almacenamiento en caché. Copia de seguridad de datos en varias localizaciones de almacenamiento: Amazon S3, Google Cloud Storage, Microsoft Azure, disco local y almacenamiento en la nube. Se realizan copias de seguridad de los datos con regularidad y se pueden restaurar en caso de desastre.
Dominios o Casos de Uso Respaldados	Se adapta a una amplia gama de casos de uso de IoT, como Hogares inteligentes, Ciudades inteligentes, Automatización industrial, Logística, Sanidad, Agricultura, Transporte, Distribución, Seguimiento de activos y Seguridad.

Apéndice IV. DeviceHive

Google Cloud IoT Core	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, SOC 2, PCI DSS, HIPAA, GDPR, FedRAMP. Reconocido por CSA.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C#, Go, .NET y Ruby entre otros
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, MQTT/AMQP/HTTP sobre WebSockets, LoraWAN, WebSockets, ModBus, XMPP, LwM2M, Google Cloud Pub/Sub, gRPC y DDS entre otros.
Almacenamiento de Datos	Datos almacenados y procesados mediante Google Cloud Storage, Bigtable, Datastore, Cloud Spanner, BigQuery y Cloud Pub/Sub. Aprovecha múltiples opciones de almacenamiento para un almacenamiento eficiente de los datos. Google Cloud Storage destaca por ser altamente escalable y fiable.
Escalabilidad	Diseñado para gestionar millones de dispositivos y escalar horizontalmente para adaptarse a despliegues IoT en constante crecimiento. Puede manejar grandes volúmenes de ingestión de datos, procesamiento y gestión de dispositivos de forma eficiente.
Seguridad	Ofrece sólidas funciones de seguridad, como autenticación de dispositivos, cifrado de datos de extremo a extremo, mecanismos de control de acceso y control de acceso basado en roles. Se integra con Google Cloud Identity and Access Management (*IAM*) para conceder permisos granulares de usuarios y dispositivos, el cual ofrece supervisión 24/7, detección de intrusiones. Este servicio está construido sobre la infraestructura de seguridad de clase mundial de Google.
Facilidad de Uso	Proporciona una interfaz web fácil de usar, CLI (Interfaz de Línea de Comandos), API, documentación y tutoriales. Diseñado para que resulte fácil para los desarrolladores, incluso sin experiencia previa en IoT.
Precio	Sigue un modelo de pago por uso basado en factores como el número de dispositivos, el volumen de datos procesados, la duración de la retención de datos, la cantidad de datos almacenados y los recursos utilizados.
Análisis de Datos	Se integra con servicios como Google Cloud Pub/Sub, Dataflow, BigQuery, Cloud Storage, Cloud Bigtable, Cloud Datastore y Cloud Dataproc para el análisis avanzado, almacenamiento y procesamiento en tiempo real de flujo de datos. También admite *machine learning*.
Gestión de Dispositivos	Ofrece funciones completas de gestión de dispositivos, incluidos el aprovisionamiento, el registro, la supervisión, las actualizaciones de firmware y la gestión del ciclo de vida de los dispositivos IoT.
Soporte para Desarrollo de Aplicaciones	Proporciona una serie de herramientas y servicios para el desarrollo de aplicaciones, como Cloud Functions, Cloud Run, Device Shadow, comandos de dispositivos, eventos de dispositivos, integración de dispositivos, análisis de datos, visualización de datos, mensajería de dispositivo a nube (y de nube a dispositivo) y servicios como Google Cloud SDK, Google Cloud Platform Console y Google Cloud APIs. Permite a los desarrolladores crear y desplegar aplicaciones IoT que procesen datos en tiempo real y activen acciones basadas en la telemetría del dispositivo.
Hardware Compatible	Compatible con una amplia gama de dispositivos y plataformas de hardware, como Raspberry Pi, Arduino, Intel Edison, Android Things, dispositivos con firmware Cloud IoT Core, dispositivos de terceros con conectividad compatible, sensores, actuadores, *gateways* y otros.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf ,Binary, YAML, Avro, CBOR, MessagePack, Thrift, mensajes Cloud Pub/Sub, TLV8, JWT e incluso formatos de serialización personalizados entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Admite la gestión de usuarios y funciones. Se integra con Google Cloud IAM para ofrecer permisos granulares de usuarios y roles, lo que incluye autenticación de usuarios, autorización de usuarios, control de acceso basado en roles (*RBAC*), creación de usuarios, asignación de roles y control de acceso para gestionar quién tiene acceso a los datos y dispositivos IoT.

Requisitos No Funcionales	Garantiza un SLA de alta disponibilidad, fiabilidad y rendimiento para despliegues IoT. Aprovecha la infraestructura global de Google para proporcionar una plataforma escalable y robusta. Puede gestionar millones de mensajes por segundo y proporciona un alto nivel de rendimiento. Se realizan copias de seguridad diarias de los datos, que pueden restaurarse en cualquier momento. Otras funciones son el equilibrio de carga y la geo-replicación.
Dominios o Casos de Uso Respaldados	Idóneo para IoT industrial, Ciudades Inteligentes, Agricultura, Sanidad, Distribución, Transporte, Automatización Industrial, Seguimiento de Activos, Vehículos Inteligentes (conectados), etc.

Apéndice V. Google Cloud IoT Core

IBM Watson IoT	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	ISO/IEC 27001, SOC 2, PCI DSS, HIPAA, GDPR, FedRAMP. Miembro de IoTSF, IIC, World Economic Forum.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C#, .NET, Android, iOS, Arduino y Ruby entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, MQTT/AMQP/HTTP sobre WebSockets, LoraWAN, WebSockets, ModBus, OPC UA, XMPP, LwM2M, IBM MQ, IBM IoT Message Gateway, BACnet, KAFKA, JMS, etc…
Almacenamiento de Datos	Ofrece almacenamiento de datos no estructurados (mediante el servicio: IBM Cloud Object Storage data buckets) en local, en híbrido y en la nube. Compatible con el uso de bases de datos relacionales (PostgresSQL, IBM Db2 Warehouse on Cloud) y NoSQL. Se integra con plataformas de almacenamiento de datos de terceros.
Escalabilidad	Escalable horizontalmente. Capaz de soportar despliegues de IoT de pequeño a gran tamaño y gestionar millones de dispositivos y miles de millones de mensajes al día. Watson IoT es un servicio en la nube y, como tal, a medida que aumente la demanda de cualquier servicio, se escalará la capacidad para satisfacer esa demanda. Los usuarios no tienen que preocuparse por eso, se hace automáticamente.
Seguridad	Está diseñado con la seguridad como prioridad. Incluye cifrado, autenticación de dos factores, autorización, control de acceso basado en roles y otras funciones de seguridad para proteger datos y dispositivos. Admite protocolos de seguridad como TLS, IPsec.
Facilidad de Uso	Ofrece un sencillo e intuitivo panel de control y API basado en web, así como numerosas funciones para facilitar la conexión de dispositivos, la recopilación de datos y la creación de aplicaciones. Proporciona una IU sencilla y bien definida donde se puede añadir y gestionar los dispositivos simple y fácilmente, controlar el acceso al servicio IoT y supervisar el uso.
Precio	Servicio de pago por suscripción que depende del número de dispositivos y el tráfico de datos. El volumen del almacenamiento y las funciones utilizadas son caracterices que también aumentan el precio de uso de la plataforma, si es que se da el caso de que Watson IoT trabaje en conjunto con otros servicios brindados por IBM.
Análisis de Datos	Ofrece *machine learning*, análisis predictivo, detección de anomalías, herramientas de visualización de datos (cuadros de mando, gráficos) e inteligencia artificial. Se integra con plataformas de análisis de datos de terceros.
Gestión de Dispositivos	Ofrece funciones integrales, como registro de dispositivos, actualizaciones de firmware y monitorización y control remotos. Los usuarios pueden gestionar los dispositivos desde una ubicación geográfica remota.
Soporte para Desarrollo de Aplicaciones	Proporciona herramientas y recursos, como API, SDK, modelos, un IDE basado en la nube, una biblioteca de código de muestra y herramientas integradas para ayudar al desarrollo de aplicaciones IoT.
Hardware Compatible	Admite una amplia gama de módulos hardware, incluidos sensores, actuadores, *gateways*, dispositivos de periferia (*edge*) y cualquier dispositivo que pueda conectarse a Internet incluyéndose también, hardware de terceros.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf ,Binary, Avro, CBOR, MessagePack, Thrift, Cap'n Proto e incluso formatos de serialización personalizados entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Sí, a través de la API, el kit de desarrollo o el panel de control.
Gestión de Usuarios y Roles	Proporciona un completo sistema de control de acceso basado en roles (*RBAC*) para gestionar a los usuarios y sus permisos a través del panel de control vía navegador web de la plataforma.
Requisitos No Funcionales	Ofrece alta disponibilidad, rendimiento y copia de seguridad de datos con un *SLA* de tiempo de actividad del 99,9%. Dispone de funciones de almacenamiento en caché y balanceo de carga.

	Las copias de seguridad pueden realizarse de manera periódica, guardando las mismas en varios proveedores de almacenamiento en la nube.
Dominios o Casos de Uso Respaldados	Se utiliza en sectores como la Fabricación, Sanidad, Energía, Transporte y Distribución, para recopilar datos, analizarlos y obtener información práctica.

IoTSens	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	Certificado UNE 178104. Reconocido por el gobierno español como plataforma referente en el ámbito del IoT.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C#, .NET, Android, iOS, Arduino, Sigfox, Ruby y Groovy entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, CoAP, MQTT/AMQP/HTTP sobre WebSockets, LoraWAN, WebSockets, ModBus, XMPP, LwM2M, NB-IoT, etc…
Almacenamiento de Datos	Compatible con almacenamiento en la nube, local e híbrido.
Escalabilidad	Plataforma horizontal y transversal capaz de evolucionar y ser ampliable gracias a la organización de cada una de sus capas.
Seguridad	Proporciona encriptación de datos, autenticación y autorización.
Facilidad de Uso	Diseñado para ser fácil de usar, con una interfaz sencilla y una amplia documentación y recursos de apoyo.
Precio	IoTsens no ofrece una tabla de precios y no facilita una estimación según una métrica definida. Se ha de contactar vía email para obtener información al respecto.
Análisis de Datos	Proporciona herramientas y funciones de análisis de datos como: visualización de datos, minería de datos, *machine learning* e inteligencia artificial.
Gestión de Dispositivos	Ofrece aprovisionamiento de dispositivos, supervisión y actualizaciones de firmware.
Soporte para Desarrollo de Aplicaciones	Ofrece un entorno de desarrollo, una biblioteca de componentes prefabricados y un espacio de compra para aplicaciones de terceros.
Hardware Compatible	Admite una amplia gama de dispositivos, incluidos sensores, actuadores y *gateways*.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf, YAML, Avro, CBOR, MessagePack, Cap'n Proto y FlatBuffers entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Proporciona un sistema de gestión de usuarios y funciones para crear y gestionar cuentas y roles de usuario.
Requisitos No Funcionales	Garantiza alta disponibilidad, funciones de rendimiento como balanceo de carga y almacenamiento en caché, y servicios de copia de seguridad y recuperación de datos.

Dominios o Casos de Uso Respaldados	Aplicable en Ciudades Inteligentes, Edificios Inteligentes, Automatización Industrial y Sanidad.

Apéndice VII. IoTSens

Kaa	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	ISO/IEC 27001, SOC 2, PCI DSS. Miembro certificado de IoTSF. Cumple con el NIST Cybersecurity Framework
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C#, Go, .NET, Android, iOS y Arduino entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, MQTT/AMQP/HTTP sobre WebSockets, LoraWAN, WebSockets, XMPP, LwM2M, Kaa Protocol (1/KP), STOMP, etc … Admite cualquier protocolo de trasmisión que hayas creado para emplearlo para comunicación añadiéndolo a la plataforma
Almacenamiento de Datos	Preintegrada con bases de datos listas para la producción, como Cassandra, MongoDB, InfluxDB y otras. Permite almacenamiento tanto en local como en la nube.
Escalabilidad	Con la arquitectura de microservicios IoT preparada por Kubernetes puede escalar infinitamente y manejar millones de dispositivos.
Seguridad	Ofrece seguridad de extremo a extremo con funciones como cifrado de datos, control de acceso basado en roles (*RBAC*) y autenticación de dispositivos.
Facilidad de Uso	Fácil de usar y amigable para desarrolladores.
Precio	Precios por suscripción: Nivel gratuito disponible, planes de precios en función de los dispositivos conectados: [1-5] 9,99$/mes, [6-15] 15,99$/mes, [16-100] 79,99 $/mes
Análisis de Datos	Proporciona análisis en tiempo real, análisis históricos, análisis *batch* y *machine learning*.
Gestión de Dispositivos	Ofrece actualizaciones de firmware y configuración, supervisión, registro y gestión remota de dispositivos.
Soporte para Desarrollo de Aplicaciones	Permite el desarrollo rápido de aplicaciones y es compatible con aplicaciones hechas a medida. Cuenta con un constructor de interfaz de usuario de "arrastrar y soltar".
Hardware Compatible	Compatible con una amplia gama de dispositivos de hardware como sensores, actuadores y *gateways*. Ej. de algunos de ellos: Raspberry Pi, Arduino o BeagleBone Black entre otros.
Formatos de Serialización Compatibles	Soporta: JSON, XML, Protobuf, Binario, YAML, Avro, CBOR, MessagePack e incluso formatos de serialización personalizados entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Gestiona usuarios y roles con control de acceso basado en roles (*RBAC*)

Requisitos No Funcionales	Garantiza un acuerdo de nivel de servicio (*SLA*) de alta disponibilidad, baja latencia y un tiempo de actividad del 99,9%. También ofrece copia de seguridad de datos.
Dominios o Casos de Uso Respaldados	Diseñado para poder trabajar en una amplia gama de casos de uso, como la Automatización Industrial, Ciudades Inteligentes, Sanidad, Logística, Distribución, Transporte, Seguimiento de Activos y Gestión de Flotas.

Apéndice VIII. Kaa

Macchina.IO EDGE	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	ISO/IEC 27001, PCI DSS, HIPAA.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, Android, iOS, Arduino y macchina.io REMOTE SDK entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, REST, HTTP, HTTPS, CoAP, MQTT/AMQP/HTTP sobre WebSockets, WebSockets, ModBus, OPC UA, SOAP, JSON-RPC, UPnP, CAN, CANopen, SNMP, etc …
Almacenamiento de Datos	Admite opciones de almacenamiento local, en la nube e híbrido, junto con almacenamiento de bases de datos como InfluxDB (DB diseñada para almacenar los datos de series temporales generados por cada dispositivo).
Escalabilidad	Alta escalabilidad, hasta diez mil dispositivos por instancia de servidor REMOTE de macchina.io (se pueden agrupar varios servidores (escalabilidad horizontal) para aumentar la capacidad hasta millones de dispositivos).
Seguridad	Proporciona un alto nivel de seguridad, incluido el cifrado TLS/SSL, el control de acceso basado en roles (*RBAC*) y medidas de prevención de pérdida de datos.
Facilidad de Uso	Ofrece una *GUI* (interfaz gráfica de usuario) fácil de usar, *CLI* (interfaz de línea de comandos) y documentación completa. Puede ser configurado por usuarios no técnicos.
Precio	Macchino.IO EDGE cuenta con 2 planes de precios, el primero, gratuito, el Open Source (GPL) está orientado para evaluar, experimentar y construir una prueba de concepto con la plataforma, mientras que el plan de pago Licencia Comercial, es para empresas que construyen dispositivos IoT profesionales.
Análisis de Datos	Proporciona un motor de análisis de datos integrado que permite: analizar datos de sensores y generar informes, visualizar datos y emplear *machine learning* e inteligencia artificial. Puede integrarse con plataformas de análisis de terceros.
Gestión de Dispositivos	Proporciona registro, configuración, gestión remota y supervisión de dispositivos y actualizaciones de firmware.
Soporte para Desarrollo de Aplicaciones	Admite el desarrollo rápido de aplicaciones con componentes y servicios preconstruidos, APIs, SDKs. Su elaborada documentación también es de gran ayuda.
Hardware Compatible	Admite una amplia gama de hardware, como: Raspberry Pi, BeagleBone, Intel Edison y otros dispositivos basados en Linux.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf, Binario, Avro y CBOR entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Proporciona un sistema de gestión de usuarios para crear y gestionar usuarios y roles.
Requisitos No Funcionales	Proporciona alta disponibilidad, con un SLA de tiempo de actividad del 99,9%. Puede gestionar grandes volúmenes de datos y realizar copias de seguridad. Se puede desplegar en un clúster para garantizar una alta disponibilidad.

Dominios o Casos de Uso Respaldados	Compatible con una amplia gama de dominios y casos de uso, como la Automatización Industrial, Ciudades Inteligentes, Sanidad, Distribución, Transporte, Fabricación, Logística y otros sectores.

Apéndice IX. Macchina.IO EDGE

Mainflux	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	ISO/IEC 27001, SOC 2, GDPR, HIPAA. Reconocido por la EU. Miembro de IIC & OIC.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C# y Go entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, CoAP, MQTT/AMQP/HTTP sobre WebSockets, LoraWAN, WebSockets, OPC UA, XMPP, LwM2M, MQTT-SN, STOMP, etc…
Almacenamiento de Datos	CassandraDB, MongoDB, InfluxDB, PostgreSQL
Escalabilidad	Escalable horizontalmente. Admite millones de dispositivos y es fácilmente escalable. Compatible con la implantación en la nube y en las instalaciones. Permite despliegue mediante Contenedores Docker.
Seguridad	Proporciona control de acceso basado en roles (*RBAC*), cifrado de datos, autenticación, autorización, conexiones PostgreSQL seguras, transporte seguro mediante certificados y seguridad de extremo a extremo mediante 2 modalidades de comunicación encriptada: - *Manager service* (proporciona autenticación comprobando la validez de los JSON Web Tokens (*JWTs*)). - *NGINX* (proxy inverso) (forma un cortafuegos, cerrando todas las rutas no públicas para el acceso externo).
Facilidad de Uso	Interfaz de usuario intuitiva y fácil de usar, API bien documentada. Proporciona herramientas y documentación para que los usuarios puedan empezar a utilizarla.
Precio	Ofrece planes de precios en los cuales los precios pueden variar, desde modos de instalación y planes de soporte absolutamente gratuitos hasta variantes empresariales personalizadas y totalmente gestionadas.
Análisis de Datos	visualización de datos y series temporales, análisis históricos y en tiempo real, *machine learning*, inteligencia artificial, motor de análisis integrado y permite la integración con herramientas de *Bussines Intelligence* (*BI*, inteligencia empresaria) de terceros.
Gestión de Dispositivos	Proporciona aprovisionamiento y supervisión de dispositivos, actualizaciones de firmware y gestión remota CRUD (Crear, Leer, Actualizar, Eliminar).
Soporte para Desarrollo de Aplicaciones	Admite una amplia gama de aplicaciones IoT, creador de interfaces de usuario, motor de reglas, sistema de notificaciones, herramientas de desarrollo de aplicaciones y modelos/plantillas.
Hardware Compatible	Admite una gran variedad de hardware IoT, incluidos sensores, actuadores y *gateways* de entre los que destacan: Raspberry Pi, Arduino, Intel Edison, AWS IoT Greengrass.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf, YAML, Avro, CBOR, MessagePack, Thrift y Pickle entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Admite políticas de control de acceso basadas en roles (*RBAC*) y control de acceso de granularidad fina (*ABAC*, *attribute-based access control*).

Requisitos No Funcionales	Ofrece alta disponibilidad y rendimiento con un SLA del 99,9%, 10.000 mensajes por segundo y copias de seguridad diarias.
Dominios o Casos de Uso Respaldados	Compatible con una amplia gama de ámbitos y casos de uso, como Fabricación, Logística, Sanidad, Ciudades inteligentes, Agricultura, Energía, Automatización industrial, Distribución y Transporte.

Apéndice X. Mainflux

Microsoft Azure IoT Central	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	Azure IoT Central, cumple las políticas y normativas de Microsoft, así como una serie de normas del sector, entre las que se incluyen ISO/IEC 27001, SOC 2, PCI DSS, HIPAA, GDPR, FedRAMP, FIPS 140-2. Certificado por CSA, NIST, IoTSF, Gartner, Forrester, International Data Corporation (IDC), Common Criteria for Information Technology Security Evaluation (CC).
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C#, .NET, Android, iOS e UWP entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, MQTT/AMQP/HTTP sobre WebSockets, WebSockets, ModBus, OPC UA, XMPP, LwM2M, Serial, etc…
Almacenamiento de Datos	Almacena los datos en la nube de Azure mediante Azure Blob Storage, lo que permite acceder a ellos desde cualquier lugar.
Escalabilidad	Es una plataforma altamente escalable, capaz de soportar millones de dispositivos, miles de millones de mensajes al día y manejar grandes volúmenes de datos. Se puede escalar tanto horizontal como verticalmente.
Seguridad	Azure IoT Central es una plataforma segura que ofrece una serie de funciones de seguridad, como, por ejemplo: autenticación de dispositivos, cifrado de datos, control de acceso basado en roles (*RBAC*), auditoría, protección de datos y seguridad de datos.
Facilidad de Uso	Azure IoT Central es una plataforma fácil de usar tanto para desarrolladores como para usuarios no técnicos. Simplifica la conexión de dispositivos, la gestión de datos y el desarrollo de aplicaciones con funciones como un editor de tipo "arrastrar y soltar" (ej. un editor web de *dashboards*), un motor de reglas y un simulador integrados y una interfaz gráfica de usuario (*GUI*).
Precio	Es un servicio basado en suscripción con un modelo de precios de pago por uso. El coste viene determinado por el número de dispositivos conectados, la cantidad de datos almacenados, el número de aplicaciones desplegadas, las funciones utilizadas y la cantidad de datos procesados.
Análisis de Datos	Proporciona múltiples capacidades de análisis de datos, como *machine learning*, inteligencia artificial, análisis en tiempo real, análisis histórico, análisis predictivo, análisis de *streaming* en tiempo real, visualización de datos en *dashboards* en tiempo real, almacenamiento de datos históricos y análisis *batch*.
Gestión de Dispositivos	Facilita diversas funciones de gestión de dispositivos, como el aprovisionamiento, el diagnóstico, la supervisión remota y las actualizaciones de firmware de dispositivos.
Soporte para Desarrollo de Aplicaciones	Ofrece un amplio conjunto de funciones de desarrollo de aplicaciones, como un mercado integrado, un portal para desarrolladores, diversos SDK y lenguajes, plantillas de aplicaciones, una interfaz de "arrastrar y soltar", un motor de reglas visual integrado, un editor de código integrado, la posibilidad de crear aplicaciones personalizadas e integrarse con aplicaciones existentes mediante conectores de aplicaciones.
Hardware Compatible	Admite diversos dispositivos de hardware, como sensores, actuadores y *gateways*.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf, Binario, YAML, Avro, MessagePack, Thrift, Parquet e incluso formatos de serialización personalizados entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Proporciona un sistema de control de acceso basado en roles (RBAC) que permite definir permisos para usuarios, grupos y perfiles. Admite funciones de gestión de usuarios y roles, como la creación de usuarios, la asignación de roles, el control de acceso y la capacidad de gestionar permisos de usuario para controlar el acceso a datos y aplicaciones de IoT.
Requisitos No Funcionales	Ofrece un SLA de alta disponibilidad y alto rendimiento del 99,9%, balanceo de carga, almacenamiento en caché y permite copias de seguridad de datos en Azure Blob Storage cada 15 minutos.

Dominios o Casos de Uso Respaldados	Azure IoT Central puede utilizarse en una amplia gama de dominios o casos de uso, como la Automatización Industrial, Ciudades Inteligentes, Seguimiento de Activos, Fabricación, Logística, Sanidad, Transporte y Energía. Se trata de una plataforma IoT de uso general.

Apéndice XI. Microsoft Azure IoT Central

Microsoft Azure IoT Hub	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	ISO/IEC 27001, HIPAA, FedRAMP, SOC 2, PCI DSS y FIPS 140-2. Está reconocido por organizaciones como Gartner, Forrester, IDC, IoT Security Alliance, IoT Security Foundation, Industrial Internet Consortium (IIC) y NIST Cybersecurity Framework.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C#, .NET, Android, iOS e UWP entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, MQTT/AMQP/HTTP sobre WebSockets, LoraWAN, WebSockets, ModBus, OPC UA, XMPP, LwM2M, STOMP, MQTT-SN, etc…
Almacenamiento de Datos	Ofrece opciones de almacenamiento de datos como Azure Storage (Azure Blob Storage y Azure Data Lake Storage entre otros), Azure SQL Database, Table Storage, Cosmos DB, almacenamiento en la nube, almacenamiento local, almacenamiento híbrido, políticas de retención de datos y purga de datos. También se integra con plataformas de almacenamiento de terceros. Usa enrutamiento de mensajes para enviar datos de telemetría desde los servicios de Azure de dispositivos IoT a Azure Storage
Escalabilidad	Altamente escalable. Admite millones de despliegues de IoT, desde pequeños a grandes, incluyendo la partición de dispositivos y el escalado automático. Se puede escalar horizontal o verticalmente.
Seguridad	Ofrece funciones de seguridad como autenticación, autorización, autenticación de dispositivos, cifrado de dispositivos, cifrado de datos y control de acceso basado en roles (*RBAC*).
Facilidad de Uso	Puede ser utilizado tanto por principiantes como por expertos. Proporciona una plataforma fácil de usar con herramientas, recursos, tutoriales, documentación y código de muestra para ayudar a los desarrolladores a gestionar los dispositivos y datos IoT.
Precio	Azure IoT Hub es un servicio de pago por suscripción. Los costes dependen del número de dispositivos conectados, los datos enviados y recibidos y el almacenamiento y funciones utilizadas.
Análisis de Datos	Ofrece funciones de análisis de datos, como análisis en tiempo real, análisis de *streaming* en tiempo real, análisis *batch*, *machine learning*, visualización de datos, inteligencia artificial, Power BI, Azure Data Studio, y se integra con plataformas de análisis de datos de terceros.
Gestión de Dispositivos	Ofrece funciones de gestión de dispositivos como: aprovisionamiento, registro, supervisión del estado y gestión centralizada de dispositivos, además de actualizaciones de firmware.
Soporte para Desarrollo de Aplicaciones	Proporciona funciones de desarrollo de aplicaciones, incluido un agente (*broker*) de mensajes integrado, una API REST, Azure Functions, Azure Stream Analytics, Azure Machine Learning, comandos, eventos y gemelos (*device twins*) de dispositivos, junto con plantillas y SDKs de aplicaciones.
Hardware Compatible	Compatible con hardware tales como sensores, actuadores, gateways, Raspberry Pi, Arduino, Intel Edison, Azure Stack, Azure IoT Edge. Asimismo, se puede integrar con hardware de terceros.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf, Binario, YAML, Avro, MessagePack, Thrift, Parquet e incluso formatos de serialización personalizados entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Sí, Azure IoT Hub permite enviar comandos a dispositivos utilizando la API de la plataforma, el kit de desarrollo, los SDKs de Azure IoT Hub o la API REST de Azure IoT Hub.
Gestión de Usuarios y Roles	Admite funciones completas de gestión de usuarios y roles. Esto incluye gestión de usuarios, de roles y de permisos, autenticación de usuarios, control de acceso basado en roles (*RBAC*) y aprovisionamiento de usuarios.
Requisitos No Funcionales	Azure IoT Hub cumple requisitos no funcionales como alta disponibilidad (con un SLA del 99,9%), rendimiento, balanceo de carga, almacenamiento en caché, copia de seguridad de datos en Azure Blob Storage y otros proveedores de almacenamiento en la nube, georreplicación y recuperación ante desastres. La plataforma

	puede escalarse para satisfacer las necesidades de aplicaciones exigentes, y se realizan copias de seguridad de los datos de forma automática y periódica para garantizar la disponibilidad en caso de fallo.
Dominios o Casos de Uso Respaldados	Azure IoT Hub se utiliza en una gran variedad de dominios y casos de uso, como la Automatización Industrial, Ciudades Inteligentes, Sanidad, Fabricación y Transporte.

Apéndice XII. Microsoft Azure IoT Hub

OpenRemote	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	ISO/IEC 27001, GDPR. Reconocido y certificado por Linux Foundation como uno de los 10 mejores proyectos Open Source y por la Open Source Initiative (OSI) como proyecto totalmente de código abierto.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C#, .NET, Ruby, Groovy. Las reglas del Motor de reglas se pueden escribir en Groovy, JavaScript, Rules JSON, or Flow model
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, MQTT/AMQP/HTTP sobre WebSockets, LoraWAN, WebSockets, LwM2M, Wake-On-Lan, X10, Velbus, KNX, Serial, SNMP, Telnet, AMX, Denon/Marantz, Domintell, DSC, Honeywell Z-Wave, Lutron, Russound, Samsung TV, etc….
Almacenamiento de Datos	Admite proveedores de almacenamiento en la nube (Amazon S3, Google Cloud Storage, Microsoft Azure), almacenamiento local y almacenamiento híbrido.
Escalabilidad	Plataforma escalable que puede utilizarse para gestionar un gran número de dispositivos, datos y usuarios. Admite escalado horizontal y vertical.
Seguridad	Utiliza medidas de seguridad para proteger los datos, como el cifrado, la autenticación y la autorización. La autenticación y autorización en el stack OpenRemote se realiza mediante Keycloak OpenID Connect Provider y utiliza OAuth 2.0. Gestión de identidades: Un activo es por defecto privado, sólo puede ser accedido por el superusuario o usuarios regulares de su dominio
Facilidad de Uso	Plataforma que ofrece una interfaz web (*GUI*) y una interfaz de línea de comandos (*CLI*). Adecuada tanto para desarrolladores como para no desarrolladores.
Precio	Plataforma gratuita y de código abierto, con opciones de pago para funciones y asistencia adicionales.
Análisis de Datos	Proporciona herramientas de análisis de datos como visualización de datos mediante *dashboards*, generación de informes, alertas, minería de datos, algoritmos de *machine learning* y admite la integración de herramientas de análisis de datos de terceros.
Gestión de Dispositivos	Proporciona funciones de gestión de dispositivos como seguimiento de activos, actualizaciones de firmware, solución remota de problemas y herramientas personalizadas de gestión de dispositivos, además de control remoto, detección, aprovisionamiento y supervisión de dispositivos.
Soporte para Desarrollo de Aplicaciones	Proporciona funciones de desarrollo de aplicaciones como: un creador de reglas de tipo "arrastrar y soltar", una API REST, funciones para la creación, implantación y supervisión de aplicaciones, automatización basada en reglas y cuadros de mando personalizados. Admite herramientas de desarrollo de aplicaciones personalizadas y de terceros.
Hardware Compatible	Admite diversos dispositivos de hardware, como sensores, actuadores, gateways, además de dispositivos hardware personalizados.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf, YAML, Avro, CBOR, MessagePack, Thrift, Cap'n Proto e incluso formatos de serialización personalizados entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Proporciona un sistema de gestión de usuarios y roles que permite crear usuarios, asignarles roles y permisos.

Requisitos No Funcionales	SLA de alta disponibilidad del 99,9% con un rendimiento máximo de 1 millón de mensajes por segundo. Ofrece opciones de backup de datos, con copias de seguridad diarias en Amazon S3.
Dominios o Casos de Uso Respaldados	Adecuado para la Automatización Industrial, Ciudades Inteligentes, Sanidad, Fabricación, Energía, Transporte, etc…

Apéndice XIII. OpenRemote

Predix Platform	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	ISO/IEC 27001, IEC 62443, HIPAA, SOC 2, PCI DSS. Reconocido por IIC, OPC, CSA, TUV SUD y British Standards Institution (BSI).
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C#, .NET, Android, iOS y Arduino entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, LoraWAN, WebSockets, ModBus, OPC UA, LwM2M, gRPC, Serial, SNMP, etc…
Almacenamiento de Datos	Ofrece distintas alternativas para el almacenamiento de datos: PostgreSQL, Cassandra, Amazon S3, Microsoft Azure, Google Cloud Platform, Google Cloud Storage, Microsoft Azure Blob Storage, etc….
Escalabilidad	Diseñado para ser escalable y poder gestionar millones de dispositivos, lo que lo hace adecuado para implantaciones de IoT a gran escala. A su vez, Predix ha demostrado que la ingesta de datos, el procesamiento analítico y la gestión de operaciones son capaces de adaptarse y escalar para satisfacer las exigentes cargas de trabajo de los líderes industriales.
Seguridad	Diseñado con funciones de seguridad integradas: autenticación, autorización, control de acceso basado en roles (*RBAC*), detección de intrusiones y transferencia cifrada de datos de extremo a nube.
Facilidad de Uso	Es una plataforma fácil de usar, que ofrece una interfaz sencilla, un entorno de desarrollo, un mercado de aplicaciones ya creadas, una comunidad de desarrolladores y diversas herramientas y recursos para ayudar a desarrolladores y usuarios finales a empezar a trabajar.
Precio	Predix Platform y los servicios individuales Predix solo se encuentran incluidos para su compra en el paquete Predix Essentials, del cual GE no ofrece una estimación o tabla de precios al respecto.
Análisis de Datos	Proporciona herramientas y servicios de análisis de datos: análisis de *streaming* en tiempo real, análisis *batch*, *machine learning*, análisis predictivo, detección de anomalías e inteligencia artificial orientada al ámbito empresarial.
Gestión de Dispositivos	Proporciona aprovisionamiento de dispositivos, actualizaciones de firmware, supervisión remota, detección de dispositivos, etc…
Soporte para Desarrollo de Aplicaciones	Ofrece herramientas y servicios como: un entorno de desarrollo, repositorio de código, entorno de testeo y apoya el desarrollo rápido de aplicaciones con componentes preconstruidos.
Hardware Compatible	Compatible con una amplia gama de dispositivos hardware, como sensores, actuadores y gateways, y admite diferentes módulos hardware, como Raspberry Pi, Intel Edison y Arduino entre otros.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf, Binario, Avro, CBOR, MessagePack, Thrift, Flatbuffers, Cap'n Proto y UBJSON entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	La plataforma Predix proporciona un sistema de control de acceso basado en roles (*RBAC*) para la gestión de usuarios y sus permisos, permitiendo a las organizaciones controlar el acceso a la plataforma y definir roles de usuario específicos.

Requisitos No Funcionales	Diseñado para ofrecer una alta disponibilidad, con un SLA del 99,9%. Ofrece funciones que garantizan un alto rendimiento, una gran fiabilidad, copias de seguridad y cifrado de datos.
Dominios o Casos de Uso Respaldados	Puede utilizarse para mejorar la eficacia operativa, reducir costes, aumentar la seguridad y optimizar procesos en diversos sectores como: Fabricación, Energía, Transporte, Sanidad, etc…

Apéndice XIV. Predix Platform

Sentilo	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	OIC, AllSeen Alliance (AllJoyn), OASIS Data Model for Sensor Networks (DMSN) & OASIS Sensor Markup Language (SenML). Bajo licencia Apache 2.0.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js y Sigfox entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, ModBus, OPC UA, XMPP, LwM2M, Serial, JMS, etc…
Almacenamiento de Datos	Compatible con MongoDB, MySQL, Cassandra, PostgreSQL, Elasticsearch y Memory BD.
Escalabilidad	Escalable horizontalmente para admitir un gran número de dispositivos y sensores. Admite desde servidores individuales a clústeres de hasta 10.000 nodos.
Seguridad	Admite diversas funciones de seguridad, como autenticación, autorización, cifrado de datos y control de acceso basado en roles (*RBAC*). La plataforma validará cualquier solicitud recibida por el sistema siguiendo la terminología *AAA, Authentication, Authorization & Accounting/Traceability* (Autenticación, Autorización y Trazabilidad).
Facilidad de Uso	Fácil de usar y gestionar, con una interfaz web (*GUI*) sencilla y una API bien documentada.
Precio	La plataforma Sentilo se encuentra a disposición del público de manera totalmente gratuita, ya que es un proyecto de Código abierto ofrecido por la *Diputació de Barcelona*. El gasto que conlleva esta solución depende del hosting escogido para el despliegue de la plataforma.
Análisis de Datos	Funciones integradas de análisis de datos, como agregación, visualización y correlación de datos, *machine learning* y detección de anomalías.
Gestión de Dispositivos	Admite el registro, configuración, supervisión, aprovisionamiento, actualización de firmware y control remoto de dispositivos.
Soporte para Desarrollo de Aplicaciones	Ofrece varios frameworks de desarrollo de aplicaciones, incluidos Node.js, Spring Boot y Django. Admite modelos de datos personalizados, reglas, alertas, plantillas y herramientas para el desarrollo y puesta en marcha.
Hardware Compatible	Compatible modelos hardware como Raspberry Pi, Arduino, Intel Edison, y dispositivos *gateways* en general.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf, Binario, YAML, Avro, MessagePack y Thrift entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Admite la gestión de usuarios y roles (*RBAC*), con capacidad para crear y gestionar usuarios, grupos, roles y permisos.

Requisitos No Funcionales	Ofrece una alta disponibilidad y rendimiento con un SLA del 99,9% de tiempo de operatividad. A su vez, también cuenta con: manejo de grandes volúmenes de datos y millones de mensajes por segundo y opciones de copia de seguridad (MongoDB, Cassandra, Elasticsearch) para proteger los datos en caso de fallo del sistema.
Dominios o Casos de Uso Respaldados	Adecuado para casos de uso de aplicación de IoT como Ciudades Inteligentes, Edificios Inteligentes, Automatización Industrial, Industria 4.0, Sanidad, Logística y Transporte.

Apéndice XV. Sentilo

Sofia2	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	ISO/IEC 27001, SOC 2, PCI DSS
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, C#, .NET, Android, iOS y Arduino entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, AMQP, REST, HTTP, HTTPS, CoAP, WebSockets, ModBus, OPC UA, LwM2M, Serial, JMS, etc...
Almacenamiento de Datos	La plataforma plantea el uso de tres repositorios distintos uno encargado de almacenar la información recibida en tiempo real (Base de Datos Tiempo Real (BDTR)), otro donde se almacena la información histórica para su posterior explotación analítica (Base de datos Histórica (BDH)) y, por último, uno donde almacenar información en diferentes estados (estructurada, semi-estructurada y no estructurada) temporalmente (Repositorio de *Staging* (HDFS))
Escalabilidad	Ofrece escalabilidad tanto horizontal como vertical.
Seguridad	Proporciona autenticación, autorización, control de acceso basado en roles (*RBAC*), copia de seguridad de los datos, seguridad en las comunicaciones, detección de intrusiones e integridad y cifrado de datos.
Facilidad de Uso	Interfaz sencilla y fácil de usar
Precio	Los planes de precios están ligados al modelo de despliegue por el que se desea optar siendo las opciones: despliegue en las instalaciones locales del cliente (PaaS), o desplegado en la nube (SaaS), dividiéndose esta opción en 2, despliegue gestionado, o no gestionado por Indra. También existe un modelo gratuito con el que experimentar y conocer lo que oferta la plataforma.
Análisis de Datos	Proporciona funciones tales como *machine learning* y análisis en tiempo real, histórico y predictivo.
Gestión de Dispositivos	Ofrece actualizaciones de *firmware* y registro, aprovisionamiento, supervisión remota y configuración de dispositivos.
Soporte para Desarrollo de Aplicaciones	Desarrollo rápido de aplicaciones IoT gracias a su generador de aplicaciones con interfaz de "arrastrar y soltar" y su compatibilidad con SDKs para lenguajes populares.
Hardware Compatible	Soporta una amplia gama de dispositivos IoT, tales como Raspberry Pi, Arduino o Intel Edison y numerosos sensores y actuadores del mercado.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf, Binario, YAML, Avro, MessagePack, Thrift, Flatbuffers, Cap'n Proto y ASN.1 entre otros
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Gestiona a los usuarios y roles mediante características como su control de acceso basado en roles (*RBAC*) encargado de la gestión de usuarios, roles y permisos.

Requisitos No Funcionales	Garantiza una alta disponibilidad (SLA del 99,9%), alto rendimiento y copia de seguridad de datos tanto en la nube como en las instalaciones propias del cliente.
Dominios o Casos de Uso Respaldados	Diseñado para poder trabajar en una amplia gama de casos de uso, como, por ejemplo, en Fabricación, Logística, Sanidad, Ciudades Inteligentes, Automatización Industrial o Distribución.

Apéndice XVI. Sofia2

Thinger.io	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	Certificado por AWS. Cumple con GDPR. Reconocido por IoTSF & CSA.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, Arduino, Sigfox y (Visual Studio Code con) PlatformIO entre otros.
Protocolos de Comunicación Compatibles	Soporta: MQTT, REST, HTTP, HTTPS, CoAP, ModBus, OPC UA, LwM2M, PSON, Serial, LPWAN, etc…
Almacenamiento de Datos	Almacena la información en la nube en soluciones seguras, eficientes y escalables como por ejemplo MongoDB, DynamoDB o AmazonS3. También emplea Databucket para almacenar datos de series temporales.
Escalabilidad	Altamente escalable. Puede albergar un gran número de dispositivos y usuarios.
Seguridad	Proporciona funciones como cifrado, autenticación y autorización.
Facilidad de Uso	Plataforma fácil de usar, tanto para desarrolladores como para no desarrolladores. Es muy sencillo trabajar con ella gracias a la gran documentación paso a paso que posee.
Precio	Oferta 3 planes: Premium, On-Premise y Thinger.io Cloud. Cada uno de ellos cuenta con diferentes características y capacidades de computación, las cuales se van incrementando/añadiendo conforme aumenta el precio del plan escogido. También existe una oferta gratuita enfocada para uso personal, o para estudiantes, el cual no cuenta con todas las funcionalidades disponibles.
Análisis de Datos	Permite el análisis de datos históricos y en tiempo real, incluyendo funciones de *machine learning*.
Gestión de Dispositivos	Incluye funciones como el registro, la configuración, la supervisión, el aprovisionamiento y la actualización de dispositivos.
Soporte para Desarrollo de Aplicaciones	Ofrece herramientas para el desarrollo de aplicaciones personalizadas, como, por ejemplo, un creador de cuadros de mando de tipo "arrastrar y soltar", una API REST o un broker MQTT.
Hardware Compatible	Compatible con Arduino, Raspberry Pi y dispositivos BLE, ESP8266.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV y Protobuf entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Permite la creación y gestión de usuarios, funciones y grupos.

Requisitos No Funcionales	Garantiza una alta disponibilidad (SLA del 99,9% de tiempo operativo) y proporciona funciones de alto rendimiento como balanceo de carga y almacenamiento en caché, y varios servicios de copia de seguridad de datos y recuperación en caso de catástrofe.
Dominios o Casos de Uso Respaldados	Adecuado para Ciudades Inteligentes, IoT industrial, Monitorización Medioambiental, Automatización Industrial, Sanidad y Fabricación entre otros casos de uso.

Apéndice XVII. Thinger.io

ThingSpeak	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	Certificado por Internet of Things Consortium, LoRa Alliance y CSA. Cumple con GDPR. Reconocido como un líder en el Mercado Global de Plataformas IIoT (Global IIoT Platform Market) por Frost & Sullivan.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++, JavaScript/Node.js, Arduino y MATLAB.
Protocolos de Comunicación Compatibles	Soporta: MQTT, REST, HTTP, HTTPS, CoAP, Microsoft Azure Event Hubs y Google Cloud Pub/Sub entre otros.
Almacenamiento de Datos	ThingSpeak trabaja sobre el concepto de “Canales”. Un canal es donde se almacenan los datos recibidos de dispositivos y aplicaciones de terceros
Escalabilidad	Plataforma escalable para proyectos de gran envergadura. Admite millones de dispositivos y puede gestionar grandes volúmenes de datos (hasta miles de millones de datos).
Seguridad	Utiliza cifrado de datos (SSL/TLS), autenticación de usuarios y control de acceso basado en roles (*RBAC*).
Facilidad de Uso	Fácil de usar para desarrolladores y no desarrolladores gracias a su sencilla interfaz.
Precio	ThingSpeak está disponible como servicio gratuito para pequeños proyectos no comerciales. Para proyectos de mayor envergadura o aplicaciones comerciales, se ofrecen cuatro tipos de licencias anuales diferentes: Estudiante, Hogar, Académico y Estándar. Cada una de ellas cuenta con diferentes características y capacidades de computación, que van incrementándose conforme aumenta el precio del plan escogido
Análisis de Datos	Proporciona visualización de datos, estadísticas, gráficos, cuadros de mando y alertas. También admite funciones de *machine learning* e inteligencia artificial.
Gestión de Dispositivos	Incluye registro de dispositivos, supervisión de estado, actualizaciones de firmware, configuración y control remoto.
Soporte para Desarrollo de Aplicaciones	Ofrece un creador de aplicaciones de tipo "arrastrar y soltar", una API REST, una API MATLAB y funciones de creación, despliegue y gestión de aplicaciones.
Hardware Compatible	Arduino, Raspberry Pi, BeagleBone Black, así como diversos sensores y actuadores.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV
¿La plataforma permite enviar comandos a los dispositivos?	Si, a través de REST API, MATLAB API o el creador de aplicaciones.
Gestión de Usuarios y Roles	Admite la gestión de usuarios y roles y el control de acceso basado en roles (*RBAC*).

Requisitos No Funcionales	Garantiza una alta disponibilidad (SLA del 99,9%). También permite gestionar grandes volúmenes de datos y ofrece servicios diarios de copia de seguridad y recuperación de datos.
Dominios o Casos de Uso Respaldados	Adecuado para su aplicación en Agricultura, Energía, Fabricación, Transporte, Ciudades Inteligentes, Automatización Industrial, Sanidad, Distribución, Logística y Monitorización Ambiental.

Apéndice XVIII. ThingSpeak

Zetta	
Característica	**Detalles**
Certificaciones y cumplimiento de la normativa	Cumple con IEEE 802.15.4, Zigbee, Z-Wave, OAuth 2.0, OpenID Connect y Transport Layer Security 1.2.
SDK & Lenguajes Soportados para Desarrollar	Soporta: Java, Python, C/C++ y JavaScript/Node.js.
Protocolos de Comunicación Compatibles	Soporta: MQTT, REST, HTTP, HTTPS, CoAP, LoraWAN, WebSockets, ModBus, OPC UA, LwM2M, Serial y DDS entre otros. Es agnóstico al protocolo de red, es decir, admite casi todos los protocolos de comunicación con los que trabajan los dispositivos.
Almacenamiento de Datos	Admite servidores locales, en la nube, almacenamiento en dispositivos periféricos y bases de datos en memoria, relacionales y NoSQL, incluidas MongoDB, MySQL y PostgreSQL.
Escalabilidad	Plataforma escalable. Puede gestionar un gran número de dispositivos, usuarios y datos.
Seguridad	Proporciona autenticación, autorización y cifrado y sigue estándares de seguridad como OAuth 2.0, OpenID Connect y TLS 1.2.
Facilidad de Uso	Plataforma fácil de usar. Fácil de configurar y utilizar tanto para principiantes como para usuarios experimentados.
Precio	IoT Zetta es una plataforma gratuita de código abierto, que cuenta con complementos opcionales de pago, como son la asistencia o la formación.
Análisis de Datos	Proporciona visualización de datos, minería de datos, *machine learning* y un motor de análisis de datos integrado.
Gestión de Dispositivos	Ofrece funciones como detección de dispositivos, aprovisionamiento, supervisión, actualizaciones de firmware y configuración de ajustes.
Soporte para Desarrollo de Aplicaciones	Proporciona una API enriquecida, un IDE integrado, integración de dispositivos, visualización de datos y un paquete de herramientas y librerías.
Hardware Compatible	Compatible con Raspberry Pi, BeagleBone Black, Intel Edison y Arduino entre otros.
Formatos de Serialización Compatibles	Soporta: JSON, XML, CSV, Protobuf, Binario, YAML, Avro y MessagePack entre otros.
¿La plataforma permite enviar comandos a los dispositivos?	Si
Gestión de Usuarios y Roles	Zetta permite la gestión de usuarios y roles. A los usuarios se les pueden asignar roles con diferentes permisos.
Requisitos No Funcionales	Garantiza un SLA de alta disponibilidad. Proporciona almacenamiento en caché con balanceo de carga y diversas opciones de copia de seguridad de datos en diferentes ubicaciones (servidores locales, almacenamiento en la nube y dispositivos periféricos).

Dominios o Casos de Uso Respaldados	Adecuado para Domótica, Automatización Industrial, Ciudades Inteligentes, Sanidad y Agricultura.

Apéndice XIX. Zetta

11. Anexos

11.1. Objetivos de Desarrollo Sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. **Fin de la pobreza.**				✓
ODS 2. **Hambre cero.**				✓
ODS 3. **Salud y bienestar.**	✓			
ODS 4. **Educación de calidad.**				✓
ODS 5. **Igualdad de género.**				✓
ODS 6. **Agua limpia y saneamiento.**				✓
ODS 7. **Energía asequible y no contaminante.**				✓
ODS 8. **Trabajo decente y crecimiento económico.**				✓
ODS 9. **Industria, innovación e infraestructuras.**	✓			
ODS 10. **Reducción de las desigualdades**.				✓
ODS 11. **Ciudades y comunidades sostenibles**.	✓			
ODS 12. **Producción y consumo responsables.**				✓
ODS 13. **Acción por el clima.**				✓
ODS 14. **Vida submarina.**				✓
ODS 15. **Vida de ecosistemas terrestres.**				✓
ODS 16. **Paz, justicia e instituciones sólidas.**				✓
ODS 17. **Alianzas para lograr objetivos.**				✓

A continuación, se detallan los ODS más directamente relacionados con este Trabajo Final de Máster:

ODS 3 - **Salud y bienestar**: Al orientar una solución IoT para la informatización de viviendas con el objetivo específico de atender a personas con diabetes, el trabajo apunta a mejorar la calidad de vida y el bienestar de este grupo poblacional. Las herramientas tecnológicas pueden facilitar la monitorización y control de la enfermedad, y así contribuir a una vida más sana.

ODS 9 - **Industria, innovación e infraestructura**: El análisis de soluciones y plataformas IoT contribuye al desarrollo de infraestructuras resilientes y promueve la industrialización inclusiva y sostenible. Además, este tipo de trabajos impulsa la innovación al explorar las mejores prácticas y herramientas disponibles para la creación de soluciones IoT.

ODS 11 - **Ciudades y comunidades sostenibles**: Al promover la creación de 'Smart houses' orientadas a necesidades específicas, se está contribuyendo a la generación de espacios urbanos más inclusivos, seguros y sostenibles. La IoT tiene el potencial de mejorar la calidad de vida en las ciudades al optimizar el uso de recursos.