



universidad
de león



**Escuela de Ingenierías
Industrial, Informática y Aeroespacial**

MÁSTER EN INGENIERÍA INDUSTRIAL

Trabajo de Fin de Máster

**CONFIGURACIÓN DE PLATAFORMA IIOT PARA
ADQUISICIÓN, ALMACENAMIENTO Y PROCESADO
AVANZADO DE DATOS INDUSTRIALES**

**IIOT PLATFORM CONFIGURATION FOR INDUSTRIAL
DATA ACQUISITION, STORAGE AND ADVANCED
PROCESSING**

Autor: Juan Ramón Cienfuegos-Jovellanos

Tutor: Miguel Ángel Prada Medrano

Cotutor: Serafín Alonso Castro

(septiembre, 2022)

Resumen

Desde hace unos años la sociedad está viviendo una revolución, en todos sus ámbitos hay dispositivos, técnicas o sistemas que generan grandes cantidades de información. Desde los datos que se obtienen en las ciudades relativos a la movilidad de los ciudadanos, datos biológicos de análisis clínicos en los hospitales o datos del comportamiento del tráfico aéreo comercial en los aeropuertos. La industria, como elemento importante en nuestra sociedad, también está sufriendo esta revolución. Las tecnologías, equipos y personas que la componen tienen la capacidad de generar más datos y de transmitirlos más allá de sus fronteras habituales. Estos conceptos se recogen en torno a lo que se denomina cuarta Revolución Industrial o Industria 4.0.

Si bien es cierto que hay una posibilidad mayor de transmisión y de generación de información, también surge la necesidad de desarrollar herramientas que permitan manejar estos recursos y sacar conclusiones que sean útiles para la mejora de los procesos y el desempeño global de la sociedad, organización o industria.

En el presente trabajo se propone la utilización de una plataforma IoT para el almacenamiento centralizado y la visualización remota y en vivo de datos provenientes de instalaciones industriales. Esta solución es capaz de interoperar con estos sistemas a través de protocolos de comunicación industrial (ej.: ModBus) permitiendo el almacenamiento organizado de la información deseada en un único punto. Esta característica hace posible la visualización simultánea de múltiples parámetros provenientes de los equipos, realizar una supervisión del estado de operación (ej: generación de alarmas) o el procesado de datos en el propio sistema. La plataforma está basada en tecnología web, ofreciendo sencillez y homogeneidad en su interacción con el usuario y, además, habilita interfaces que permiten la fácil obtención de los datos almacenados para, por ejemplo, ser procesados mediante técnicas avanzadas (Big Data / Inteligencia Artificial) y generar nuevas evidencias que ayuden en el diagnóstico de fallos o en la toma de decisiones de negocio.

Abstract

Nowadays, society is living a revolution. There are technics, devices or systems everywhere generating enormous amounts of information. From mobility data of people generated in cities, biological data generated from medical testing in hospitals or commercial air traffic patterns data in airports. Industry, as a key actor in society, is also experimenting that revolution. Industrial devices, facilities or people are generating day by day more and more information. These systems have also developed the ability to transmit that information far away beyond their traditional limits. These concepts are integrated in, what is called, the fourth Industrial Revolution or Industry 4.0.

Having a much greater capacity of transmitting and generating information, means creating new tools in order to deal with it. These new elements will need to be developed to handle great volumes of data and to extract new features from it, which will help to improve processes and global performance of society, organizations or industries.

The present project introduces a platform for the centralized storage and live and remote visualization of data originated at industrial facilities. This solution is capable of interoperate with industrial equipment through industrial communication protocols, such as ModBus, allowing an organized storage of information in one point, simultaneous visualization of data from different devices or facilities, operation status monitoring (e.g.: alarms generation) or data processing.

The platform is developed under web-based technology, offering simplicity and user interface standardization. In addition, it offers the possibility of easily obtaining stored information in order to perform an advanced data processing, generating new evidence which can help in fault diagnosis or in business decision making.

Índice de contenidos

Resumen.....	2
Abstract.....	3
Índice de contenidos	4
Índice de figuras	5
Glosario de signos, símbolos, unidades, abreviaturas, acrónimos o términos	7
1. Introducción.....	8
2. Estado del arte	11
3. ThingsBoard.....	24
4. Metodología y configuración.....	37
5. Resultados	48
6. Conclusiones.....	61
Bibliografía	64
Anexos	67

Índice de figuras

Figura 2.1. IoT Edge (Fuente: Azure [7]).....	13
Figura 2.2. AWS IoT Core (Fuente: AWS [9]).....	15
Figura 2.3. Arquitectura IBM Watson IoT (Fuente: IBM Watson [12])	18
Figura 3.1. Arquitectura interna ThingsBoard (Fuente: ThingsBoard [18])	27
Figura 3.2. Interfaz de usuario (Fuente: elaboración propia).....	31
Figura 3.3. Arquitectura IoT Gateway (Fuente: IoT Gateway [27])	34
Figura 4.1. Diagrama de red de instalación local (Fuente: elaboración propia) ...	38
Figura 4.2. Esquema de red instalación distribuida (Fuente: elaboración propia)	39
Figura 4.3. Conjunto de capturas de ModRSim2 (Fuente: elaboración propia / ModRSim2 [28])	45
Figura 4.4. Pantalla principal OPC UA Simulation Server (Fuente: elaboración propia / Prosys [29])	46
Figura 4.5. Captura OPC UA Simulation Server configuración objetos (Fuente: elaboración propia / Prosys [29]).....	47
Figura 5.1. Usuario creado (Fuente: elaboración propia).....	49
Figura 5.2. Dispositivos creados (Fuente: elaboración propia).....	50
Figura 5.3. ModRSim2 con valor ModBus a enviar a ThingsBoard (Fuente: elaboración propia).....	51
Figura 5.4. Valor ModBus recibido en plataforma ThingsBoard (Fuente: elaboración propia)	52
Figura 5.5. Valor OPC a enviar en OPC Simulation Server (Fuente: elaboración propia)	53
Figura 5.6. Valor OPC recibido en ThingsBoard (Fuente: elaboración propia)	53
Figura 5.7. Paneles de visualización de datos (Fuente: elaboración propia)	54
Figura 5.8. Detalle gráfico valores de temperatura (Fuente: elaboración propia) .	55
Figura 5.9. Detalle tabla valores temperatura (Fuente: elaboración propia).....	55
Figura 5.10. Panel de alarmas (Fuente: elaboración propia)	56
Figura 5.11. Detalle panel de alarmas (Fuente: elaboración propia).....	57
Figura 5.12. Registro alarmas (Fuente: elaboración propia)	57

Figura 5.13. Regla disparo de alarma alta temperatura (Fuente: elaboración propia)
..... 58

Figura 5.14. Valor promedio de temperatura (Fuente: elaboración propia) 59

Figura 5.15. Detalle valor promedio (Fuente: elaboración propia)..... 59

Glosario de signos, símbolos, unidades, abreviaturas, acrónimos o términos

IoT – Internet de las cosas - *Internet of Things*

IIoT – Internet de las cosas industrial - *Industrial Internet of Things*

PaaS – Plataforma como servicio - *Platform as a Service*

IA – Inteligencia Artificial

BI – Inteligencia de Empresarial – *Business Intelligence*

ERP – *Gestión de Recursos Empresariales - Enterprise Resources Planning*

M2M – Máquina a Máquina – *Machine To Machine*

IT – Tecnologías de la Información – *Information Technologies*

Topic – forma de organización de envíos e información en protocolos de comunicación del tipo MQTT o AMQP. Se puede entender como una cola que suministra la información de un publicador a todos los suscriptores que tenga

SO – Sistema Operativo - *Operating System*

RTOS – Sistema Operativo en Tiempo Real – *Real Time Operating System*

Gateway – Pasarela o puerta de enlace – permite la conexión de aparatos o redes mediante la traducción de la información que se utiliza en los extremos

Back-end – servidor de un página web encargado de gestionar la información que posteriormente es mostrada al usuario a través del interfaz web

1. Introducción

La sociedad en todo su conjunto y, en concreto, la industria está sufriendo una revolución. El aumento de la capacidad de procesado de los dispositivos, la mejora en las técnicas de análisis de la información, el progreso de las simulaciones o el incremento de velocidad y extensión de las redes de comunicación permiten que los datos se adquieran, procesen y fluyan sin barreras de manera instantánea en cualquier parte del mundo.

El término acuñado por el economista alemán Klaus Schwab, Cuarta Revolución Industrial o Industria 4.0, recoge la idea de la aparición de nuevas tecnologías y técnicas que incrementan la fusión entre los mundos físico, digital y biológico. Según sus palabras: “La Cuarta Revolución Industrial genera un mundo en el que los sistemas de fabricación virtuales y físicos cooperan entre sí de una manera flexible a nivel global. Sin embargo, no consiste solo en sistemas inteligentes y conectados. Su alcance es más amplio y va desde la secuenciación genética hasta la nanotecnología, y de las energías renovables a la computación cuántica. Es la fusión de estas tecnologías y su interacción a través de los dominios físicos, digitales y biológicos lo que hace que la Cuarta Revolución Industrial sea diferente a las anteriores” [1].

La aparición de tecnologías revolucionarias de amplio espectro no son condición suficiente para que supongan una nueva era en la industria. Una Revolución Industrial se considera que tiene lugar cuando se produce un cambio en la forma de pensar y de actuar derivado de las nuevas posibilidades que abren las tecnologías innovadoras.

La Industria 4.0 significa inmediatez y velocidad de acceso a los datos y ha permitido la aparición de las fábricas inteligentes o la gestión online de la producción. Así, la organización tradicional de la producción centrada solo en una instalación ha desaparecido, pasando a una mentalidad más global y distribuida donde las órdenes y el punto de acción pueden estar totalmente desacoplados geográficamente hablando. Estos cambios, si son aprovechados pueden suponer mejoras en las operaciones de negocio y el crecimiento de los ingresos,

transformando los productos, la cadena de suministro y las expectativas de los clientes.

Las empresas, organizaciones y personas tendrán que ser capaces de adaptarse a todos estos cambios para poder sacar el máximo partido de ellos, pero, sobre todo, para no desaparecer. La velocidad y el impacto con la que aparecen nuevas tecnologías es otra de las características que tiene esta revolución y la no asimilación puede suponer que la organización no sea competitiva en un mercado altamente cambiante, significando el fin de su actividad.

Algunas de las tecnologías implicadas en la Cuarta Revolución Industrial son la robótica, la analítica, la inteligencia artificial, las tecnologías cognitivas, la nanotecnología, el procesamiento en la nube y el Internet of Things (IoT), entre otras [2].

De entre estas tecnologías, el Internet de las Cosas (IoT – *Internet of Things*) es especialmente relevante para el desarrollo de este Trabajo Fin de Máster.

IoT es el término que engloba la capacidad de cualquier objeto físico (“cosa”) capaz de incorporar sensores, software y otras tecnologías con el fin de conectarse e intercambiar información con otros dispositivos a través de Internet [3]. Estos dispositivos pueden ser desde objetos comunes (electrodomésticos, *wearables*...) a herramientas más asociadas al ámbito industrial (todo tipo de sensores, robots, máquinas...).

Desde el punto de vista industrial, esta capacidad de obtención de información desde dispositivos integrados directamente a las redes de comunicación y, en concreto, a Internet es muy interesante [4]. Permite generar gran cantidad de datos fácilmente accesibles que posibilitan la aplicación de técnicas avanzadas utilizando herramientas como el *big data*, la computación en la nube o la inteligencia artificial. El objetivo principal de todo este proceso es la generación de evidencias, conclusiones y acciones que mejoren el desempeño global de las instalaciones y organizaciones.

Todos los conceptos presentados con anterioridad en este apartado, desde el cambio de paradigma de la producción hasta la aparición de nuevas tecnologías y técnicas tienen una cosa en común, los datos. Todos los dispositivos son ahora capaces de generar y consumir grandes cantidades de datos, siendo estos un recurso

muy útil para obtener información que ayude a la mejora de sistemas, procesos y organizaciones.

Desde nuestro punto de vista como ingenieros industriales, una de las muchas actividades en las que podemos colaborar en todo este proceso de cambio es en la creación o desarrollo de herramientas que ayuden a las empresas, organizaciones y personas a su adaptación a este nuevo paradigma.

A lo largo de este Trabajo Fin de Máster, se presenta la utilización de una plataforma IIoT como herramienta para el almacenamiento, procesado y visualización de datos provenientes de instalaciones industriales. La plataforma permitirá la centralización del almacenamiento y la monitorización y acceso remoto para la aplicación de técnicas avanzadas de procesamiento de datos.

1.1. OBJETIVOS

Los objetivos de este Trabajo Fin de Máster son:

- Desplegar y configurar una plataforma IoT.
- Simular dispositivos industriales que proporcionen datos en diferentes protocolos de comunicación de uso industrial.
- Demostrar la capacidad de la plataforma para trabajar en entorno de Industria 4.0.

Objetivos secundarios:

- Explorar la capacidad de visualización de información integrado en la plataforma.
- Visualizar de manera remota los datos insertados en la plataforma

2. Estado del arte

Ante la gran cantidad de datos que se generan en todos los ámbitos de la sociedad y la capacidad cada vez más importante de procesamiento de estos datos, ya sea por nuevas técnicas o tecnología más avanzada, son múltiples los fabricantes de componentes industriales, proveedores de servicios en la web o de software que tienen entre sus productos distintos tipos de plataformas y servicios que ofrecen capacidades de almacenamiento, procesado datos y/o visualización de información.

Con el objeto de conocer la situación del mercado con respecto a este tipo de software se han analizado una serie de fabricantes que ofrecen servicios y plataformas con características similares a lo buscado en este Trabajo Fin de Máster. A lo largo del presente apartado se realiza un análisis de algunos de los más destacados.

2.1. PLATAFORMAS

2.1.1. POWER BI (MICROSOFT)

Power BI [5] es una herramienta de análisis empresarial que pretende acercar los datos y la toma de decisiones de negocio. Proporciona servicios para conectarse a la fuente de datos, modelarlos y visualizarlos para generar respuestas a los problemas empresariales ayudado por la IA incorporada en la plataforma.

Como muchas de las herramientas de Microsoft está pensada para poder trabajar dentro de una gran organización, aprovechando el trabajo colaborativo y pudiendo compartir la información de manera sencilla entre todos los implicados.

Presenta más de 500 conectores a través de los cuáles se puede obtener información para cargar en la plataforma. La mayor parte de ellos están centrados en establecer comunicación con bases de datos o similares. Esto hace que esta plataforma esté más pensada para trabajar con información ya tratada y almacenada que para conectarse directamente a los puntos de origen de la

información en un entorno industrial, como puede ser el caso de sensores, robots o máquinas.

2.1.2. AZURE IOT (MICROSOFT)

Es un conjunto de servicios y plataforma que permiten la creación y desarrollo de soluciones en la nube para entornos IoT [6]. Los diferentes servicios y productos que se ofrecen son:

- **Azure IoT Hub:**
Aplicación que permite la conexión con dispositivos IoT estableciendo una comunicación segura entre los puntos. Tienen un back-end desplegado en la nube que permite la monitorización sencilla de todos los dispositivos conectados. Además, permite la conectividad con el resto de las plataformas/servicios Azure sin necesidad de hacer traducciones o transformaciones. Los protocolos con los que funciona son: HTTPS, AMQP, AMQP sobre WebSockets, MQTT y MQTT sobre WebSockets.
- **Azure IoT Central:**
Plataforma de aplicaciones como servicio (PaaS) para entornos IoT. Permite el procesamiento de la información procedente de los dispositivos de manera sencilla a través de su interfaz web y la gestión de estos dispositivos. Proporciona al usuario plantillas de dispositivos o aplicaciones que facilitan el trabajo con la plataforma. Los resultados del procesamiento de esta información se pueden exportar al exterior a través de API REST.
- **Azure Digital Twins:**
Plataforma como servicio (PaaS) que permite la creación de grafos gemelos basados en modelos digitales de entornos completos, que pueden ser edificios, fábricas, granjas, redes energéticas industriales, etc. Estos modelos digitales se pueden usar para realizar simulaciones y obtener información que mejore la organización. En el entorno IoT, permite crear un gemelo de la instalación a través de la conexión de los gemelos de los dispositivos que ofrece IoT Hub.

- Azure IoT Edge:

Componentes pensados para ser ejecutados como microservicios que permiten establecer contenedores estándar que faciliten la comunicación entre los dispositivos IoT y la aplicación IoT Hub. Consta de tres componentes: contenedores IoT Edge, entorno de ejecución e interfaz basada en la nube para la supervisión de los contenedores.

Una de las aplicaciones de este componente es facilitar la comunicación de dispositivos que utilizan protocolos de comunicación distintos a los de IoT Hub, mediante la traducción a nivel de protocolo y a nivel de identidad.

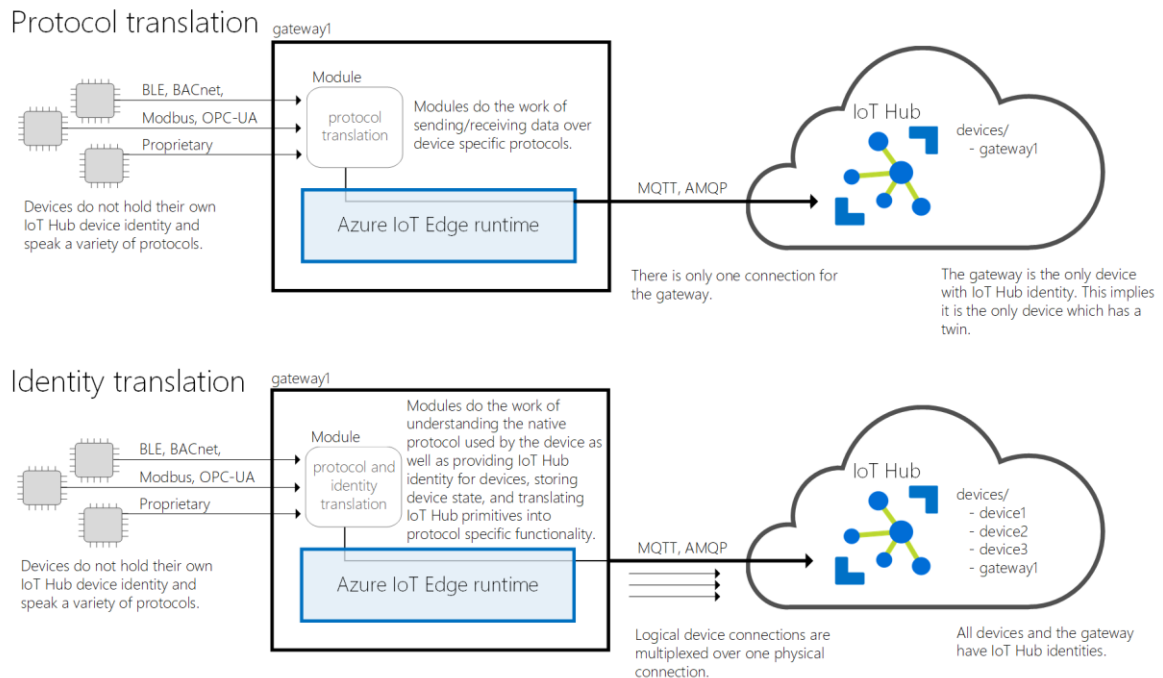


Figura 2.1. IoT Edge (Fuente: Azure [7])

- Azure Percept:

Conjunto de soluciones software, hardware y servicios que tiene por objetivo mejorar la transformación digital empresarial a través de tecnologías basadas en IoT e Inteligencia Artificial. A través de estas soluciones se intenta identificar el hardware necesario para la obtención de información de la organización, procesar la información de manera sencilla utilizando técnicas de IA y finalmente obtener conclusiones que ayuden en el desempeño global.

Es una solución completa para la empresa gestionada desde Azure.

- Azure Sphere:

Es una plataforma de comunicación para dispositivos IoT que asegura la comunicación segura entre los dispositivos y los servicios en la nube. Cuenta también con una parte hardware en forma de microcontrolador (MCU). Dentro de la arquitectura de instalación que propone Azure Sphere, este MCU será el que utilicen los dispositivos IoT y asegure la comunicación segura.

- Windows IoT:

Versión del Sistema Operativo Windows 10 pensada para dispositivos pequeños con o sin pantalla.

- Azure RTOS (Real Time Operating System):

Sistema Operativo en Tiempo Real pensado para dispositivos del entorno IoT que funcionen con microcontroladores. Está pensado para ser ejecutado en dispositivos con restricciones. Tienen características como: bajo consumo de potencia, ideal para dispositivos alimentados por batería; o bajo consumo de memoria, necesita menos de 64 KB de memoria flash.

Como se ha podido ver, la solución global que propone Azure es muy completa, de gran capacidad y con un soporte por parte de Microsoft. Se adapta a casi cualquier necesidad que se puede presentar en cada tipo de empresa. De hecho, algunos de los fabricantes que presentan plataformas IoT están asociados a Azure para el desarrollo de su software. Finalmente, todos los servicios de Azure, también los relacionados con IoT, requieren de la contratación de un plan para poder utilizarlos.

2.1.3. AWS IOT (AMAZON)

AWS [8] ofrece distintos servicios y soluciones IoT pensados para conectar y administrar dispositivos e integrar sus datos en el entorno AWS. Estos servicios son:

- FreeRTOS: sistema operativo para dispositivos pequeños y de poca potencia.
- AWS IoT GreenGrass: servicio para la creación, implementación y administración de software de dispositivos IoT.
- AWS IoT ExpressLink: módulos hardware para dispositivos IoT que faciliten la conexión con la plataforma de AWS.

- AWS IoT EduKit: servicio para la formación de estudiantes y profesionales de las opciones hardware y software que ofrece AWS IoT.
- AWS IoT Core: plataforma que permite la recepción de mensajes de dispositivos IoT y la transmisión de los datos proporcionados al resto de servicios AWS de manera sencilla. Por un lado, la comunicación entre los dispositivos y la plataforma está basada en MQTT y, por otro lado, la información se puede obtener de la plataforma a través de REST APIs. Ofrece la opción de integrar gateways para operar con otros protocolos, pero el desarrollo recae en el usuario del servicio.

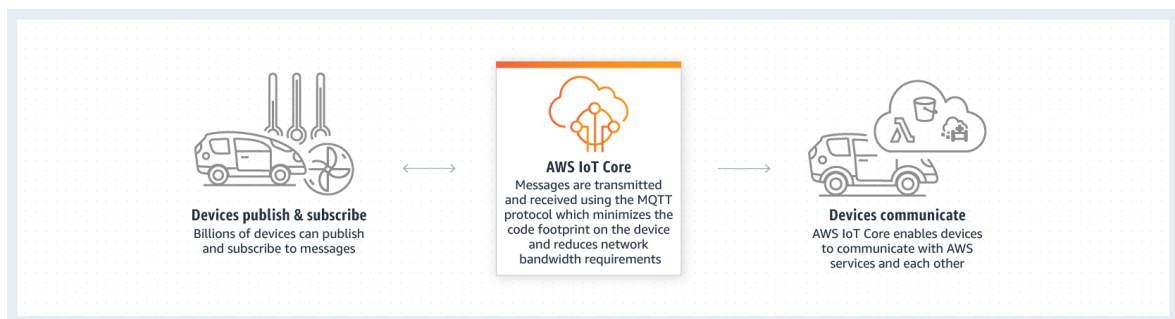


Figura 2.2. AWS IoT Core (Fuente: AWS [9])

- AWS IoT Device Defender: servicio que facilita la administración de la seguridad de los dispositivos IoT conectados a la plataforma. Facilita la auditoría de configuraciones, la autenticación de dispositivos, la detección de anomalías y la recepción de alertas.
- AWS IoT Device Management: servicio que ayuda en el registro, gestión, supervisión y administración de los dispositivos IoT de la red del cliente. Teniendo toda la información en la nube e incorporando los servicios de seguridad que ofrece AWS.
- AWS IoT FleetWise: solución pensada para fabricantes de automóviles. Permite la recolección de información casi instantánea de los datos de cada automóvil, haciendo que esté disponible en la nube en tiempo real. Así, esta información se puede utilizar para crear aplicaciones y realizar análisis que ayuden a mejorar la calidad, seguridad y autonomía de los automóviles.

AWS ofrece sus servicios de manera gratuita hasta cierto volumen de mensajes o actividad. A partir de esos valores hay que contratar un plan también basado en el volumen de información con el que se vaya a trabajar.

Como ocurre con Azure, AWS como proveedor de servicios de almacenamiento en la nube y similares, participa de manera indirecta en muchas de las plataformas de otros fabricantes ofreciendo soporte y servicios para el desarrollo.

2.1.4. FUSION CLOUD INTERNET OF THINGS (ORACLE)

Fusion Cloud IoT de Oracle [10] es un conjunto de aplicaciones principalmente alojadas en la nube que permiten la conexión de dispositivos para obtener más visibilidad, información y mejorar la eficiencia de la organización. Las soluciones que se ofrecen son:

- Internet of Things Cloud Service: Plataforma como Servicio (PaaS) alojada en la nube que permite gestionar la conexión de los dispositivos en tiempo real, extraer conclusiones de los datos recibidos e integrar esta información con otras aplicaciones. Las comunicaciones desde dispositivos están orientadas a tecnologías como REST API implementadas directamente desde Java, C o JavaScript.
- Asset Monitoring: facilita la integración de la información de todos los activos empresariales en la misma plataforma y generar analíticas en paneles que permitan realizar un seguimiento visual del estado.
- Production Monitoring: permite la recolección de datos de máquinas o instalaciones, para obtener un diagnóstico global o particular. Ofrece la utilización de técnicas de análisis basadas en el mantenimiento predictivo que generen recomendaciones y acciones a llevar a cabo.
- Fleet Monitoring: solución que permite monitorización de los datos de vehículos, conductores y rutas con el objetivo de conocer el estado de la flota o sus costes.
- Connected Worker: permite tener la información en tiempo real de trabajadores con el objetivo de cumplir con la normativa de seguridad y saber en todo momento dónde están los trabajadores y cómo es el entorno de trabajo.

En este caso, Oracle proporciona soluciones más centradas en determinadas facetas de negocio y no ofrece la flexibilidad de adaptación que tienen otras de las plataformas que se presentan.

Al igual que en casos anteriores, todos los servicios que se ofrecen desde Oracle requieren la contratación de un plan de suscripción.

2.1.5. IBM WATSON

IBM Watson [11] es un conjunto de aplicaciones, herramientas y soluciones (ej.: modelado visual de datos, catálogos de datos, etc.) preparadas para que las empresas reduzcan costes y obstáculos a la hora de implantar la IA. Además, tiene el objetivo de optimizar resultados y utilizar de manera responsable la IA:

IBM Watson también cuenta con una plataforma IoT alojada en la nube [12]. Cuenta con un interfaz de usuario en la web, posibilidad de gestionar los dispositivos conectados, comunicación basada en MQTT o HTTP o acceso a los datos desde el exterior usando API REST.

Resaltar que no ofrece soportes para otros protocolos y queda del lado del usuario realizar la traducción para poder acoplarlos.

Dispone de una versión ligera y sin necesidad de suscripción, pensada para comenzar a probar la plataforma y plantear una solución inicial. Las versiones de uso más profesional para entrar en producción conllevan la contratación de un plan de pago.

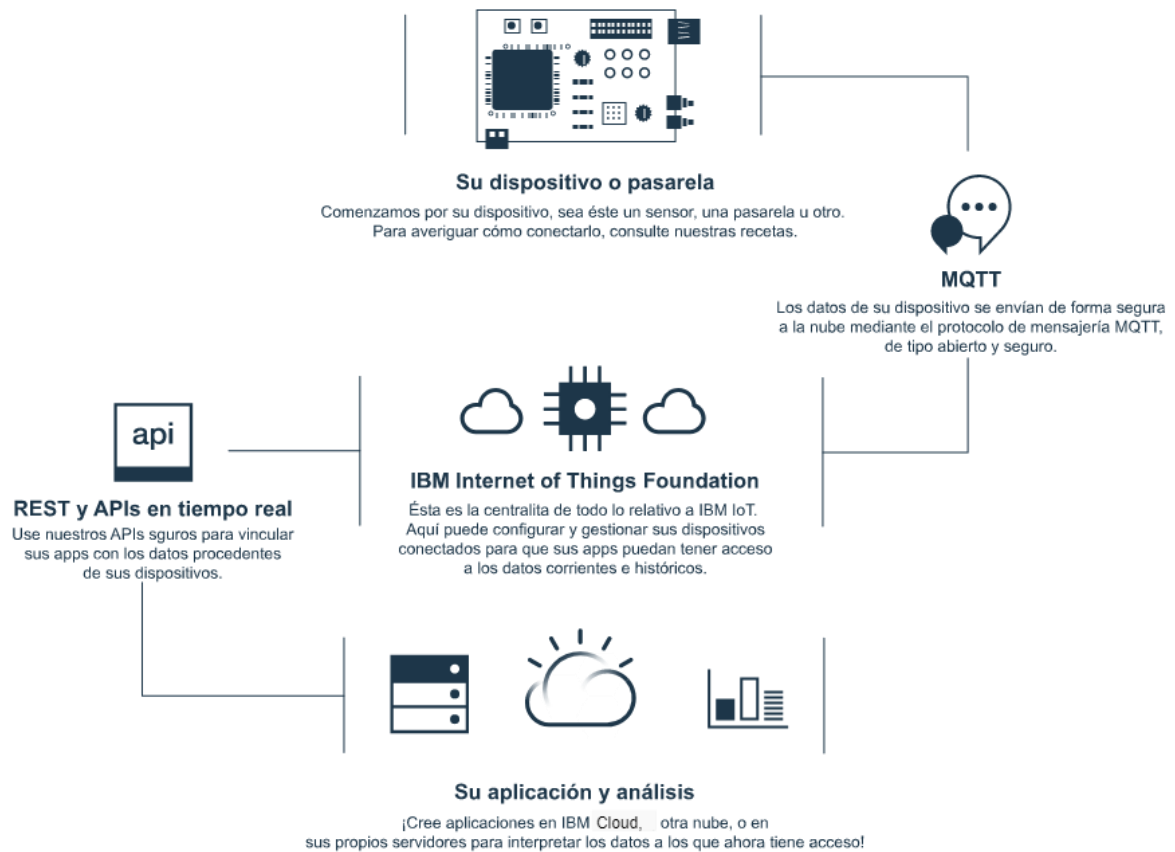


Figura 2.3. Arquitectura IBM Watson IoT (Fuente: IBM Watson [12])

2.1.6. MIND SPHERE (SIEMENS)

Mind Sphere [13] es el sistema operativo abierto IoT pensado para trabajar en la nube del fabricante Siemens. Permite conectar todos los dispositivos y sistemas, extraer los datos y procesarlos para obtener información valiosa.

Según el fabricante, las principales características de la plataforma son:

- Seguridad desde el proceso de diseño hasta el de producción.
- Amplias opciones para conectar dispositivos.
- Plataforma abierta como servicios (PaaS).
- Desarrollo rápido de soluciones.
- Estabilidad que supone estar dentro del entorno Siemens.

Desde el punto de vista de la conectividad, permite múltiples opciones muy pensadas para el ámbito industrial. Además, gracias a la experiencia del fabricante se han desarrollado componentes específicos para poder conectar sistemas *legacy*

En cuanto al procesado de datos, ofrece diferentes herramientas basadas en IA, *Machine Learning*, etc. Cabe destacar que es una plataforma que nace con la colaboración de Azure (Microsoft) y AWS (Amazon Web Services) con Siemens.

2.1.7. ECOSTRUXURE (SCHNEIDER)

EcoStruxure [14] es la plataforma compatible con arquitectura IoT de uso inmediato, abierto e interoperativo que ofrece el fabricante Schneider Electric. Pensada para trabajar en entornos domésticos, edificios, centro de datos, infraestructuras e industrias. Como en casos anteriores permite la conexión con elementos generadores de datos, para el posterior procesado y generación de información a altos niveles. Es una herramienta que permite la conexión vertical dentro de la organización, permitiendo que la información fluya desde el nivel de campo hasta el nivel más alto de gestión.

En el caso de esta plataforma, está pensada para ofrecer soluciones más cerradas dependiendo del sector involucrado. De esta manera el fabricante asesorará al cliente para ofrecerle la mejor solución posible.

2.1.8. ABB IOT GATEWAY

IoT Gateway de ABB [15] es un componente de pago para la mejora de la conectividad que permite unir el bus de campo de una fábrica con el sistema ERP o la nube. Basado en protocolos estándar de la industria, OPC-UA y MQTT entre otros. Está pensado para cerrar el hueco que hay entre el mundo IT, basado en tecnología IP, y el mundo de la producción con sus protocolos de comunicación particulares.

Los beneficios que el fabricante dice que aportan son:

- Datos estandarizados y seguros.
- Interoperabilidad entre dispositivos, máquinas e industrias.
- Comunicación integrada del sensor a la nube.
- Independiente del fabricante y de la plataforma.
- Permitir a los fabricantes el intercambio de información independiente.

2.1.9. ARDUINO CLOUD

Arduino Cloud [16] es un servicio que permite la conexión, programación y despliegue de dispositivos basados en Arduino. Está pensado para poder recibir y visualizar en paneles gráficos los datos provenientes de los dispositivos con el objetivo de monitorizarlos y controlarlos

Arduino Cloud solo funciona con las placas propias del fabricante o compatibles con este servicio (ESP32 y ESP8266). La comunicación placa-cloud se realiza mediante Wi-Fi, LoRaWAN@ y otras redes móviles.

Esta es una solución pensada para el entorno Arduino y como todo lo que ofrecen son soluciones de bajo coste (placas ofertadas) o servicios gratuitos (cloud).

2.1.10. ZETTA

Zetta [17] es una plataforma open-source basada en tecnología Node.js pensada para crear servidores IoT que se ejecuten en la nube o en servidores distribuidos en diferentes zonas geográficas. Esta solución incluye tecnologías modernas como REST APIs (comunicación con dispositivos y agentes externos), WebSockets (notificación de eventos) o programación reactiva.

Es una solución completamente gratuita y muy configurable. Sin embargo, su sencillez hace que requiera de muchos conocimientos de desarrollo software y de tecnologías complementarias por parte del usuario para poder ser desplegada con garantías.

2.1.11. THINGSBOARD

ThingsBoard [18] es una plataforma de código abierto pensada para la gestión de dispositivos IoT (Internet Of Things) y desarrollada para que tenga una puesta en servicio sencilla, fácilmente gestionable y permita escalabilidad.

Desde un interfaz de usuario web, permite el registro y gestión de los dispositivos, así como la visualización o el procesado de los datos recibidos y la monitorización de los dispositivos.

Tiene la opción de despliegue en la nube (opción de pago) o en servidores físicos (pago y gratuita). Los protocolos de comunicación soportados para el intercambio

de información con dispositivos son: MQTT, HTTP, CoAP, SNMP y LwM2M. Además, ofrece la opción de instalar un componente adicional que permite utilizar otros protocolos como: REST, BLE, HTTP(S), FTP, SNMP, ODBC, BACnet, OPC-UA o ModBus.

También ofrece la opción de acceder a la información desde aplicaciones externas mediante un interfaz API REST.

2.2. PROTOCOLOS DE COMUNICACIÓN

Como se ha visto, cada una de las plataformas analizadas presenta una serie de características propias, siendo una de ellas los interfaces de comunicación que ofrecen para interactuar con dispositivos o sistemas.

A continuación, se hace una pequeña descripción de algunos de los protocolos de comunicación involucrados en estos interfaces:

- AMQP (Advanced Message Queueing Protocol) [19]
Protocolo abierto en la capa de aplicación de un sistema de comunicación. Se caracteriza por: estar orientado a los mensajes, encolamiento, enrutamiento, seguridad y exactitud. El protocolo establece las características tanto del servidor que gestiona los mensajes como de los clientes de mensajería, permitiendo la interoperación entre componentes que utilizan tecnologías de diferentes operadores.
Ofrece seguridad a través de SASL y/o TLS y va sobre TCP.
- MQTT (Message Queueing Telemetry Transport) [20]
Protocolo de red ligero de tipo publicación/suscripción que transporta mensajes entre dispositivos, suele ir sobre TCP/IP. Está pensado para lugares con restricciones de acceso a la red donde el ancho de banda disponible es limitado. Otras de sus características son:
 - Dispone de un bróker encargado de gestionar las conexiones y la organización de la información al que se conectan clientes (dispositivos) para transmitir/recibir datos. La información es estructurada en forma de *topics*.

- El tamaño del mensaje está entre 2 bytes y 256 MB. El protocolo de transporte suele ser TCP, pero puede funcionar sobre UDP o Bluetooth.
 - No lleva seguridad intrínseca, pero se puede incorporar TLS.
 - Puerto por defecto 1883 y encriptado 8883.
- CoAP (Constrained Application Protocol) [21]
Protocolo de aplicación de Internet para dispositivos con restricciones (baja potencia, sensores inalámbricos...). Dentro de este protocolo a los dispositivos se les identifica como "nodos". Los nodos se comunican a través de Internet en redes con restricciones. Algunas características son:
 - Fácilmente traducible a HTTP.
 - Protocolo de la capa de servicio.
 - Soporta multicast- Se caracteriza por ser multicast, muy "low overhead" y simple.
 - Puede utilizar mecanismos como SMS.
 - HTTPS [22]
Protocolo de la capa de aplicación para la transmisión de textos de hipertexto como HTML. Diseñado para la comunicación entre los navegadores y los servidores web, se basa en un modelo cliente-servidor. Suele correr sobre TCP, aunque puede implementarse sobre cualquier capa de transporte siempre que sea de confianza.
 - ModBus TCP [23]
Protocolo de comunicación [24] en los niveles 1 (capa física), 2 (capa de enlace de datos) y 7 (capa de aplicación) del modelo OSI. Actualmente es un estándar de comunicaciones en la industria. Algunas de sus características son:
 - Diseñado para aplicaciones industriales
 - Público y gratuito
 - Fácil implementación
 - Usa bloques de datos sin suponer restricciones
- Cada dispositivo en una red ModBus tiene una dirección asignada y todas las instrucciones que tengan como destino este dispositivo contendrán su

dirección. Es un protocolo broadcast, todos los dispositivos reciben la información, pero solo el destinatario la interpreta.

Cuando corre sobre TCP, el puerto por defecto es el 502.

- OPC-UA [25]

Es un modelo de intercambio de datos multiplataforma, abierto, orientado a servicios y seguro, pensado para las comunicaciones industriales a todos los niveles. Sigue el estándar IEC62541 [26].

La comunicación se realiza a través de una arquitectura cliente-servidor, siendo el servidor el generador de los datos. El estándar establece modelos de datos estandarizados para facilitar la comunicación (alarmas, históricos o acceso a datos).

2.3. PLATAFORMA SELECCIONADA

Atendiendo a los objetivos de este Trabajo Fin de Máster, a las capacidades de las que se dispone y las características de las plataformas analizadas, la plataforma seleccionada es ThingsBoard. Las razones que justifican esta selección son:

- Soporte de protocolos de comunicación industrial como ModBus o OPC-UA.
- Fácil conexión de dispositivos industriales.
- Visualización online de los datos recibidos.
- Almacenamiento de la información.
- Posibilidad de realizar procesamiento de datos.
- Posibilidad de ampliar los protocolos de comunicación soportados.
- Acceso a la información almacenada en la plataforma desde aplicaciones externas.
- Instalación sencilla y asequible.
- Plataforma única. Ofrece prácticamente todos sus servicios desde la propia plataforma sin necesidad de configurar elementos extras.
- Open-source (accesible y gratuita).

3. ThingsBoard

Este apartado incluye la explicación detallada de qué es la plataforma ThingsBoard [18], qué opciones presenta para los diferentes tipos de usuario, cuáles son sus características principales, cómo funciona internamente, el procesamiento de datos que permite o las formas de interoperar con sistemas externos.

3.1. QUÉ ES THINGSBOARD

ThingsBoard es una plataforma de código abierto pensada para la gestión de dispositivos IoT (Internet Of Things) y desarrollada para que tenga una puesta en servicio sencilla, fácilmente gestionable y permita escalabilidad. Los dispositivos IoT se caracterizan por disponer de software de procesamiento y comunicación de información, ya sea a través de Internet o de cualquier otro protocolo de comunicación. Si bien es cierto que el término IoT se asocia a dispositivos más bien pequeños o simples, se pueden considerar dentro de este término todo dispositivo con capacidad de transmitir información a través de una red de comunicación (robots industriales, contadores de electricidad, pequeños sensores, etc.). De esta manera, un aparato, ya sea un teléfono móvil, sensor o equipo industrial, puede ser conectado para introducir información en la plataforma.

Las principales funciones por las que destaca ThingsBoard son:

- Gestión de dispositivos, activos y clientes, y definición de la relación entre ellos.
- Recolección y visualización de datos.
- Análisis de la información entrante.
- Generación de alarmas.
- Control de dispositivos remotos.
- Paneles de visualización de información dinámicos.
- Redireccionamiento de información hacia otros dispositivos o sistemas.
- ...

ThingsBoard está formado por un núcleo único con un comportamiento característico. Sin embargo, se presenta a los usuarios en tres formas distintas dependiendo la finalidad última que se le quiera dar a la plataforma.

Las tres opciones son:

- Thingsboard Community Edition: solución open-source gratuita y colaborativa. Requiere que se despliegue en equipos propios.
- ThingsBoard Professional Edition: solución profesional, cuenta con un soporte más completo, mayor sencillez de configuración y funcionalidad extendida. Está pensada para ser usada por empresas y permite la personalización con la imagen corporativa en los interfaces de usuario. Al igual que la opción comunidad necesita ser desplegada en equipos propios.
- ThingsBoard Cloud: similar a la opción profesional, ofreciendo funcionalidad extendida, soporte y capacidad de personalización. La diferencia reside en el despliegue de la plataforma, en este caso se despliega en la nube. Esta característica permite al usuario reducir costes, mejorar la disponibilidad, la persistencia de los datos y reducir los tiempos de puesta en servicio.

3.2. CONCEPTOS CLAVE

Para entender el funcionamiento de la plataforma ThingsBoard, hay que tener claros una serie de conceptos específicos en torno a los que se desarrolla la filosofía de trabajo con la plataforma. Estos conceptos son:

- *Tenants*: entendidos como unidades de negocio independiente. Es un sujeto u organización que produce o tiene dispositivos generadores de información. Este tipo de usuario puede configurar dispositivos dentro de la plataforma para que sean después consultados y utilizados por los *users/customers*.
- *Customers*: es otra unidad de negocio que compra o utiliza los dispositivos
- *Users*: son entidades que pueden consultar los *dashboards* en los que se muestran los datos dentro de la plataforma. Un *customer* puede tener múltiples *users*.
- *Devices*: dispositivos o sistemas que generan la telemetría que se introduce en la plataforma.

- *Assets*: entidades abstractas en torno a las que se organizan los *devices*. Por ejemplo: fábricas, compañías, vehículos...
- *Alarms*: eventos identificados como importantes a partir de los cuales se genera una notificación para el usuario.
- *Dashboard*: gráfico en el que se puede visualizar la información de los dispositivos.
- *Rule Node*: cada una de las unidades de procesamiento de la información de entrada. ThingsBoard tiene la capacidad de instalarse de manera clusterizada mejorando así su tasa de servicio.
- *Telemetry*: telemetría o datos periódicos generados por los dispositivos.
- *Attributes*: datos generados por los dispositivos relativos a alguna característica o configuración propia. No suelen mandarse periódicamente, solo cuando hay algún cambio en el funcionamiento.
- *Rule Chain*: define el flujo de la información dentro del *Rule Engine* (detallado en el siguiente subapartado).

3.3. ARQUITECTURA

Independientemente del tipo de instalación que se haya elegido, ThingsBoard tiene una arquitectura diseñada para que la plataforma sea:

- Escalable: el uso de tecnologías de código abierto líderes, facilitan la escalabilidad horizontal de la plataforma.
- A prueba de errores.
- Eficiente y robusta:
- Personalizable: permite añadir nueva funcionalidad a través de widgets o el motor de reglas.
- Duradera: almacena los datos para que estén siempre disponibles.

La estructura interna (Figura 3.1) se organiza, de manera simplificada, en torno al núcleo de la plataforma, el motor de reglas, las bases de datos y los interfaces. A continuación, se hace una pequeña descripción de cada una de las partes:

- ThingsBoard Core: desarrollado en tecnología java, es el responsable de gestionar las llamadas API REST, las subscripciones por WebSockets para

enviar información de telemetría o atributos de los dispositivos, procesar los mensajes a través del Rule Engine y monitorizar el estado de conexión de los dispositivos conectados a la plataforma. Es el elemento más importante dentro del sistema, ya que gestiona todas las demás partes.

- Rule Engine: permite realizar un procesado de la información. Su metodología está enfocada a eventos. La mejor manera de entender su funcionamiento es ver ejemplos de uso:
 - Validación o modificación de la telemetría de entrada antes de ser almacenada en la base de datos.
 - Agrupar los datos provenientes de diferentes dispositivos en torno a un mismo activo.
 - Crear/actualizar/borrar alarmas en función de unas condiciones determinadas.
 - Disparar llamadas API REST.
 - Integrarse con fuentes de información externas como AWS services, Kafka, etc.

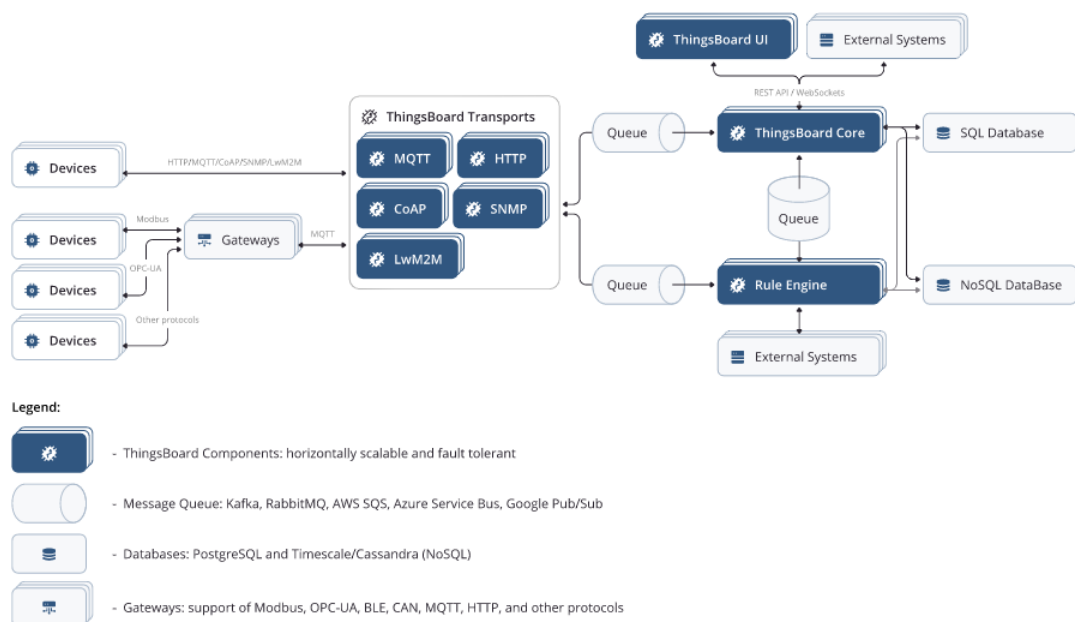


Figura 3.1. Arquitectura interna ThingsBoard (Fuente: ThingsBoard [18])

- Interfaces de entrada de datos: permiten la comunicación de la plataforma con los diferentes dispositivos generadores de datos y a los que se puede mandar órdenes. Estos interfaces pueden dividirse en tres grupos:
 - Interfaces nativos: soportan comunicación directa a la plataforma a través de protocolos de comunicación IoT (MQTT, HTTP, CoAP, SNMP y LwM2M).
 - Gateway: extiende las opciones de conectividad a través de un componente aparte. Es interesante para este trabajo, pues habilita la opción de utilizar protocolos de comunicación industrial. En el subapartado 3.8 se amplía la información de este componente.
 - Integración con otras plataformas: a través del Rule Engine se ofrece la posibilidad de integrarse con otras plataformas similares a ThingsBoard. Algunas de estas plataformas se han analizado en el apartado 2.1 (Azure IoT, AWS services, IBM Watson...).
- Interfaz de usuario: interfaz web desarrollado en JavaScript. Permite la interacción de usuario con la plataforma. Es poco configurable para permitir mayor estabilidad y sencillez de uso.
- Interfaz de acceso a los datos: permite acceder a los datos almacenados en la base de datos a través de la subscripción a WebSockets o mediante petición por API REST. Esta opción puede ser interesante para descargar los datos de la plataforma y aplicar análisis más allá de las opciones que ofrece ThingsBoard.
- Bases de datos: utilizadas para el almacenamiento de las entidades (*customers, users, devices...*) y de datos (telemetría y atributos generados por los dispositivos). Hay dos opciones de configuración de la base de datos:
 - Base de datos relacional: todos los datos (entidades y telemetría/atributos) se almacenan en una base de datos tipo SQL, en concreto los desarrolladores de la plataforma recomiendan el uso de PostgreSQL.
 - Híbrida: combina el uso de una base de datos relacional para la información de las entidades y una base de datos no relacional

para la telemetría y atributos. La base de datos no relacional aconsejada en Cassandra.

3.4. PROCESADO DE INFORMACIÓN

En apartados anteriores, se ha mencionado la capacidad de la plataforma para procesar la información. En concreto, se ha descrito una parte encargada de realizar esta tarea, *Rule Engine*. En este subapartado se quiere dar una explicación más extensa de las opciones que ofrece esta potente característica.

El tratamiento de la información que realiza la plataforma puede considerarse un preprocesado. Esto es debido a que la manera más habitual de activar este mecanismo es asociándolo a un evento de entrada de información. Así, cuando un nuevo valor de telemetría o de atributo es recibido desde un dispositivo, se pasa por el algoritmo asociado a ese dispositivo. El resultado contendrá el valor original modificado, el mismo o podrá contener más o menos campos y este valor o valores serán los que estén disponibles para generar pantallas de visualización y se almacenarán en la base de datos.

A continuación, se describen los grupos de acciones de procesado que hay y alguna de las acciones que hay dentro de ellos:

- Filtro: permiten filtrar el evento de entrada en función de distintos parámetros: tipo de mensaje, estado de alarmas, localización GPS... De esta manera se pueden descartar mensajes o enviarlos a distintos destinos en función de sus características.
- Enriquecimiento: añade más información que puede resultar útil más adelante en el ciclo de vida del mensaje.
- Transformación: permiten transformar el mensaje original añadiendo, quitando o transformando la información recibida.
- Acción: desencadenan una acción cada vez que les llega un evento. Estas acciones pueden ser alarmas, borrado de valores, contado de mensajes, guardado de la información, etc.
- Externo: permite traducir la información desde plataformas externas a ThingsBoard. Con estas acciones, se puede enviar información a la

plataforma hacia AWS services, Kafka, Azure IoT, etc. O enviar mensajes SMS o correos electrónicos. Algunas de las transformaciones que se ofrecen solo están disponibles desde la versión profesional.

- Flow: permite redireccionar los mensajes hacia otros puntos/componentes de la plataforma.

Las opciones que permite es componente son muy amplias y potentes. Bien es cierto que su configuración puede resultar pesada y no demasiado intuitiva.

3.5. INTERFAZ DE USUARIO

La manera principal que tiene el usuario de interactuar con ThingsBoard es a través de su interfaz web. En él se pueden configurar dispositivos, reglas de filtrado/validación, pantallas de visualización de datos o alarmas, etc.

Para acceder a la plataforma se dispone de un usuario y contraseña que serán generados a partir de la acción del administrador del sistema. Además de los usuarios, se pueden configurar diferentes perfiles que permiten clasificar a los usuarios en distintos grupos y así seleccionar cuáles son las acciones que pueden realizar y los datos a los que pueden acceder.

En la Figura 3.2, se puede apreciar como es este interfaz. Tiene una disposición muy genérica, teniendo una parte superior de configuraciones generales (ej.: gestiones de usuario), en la parte izquierda dispone un panel de selección de la función a mostrar y en la parte central muestra la información que se ha seleccionado para ver.

En el ejemplo de esta captura de pantalla, se puede ver distintos *dashboards* mostrando datos generados a partir de la simulación de un sensor de temperatura. En este ejemplo se ve un gráfico que muestra la evolución de la magnitud durante la última hora, los valores que se han ido recibiendo y las alarmas que se han disparado.

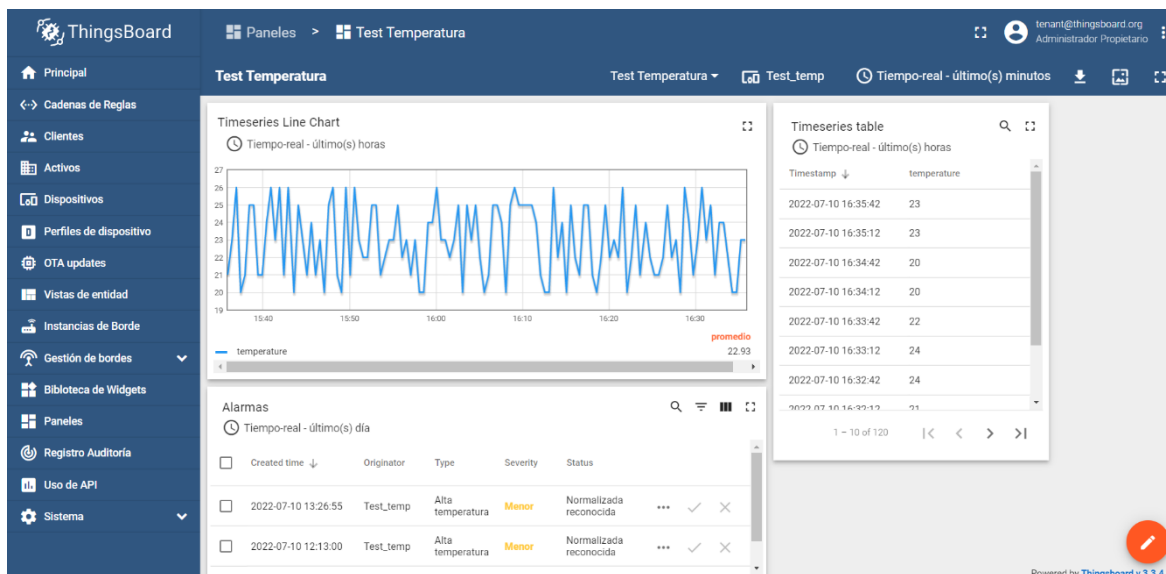


Figura 3.2. Interfaz de usuario (Fuente: elaboración propia)

ThingsBoard tiene multitud de pantallas de visualización de datos, más allá de las que aquí se ven. Además, ofrece la posibilidad de poder diseñar las pantallas que se desee dando así mayor posibilidad de personalización.

3.6. REQUISITOS DE INSTALACIÓN

ThingsBoard establece unos requisitos mínimos del sistema para que pueda ser instalado. Si bien es cierto que para la instalación más sencilla no son exigentes, hay que tenerlos en cuenta. Estos requisitos son:

- 1 GB RAM. Si se utiliza la instalación con bases de datos híbrida (requiere instalación de Cassandra), 8 GB RAM.
- SO basado en Linux de los indicados en el próximo apartado.
- Tener instalado Java 11.

3.7. OPCIONES DE INSTALACIÓN

ThingsBoard ofrece diferentes opciones desde el punto de vista de instalación. A continuación, se presentan las diferentes

- Distribución Linux:
 - Contenedor en Windows (Docker)
 - Contenedor en IOS/Linux (Docker)
 - Ubuntu 18.04 LTS o 20.04 LTS
 - Raspian
 - RHEL/CentOS
- Base de datos:
 - PostgreSQL
 - Híbrida (PostgreSQL + Cassandra)
 - Híbrida II (PostgreSQL+ TimescaleDB)
- Servicio de encolado interno:
 - En memoria
 - Kafka
 - Kafka en Docker
 - AQS SQS
 - Google Pub/Sub
 - Azure Service Bus
 - RabbitMQ
 - Confluent Cloud

3.8. IOT GATEWAY

Como se ha visto en apartados anteriores, una de las ventajas que presenta ThingsBoard y por la que ha sido elegida para este trabajo, es la posibilidad de establecer conexiones a través de protocolos de comunicación industriales como ModBus o protocolos de automatización en edificios como BACnet. Si bien es cierto que ThingsBoard no ofrece esta característica directamente, si ofrece la opción de instalar un componente adicional que hará las funciones de “adaptador”.

A lo largo de este apartado se explica qué es el IoT Gateway de ThingsBoard, cuáles son sus características y la configuración seleccionada para este trabajo.

IoT Gateway, al igual que ThingsBoard, es una solución de código abierto pensada para ser ejecutada en sistemas Linux, pero basada en tecnología Python 3.7+. Permite la conexión de sistemas *legacy* o de terceros que operen con protocolos de comunicación, distintos a los que soporta ThingsBoard, de manera sencilla y perfectamente alineada con la plataforma. Además, ofrece ciertas características para ofrecer un servicio de calidad. Estas son:

- Persistencia de los datos recogidos mediante un almacenamiento propio que los conservará hasta su envío. Esto es importante para no perder la información en caso de fallo en la red de comunicación con la plataforma o fallo en el hardware del sistema.
- Reconexión automática a la plataforma. El usuario o administrador no se tiene que preocupar de la conexión entre Gateway y ThingsBoard, una vez configurada correctamente esta comunicación ya no hay que preocuparse de monitorizarla.
- Adaptación de los datos de los diferentes protocolos al formato con el que trabaja ThingsBoard. Este formato utiliza una de las interfaces que ofrece la plataforma para la transmisión de información, la conexión usando MQTT.

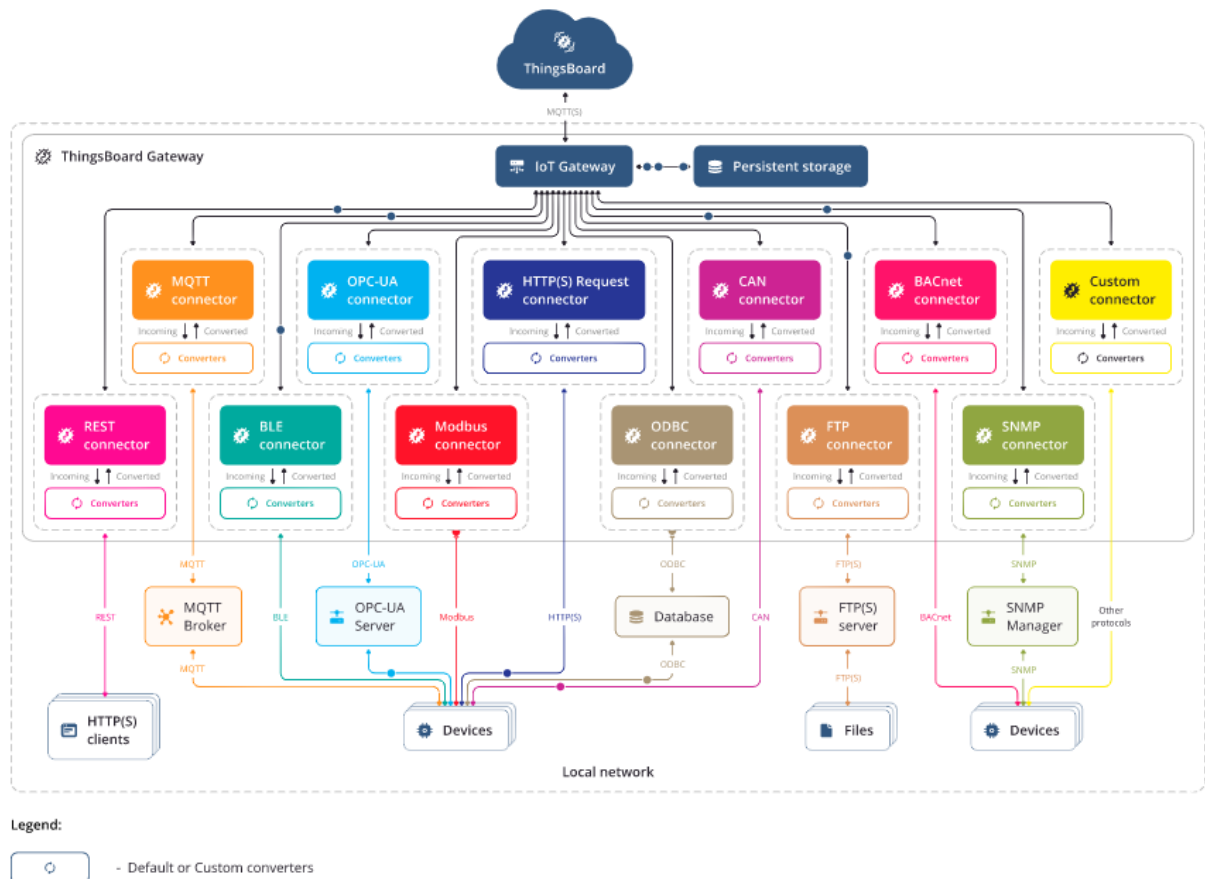


Figura 3.3. Arquitectura IoT Gateway (Fuente: IoT Gateway [27])

Para entender bien este componente hay que conocer cómo trabaja internamente y cuáles son sus partes. Como se puede ver en la Figura 3.3, la estructura es bastante sencilla, facilitando la escalabilidad y una futura ampliación de los protocolos soportados. La arquitectura interna del Gateway se organiza en torno a los siguientes elementos:

- **Conector (*connector*):** es el subcomponente encargado de establecer la conexión con los sistemas externos (ej.: OPC-UA) o directamente con los dispositivos (ej.: ModBus). Escuchará pasivamente los datos que le remitan o consultará activamente esos valores dependiendo del tipo de protocolo de comunicación utilizado.
- **Convertidor (*converter*):** parte encargada de traducir la información del formato del protocolo de origen al formato común que utiliza ThingsBoard. Al igual que los conectores, los convertidores son específicos de cada protocolo. Esta operación se ha de realizar en dos sentidos: de ThingsBoard

hacia el protocolo específico o viceversa. Esto es necesario ya que los datos/instrucciones tienen que fluir en ambas direcciones.

- Almacenamiento de eventos (*events storage*): el gateway almacena la información que recibe hasta que es transmitida a ThingsBoard con el objetivo de conservarla en caso de que haya algún tipo de fallo que impida la transmisión. Hay tres tipos de almacenamiento configurables: en memoria, en fichero persistente o en base de datos de tamaño reducido (SQLite).
- Cliente ThingsBoard: la comunicación con la plataforma se realiza a través del interfaz MQTT. El envío y recepción de información que se realiza en esta parte solo tiene lugar cuando se confirma que hay conexión con ThingsBoard, para evitar generar consumos innecesarios.
- Servicio Gateway: es la parte encargada de organizar y controlar los anteriores subcomponentes, también realiza una recolección de estadísticas que pueden ser enviadas a la plataforma para poder monitorizar el Gateway.

En cuanto a los protocolos soportados nativamente por IoT Gateway, actualmente son:

- REST
- MQTT
- BLE
- HTTP(S)
- FTP
- SNMP
- ODBC
- BACnet
- OPC-UA
- ModBus

Como se ha comentado, IoT Gateway es una solución de código abierto, esto hace que sea muy adaptable para el usuario siempre y cuando tenga los conocimientos necesarios para hacer las modificaciones necesarias. Prueba de ello, es la opción que ofrece de poder incorporar protocolos de comunicación más allá de los que incluye por defecto la herramienta.

Esta opción se puede habilitar utilizando el denominado “Custom Connector”. Es un conector igual que los ya configurados, pero sin un protocolo de comunicación asociado. Es un conector “vacío” que permite al usuario implementar la traducción y comunicación ad hoc al protocolo deseado. Al igual que el resto del componente está desarrollado en Python, siendo relativamente sencillo poder incorporar código para hacer la conversión y conexión necesaria. Como inconveniente hay que señalar que su configuración es bastante más compleja que los conectores nativos del Gateway y requiere más esfuerzo y pruebas para poder ponerlo en funcionamiento.

Desde el punto de vista de la instalación, IoT Gateway contempla que puede ser instalado en las siguientes distribuciones de Linux:

- Ubuntu
- RHEL/CentOs
- Raspbian
- Instalación como paquete de Python
- Contenedor en Windows (Docker)
- Contener en IOS o Linux (Docker)

También se establecen unos requisitos mínimos del sistema para poder instalar el componente. Sin embargo, en este caso coinciden con los necesarios para poder instalar el sistema operativo.

Finalmente, señalar que IoT Gateway es un componente que está bajo desarrollo, de código abierto y desarrollado por la comunidad. Hay que tener en cuenta estos puntos porque significan que el Gateway está en una fase de inmadurez en la que se pueden presentar problemas de estabilidad, liberación de actualizaciones en cortos espacios de tiempo (casi mensualmente) y no dispone de documentación extensa y precisa que ayude en la resolución de problemas. Esto se traduce en un mayor esfuerzo de diagnóstico de problemas y mayor tiempo de puesta en servicio debido al retraso que puedan introducir los contratiempos encontrados.

4. Metodología y configuración

El presente apartado tiene por finalidad presentar la solución adoptada y describir en detalle cómo se ha llevado a cabo. El objetivo principal de este proyecto es desplegar una instancia de la plataforma seleccionada (ThingsBoard), sobre la que se realizarán pruebas con la intención de comprobar algunas de las características que ofrece.

En los siguientes subapartados, se explica cómo es la arquitectura de la maqueta de pruebas a utilizar, los componentes que la componen, las configuraciones a realizar en todos sus elementos y las herramientas de simulación utilizadas para poder llevar a cabo las pruebas.

4.1. ARQUITECTURA DESPLEGADA

La arquitectura básica planteada consta de tres elementos: la plataforma ThingsBoard, IoT Gateway y un equipo o instalación industrial simulado capaz de generar información. Para la realización de las pruebas con ThingsBoard se replicará el comportamiento de un sensor industrial, de manera que se genere la información que podría emitir un sensor de temperatura o de nivel conectado a la plataforma. La principal característica que ha de tener este simulador es ser capaz de comunicar en el protocolo de comunicación industrial que se utilice durante las pruebas.

De esta manera, el dispositivo/instalación industrial será emulado mediante un simulador de los protocolos de comunicación y generará la información que se inyectará a la plataforma ThingsBoard. IoT Gateway hará de nexo entre los dos componentes y será el encargado de realizar la traducción y adaptación de la información.

Esta arquitectura utilizando un simulador permite una rápida configuración de diferentes tipos de equipos, no teniendo que adaptar las pruebas a una instalación física. Además, presenta la ventaja de poder sustituir fácilmente el simulador por equipos reales pudiendo conectar el set de pruebas a una instalación real.

Hay que señalar también que los protocolos de comunicación que se van a utilizar son ModBus TCP y OPC UA. Han sido elegidos debido a que su implantación está muy extendida del ámbito industrial y, además, ya son soportados por el IoT Gateway de ThingsBoard, por lo que su configuración y nivel de madurez serán superiores al desarrollo de un conector nuevo dentro de la plataforma.

Se realizarán dos tipos de instalación para poder probar diferentes aspectos en cada una de ellas. Estas instalaciones son:

- Una instalación local dentro de un solo equipo, más sencilla y fácilmente configurable. Esta instalación hará uso de una máquina virtual dentro de un ordenador personal. Incorporará también simuladores de dispositivos ModBus y OPC UA. Los principales elementos que la componen son:
 - Plataforma ThingsBoard e IoT Gateway instalados en máquina virtual sobre Ubuntu.
 - Simuladores de comunicación ModBus (4.3.1) y OPC UA (4.3.2).
 - PC, portátil o smartphone con capacidad de abrir en un navegador y conectarse a la misma red local de la instalación.

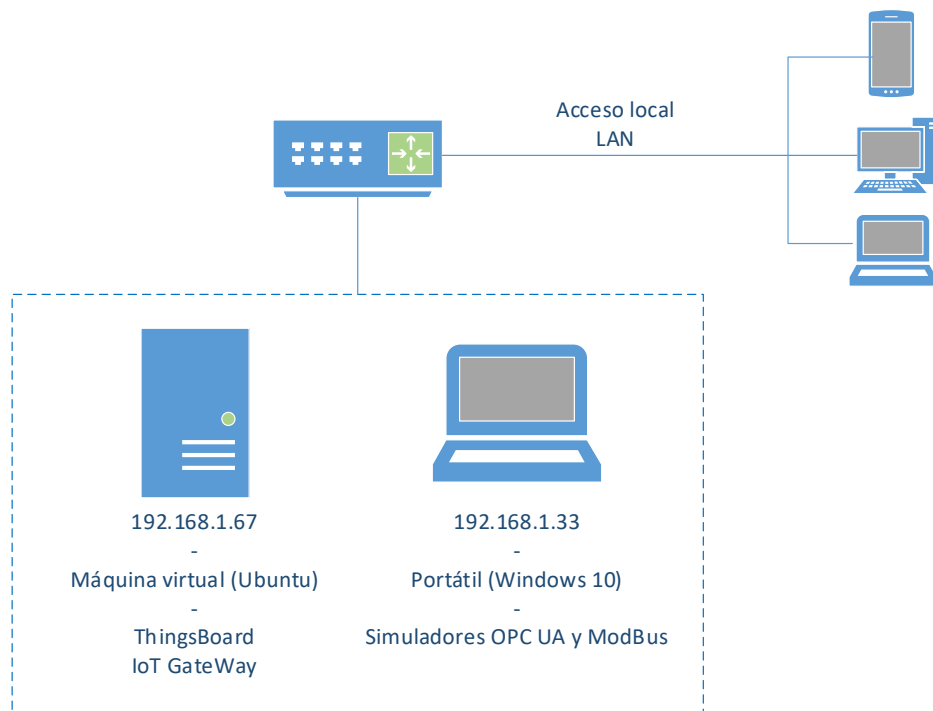


Figura 4.1. Diagrama de red de instalación local (Fuente: elaboración propia)

- Instalación distribuida: llamada así para diferenciarla de la anterior y resaltar la capacidad de conexión en remoto. En este caso, plataforma y punto generador de datos estarán en equipos diferentes simulando una situación más real. No tendrá simuladores de protocolos de comunicación, se inyectará la información a través de peticiones REST. El acceso remoto se hará haciendo uso de una VPN. Los elementos que componen esta instalación son:
 - Plataforma ThingsBoard e IoT Gateway instalados en servidor Raspberry con Raspbian.
 - Fuente generadora de datos en servidor Raspberry con Raspbian.
 - PC, portátil o smartphone con capacidad de abrir en un navegador y conectarse a la misma red local de la instalación o la VPN.

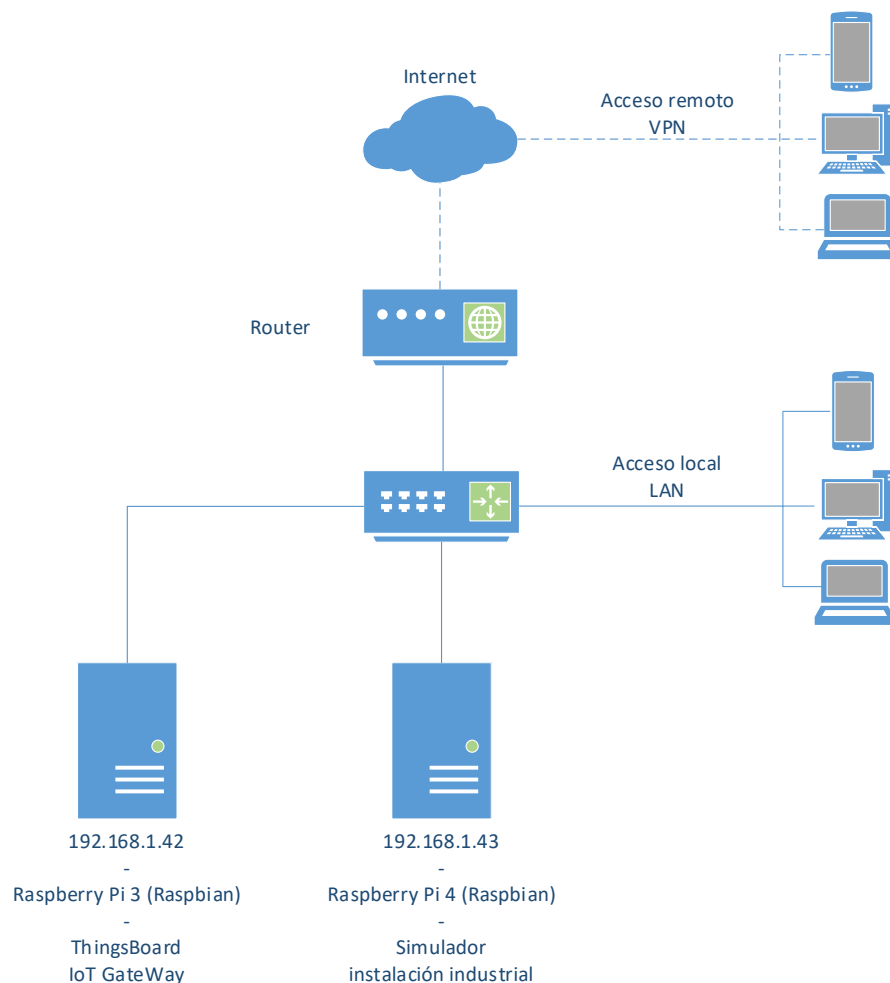


Figura 4.2. Esquema de red instalación distribuida (Fuente: elaboración propia)

A continuación, se describen los elementos que componen los sets de simulación y sus características más en detalle

4.1.1. PLATAFORMA THINGSBOARD

Elemento principal de la maqueta de pruebas, es encargado de gestionar toda la información que se reciba y de organizarla.

En este caso, ThingsBoard es desplegada en la máquina principal de cada set. Las características de las máquinas utilizadas son:

- Instalación local
 - SO: Máquina virtual con Ubuntu 20.04.4 LTS sobre Windows 11
 - Hardware: Huawei D15 Matebook
- Instalación distribuida:
 - SO: Raspbian 11
 - Hardware: Raspberry Pi 4

4.1.2. IOT GATEWAY

Componente encargado de habilitar la comunicación ModBus con el simulador y pasar la información a ThingsBoard a través de MQTT. Al igual que ThingsBoard, IoT Gateway es desplegado en la máquina principal del set de pruebas. Las características son las mismas que en el caso anterior.

4.1.3. SIMULADOR MODBUS

Para emular la comunicación ModBus se ha utilizado el simulador de ModBus ModRSsim2. Este simulador permite fácilmente gestionar todos los bytes de memoria como un servidor ModBus. Se ha ejecutado desde en la máquina con SO Windows 11 en los dos sets de pruebas. En el subapartado 4.3.1 se habla más en detalle.

4.1.4. SIMULADOR OPC-UA

En el caso de OPC se ha utilizado la herramienta Prosys OPC UA Simulation Server para simular la comunicación. En este caso, permite generar los objetos a intercambiar por OPC e ir cambiando valores de manera manual o automática. En el subapartado 4.1.4 se explica más en detalle.

4.2. CONFIGURACIÓN

A la hora de desplegar la plataforma y el Gateway, una de las partes que más dificultades plantea es la selección del tipo de instalación que se desea de entre las que ofrece ThingsBoard y su posterior configuración.

En los siguientes subapartados, se recogen las diferentes opciones de instalación elegidas y las configuraciones de ThingsBoard e IoT Gateway.

4.2.1. CONFIGURACIÓN THINGSBOARD

Las elecciones de configuración de la instalación escogidas para ThingsBoard son:

- Despliegue sobre servidor físico (Ubuntu o Raspbian)
- Base de datos SQL, en concreto PostgreSQL
- Sistema de gestión de eventos entre los servicios

Son las opciones elegidas al ser las recomendadas para el tipo de instalación que se necesita (a pequeña escala y en local).

La instalación de ThingsBoard es bastante sencilla y también su posterior configuración.

En el Anexo 1, se incluye el fichero de configuración que se ha utilizado durante las pruebas efectuadas en este proyecto. En él se ve las líneas que hay que añadir relativas a la configuración de la base de datos. El usuario solo ha de introducir la contraseña configurada en el usuario de PostgreSQL para que la plataforma pueda conectarse.

A modo informativo, en la página oficial de la plataforma hay una guía de instalación paso por paso bastante detallada que ayuda en este proceso.

4.2.2. CONFIGURACIÓN IOT GATEWAY

En el caso del IoT Gateway, la única elección de instalación que hay disponible es la elección del tipo de servidor sobre el que se va a desplegar el componente. En este caso se realiza sobre el mismo servidor físico que ThingsBoard: Ubuntu para instalación “local” y Raspbian para instalación “distribuida”.

El proceso de instalación es relativamente sencillo, se instala un paquete *.deb* que después será ejecutado como servicio.

Una vez instalado el componente, hay que configurarlo para que se adapte a nuestra instalación. Se configura mediante ficheros, habiendo un fichero principal (*tb_gateway.yaml*) y después hay un fichero de configuración por cada protocolo de comunicación. En este caso, se utilizarán los protocolos ModBus y OPC UA, por lo que solo habrá que configurar un fichero para cada protocolo correspondiente (*modbus.json* y *opc.json*).

A continuación, se detallan los puntos más importantes de configuración en ambos ficheros:

- *tb_gateway.yaml*
 - Dirección y puerto en el que está desplegado ThingsBoard (*localhost:1883*).
 - *accessToken*: identificador del dispositivo generado durante la creación desde ThingsBoard. Es necesario para que la plataforma sea capaz de asociar la información recibida a un dispositivo.
 - *storage*: tipo de almacenamiento interno del Gateway. En el caso de esta instalación se ha utilizado la opción *SQLite*.
 - *connectors*: conectores que se van a utilizar indicando la localización del fichero de configuración correspondiente. En este caso solo se indica el conector *ModBus*.

- *modbus.json*

Este fichero de configuración permite configurar el Gateway con un servidor para consultar información en los distintos esclavos o configurar el Gateway como esclavo para poder enviar información a un servidor Gateway.

En este caso se utiliza la configuración como maestro indicando la presencia de un esclavo del que se lee la información. Aspectos a destacar de este fichero:

- Dirección y puerto en el que está desplegado el dispositivo ModBus.
- Tipo de comunicación ModBus, en este caso TCP.
- Diferentes periodos de tiempo (periodo de consulta, periodo de reconexión, tiempo de espera a la reconexión...).
- Atributos y telemetrías a solicitar. En este caso hay que indicar la dirección de memoria a consultar, el tipo de variable (int, float...), el código de función ModBus y el número de objetos a leer.

- opcua.json

Fichero de configuración para comunicar el Gateway con un dispositivo que utilice OPC UA, permite la recepción en forma de objetos OPC. La configuración es sencilla, solo hay que indicar el nombre del objeto y la dirección de memoria. Los parámetros de configuración más importantes de este fichero son:

- URL donde se levanta el servidor OPC.
- Dirección de memoria donde se encuentran los objetos OPC a comunicar.
- Nombre y dirección de cada uno de los objetos.

Estos ficheros se pueden consultar en los anexos: Anexo 2 - Fichero de configuración IoT Gateway, Anexo 3 - Fichero de configuración conector ModBus IoT Gateway y Anexo 4- Fichero de configuración conector OPC UA IoT Gateway.

4.3. SIMULADORES DE COMUNICACIONES INDUSTRIALES

Uno de los aspectos claves de este proyecto, que lo hace interesante dentro del ámbito industrial, es la posibilidad que ofrece Thingsboard para utilizar protocolos de comunicación industrial y, por lo tanto, interoperar con equipos y elementos industriales.

Como se ha detallado en el apartado 3.8, el componente que permite esta conexión es IoT Gateway. Ofrece la opción de comunicación utilizando 10 protocolos de

manera nativa y la posibilidad de personalizar cualquier otro protocolo. Como se ha visto, los protocolos con los que se va a probar esta capacidad de comunicación son ModBus y OPC-UA. Son los protocolos elegidos por ser ampliamente utilizados en la industria (son estándares industriales) y estar ya implementados en IoTGateway.

A la hora de diseñar la maqueta de pruebas del proyecto se ha decidido utilizar simuladores de estos dos protocolos de comunicación. La razón de utilizar simuladores es poder establecer comunicaciones con elementos externos de manera manejable y permitir probar diferentes opciones dentro de ThingsBoard para comprobar el funcionamiento de la plataforma. En los dos siguientes subapartados, se presentan los dos simuladores utilizados y las características de cada uno de ellos.

4.3.1. MODRSIM2

ModRSim2 [28] es un simulador de esclavos ModBus tanto para RS-232 como para TCP/IP. Es una solución de código abierto

A continuación, se incluyen algunas capturas del programa y la explicación de ciertas características y funcionalidades que tiene:

- La pantalla principal presenta un menú de configuración en la parte superior. Se pueden configurar aspectos como: puerto de comunicación, tipo de comunicación ModBus (TCP o RS232), tipos de simulación, etc.
- La parte central de la pantalla la componen las direcciones de memoria del esclavo. De esta manera, el usuario puede elegir el tipo de variable a utilizar (coils, analógicas, digitales, holding registers o registros extendidos), su dirección y el valor a introducir.
- Una de las opciones que presenta es poder configurar el programa para que vaya variando el valor de las variables de manera automática.

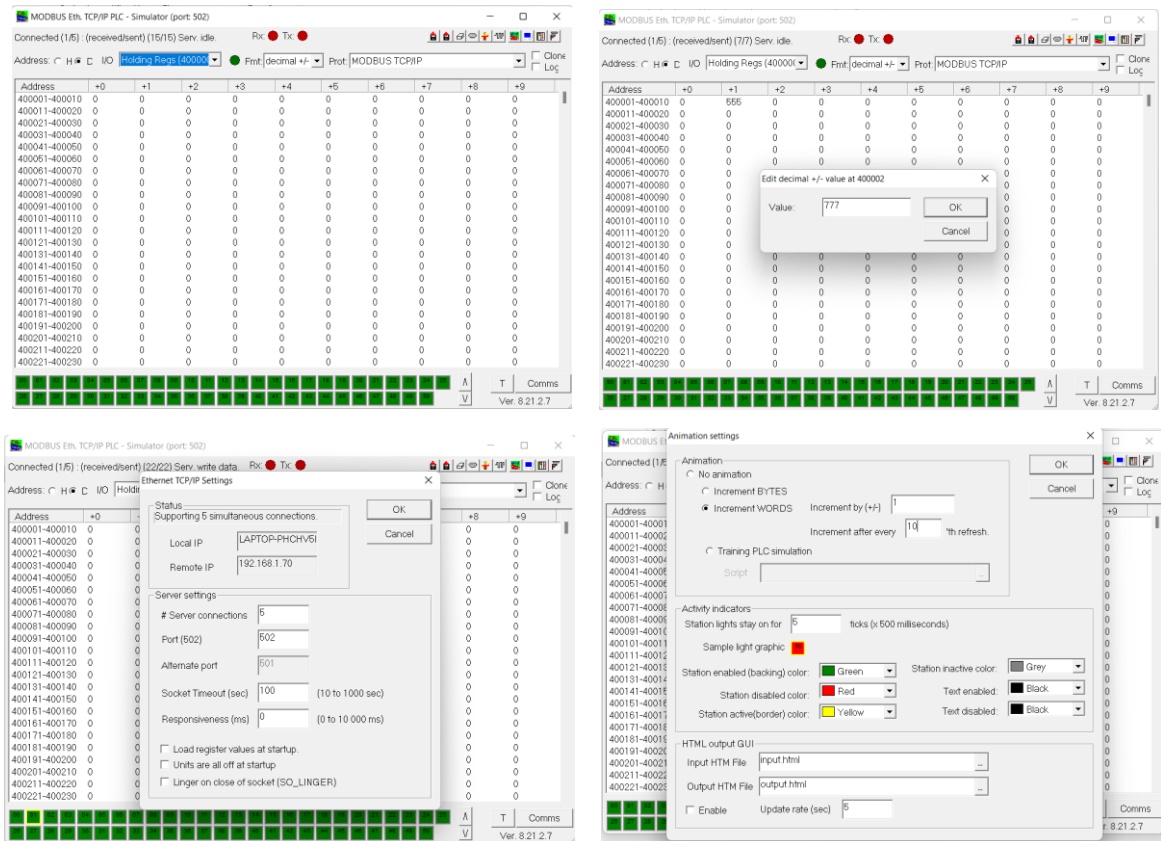


Figura 4.3. Conjunto de capturas de ModRSim2 (Fuente: elaboración propia / ModRSim2 [28])

4.3.2. PROSYS OPC UA SIMULATION SERVER

Prosys OPC UA Simulation Server [29] es un software desarrollado por la empresa Prosys OPC Ltd (proveedor de productos y servicios basados en tecnología OPC UA). Este programa tiene por función principal ofrecer la posibilidad de probar clientes, conexiones y modelos de información OPC UA antes de ser puestos en funcionamiento en un entorno real. Está basado en tecnología Java lo que facilita que sea multiplataforma, estando disponible para Windows, Linux y MacOS: Algunas de las características que indica el fabricante son:

- Modelo de información actualizado a la última versión OPC UA.
- Simulación de datos y eventos. Señales de simulación predefinidas ideales para simular el comportamiento de equipos reales.
- Interfaz de usuario sencillo que permite configurar diferentes modelos de datos, conexiones o simulaciones.

- Soporta diferentes tipos de conexión y todos los modos y políticas de seguridad del estándar OPC UA:
- Certificado de cumplimiento OPC UA por la fundación OPC.

A continuación, se presentan capturas de pantalla del simulador que ayudan a ver cómo funcionar con este programa y cómo se ha utilizado:



Figura 4.4. Pantalla principal OPC UA Simulation Server (Fuente: elaboración propia / Prosys [29])

- Pantalla principal: aparece nada más arrancar el programa. Presenta información relativa al status en el que se encuentra el servidor OPC o las direcciones en las que está levantado. En la parte superior se pueden ver el resto de las pestañas que permiten el acceso a las pantallas en las que se puede configurar los modelos de información.
- Modelo de datos: el programa permite configurar diferentes objetos y tipos de datos que serán intercambiados a través de OPC.

En la siguiente captura, se ve la pestaña de configuración de objetos de información. En ella, se aprecia como se configura una simulación en la que el programa automáticamente va cambiando los valores de las variables siguiendo ciertas funciones.

The screenshot displays the Prosys OPC UA Simulation Server interface. The main window is titled "Prosys OPC UA Simulation Server" and includes a menu bar with "Options" and "Help". Below the menu bar, there are tabs for "Status", "Objects", "Types", and "Namespaces". The "Objects" tab is active, showing a tree view of the simulation objects. The "level" object is selected, and its configuration is displayed in the right-hand pane.

The configuration pane for the "level" object shows the following settings:

- Attributes:** Inherit from BaseDataVariableType (checked).
- Value Type:** Counter.
- Data Type:** Int32.
- Array Size:** 0.
- Value Parameters:**
 - Min Value: 0.
 - Increment: 1.
 - Max Value: 100.
 - Direction: Up.
- Value Plot:** A line graph showing the value of the object over time. The y-axis ranges from 0 to 96. The plot shows a sawtooth pattern, starting at 0, increasing linearly to 96, and then resetting to 0. The current value is 50.

The bottom status bar shows the current value of the selected object: "Current Value: 50".

Figura 4.5. Captura OPC UA Simulation Server configuración objetos (Fuente: elaboración propia / Prosys [29])

5. Resultados

A lo largo de este apartado se detallan las pruebas realizadas con la plataforma ThingsBoard, así como los objetivos buscados y resultados obtenidos tras la instalación de la plataforma y realización de pruebas de funcionamiento de esta.

5.1. PRUEBAS REALIZADAS

5.1.1. OPERATIVIDAD EN DIFERENTES SISTEMAS OPERATIVOS (SOS)

Descripción

Instalación de ThingsBoard e IoT Gateway en SOs diferentes. Los sistemas operativos son: Ubuntu 20.04.4 LTS, Raspbian 11, RHEL 8 y RHEL 9.

Objetivos

Comprobar la capacidad del sistema ThingsBoard de ser desplegado en diferentes distribuciones Linux y bajo diferentes arquitecturas (máquina virtual, servidor físico y contenedor).

Resultados

Se ha podido realizar la instalación de ThingsBoard en todos los sistemas operativos probados. En el caso de IoT Gateway, se ha conseguido instalar en Ubuntu y Raspbian. Sin embargo, en RHEL se ha realizado la instalación, pero no se ha conseguido arrancar el componente en un primer momento debido a problemas de compatibilidad con ciertas librerías. Tras reinstalar las librerías problemáticas a versiones anteriores, se solucionó el problema.

Destacar también que la instalación de la base de datos PostgreSQL (se realiza de manera independiente a la instalación de ThingsBoard) varía en función de la distribución.

5.1.2. GESTIÓN DE USUARIOS EN LA PLATAFORMA

Descripción

Creación y gestión de usuarios dentro de ThingsBoard. La prueba consiste en la creación de un usuario desde el administrador del sistema. Posteriormente se accede al sistema con el usuario creado y se comprueba que tiene acceso a los datos esperados.

Objetivos

Comprobar la capacidad de la plataforma de gestionar diferentes usuarios y configurar.

Resultados

Creación de usuarios satisfactoria, pudiendo operar todas las funcionalidades deseadas sin problemas.

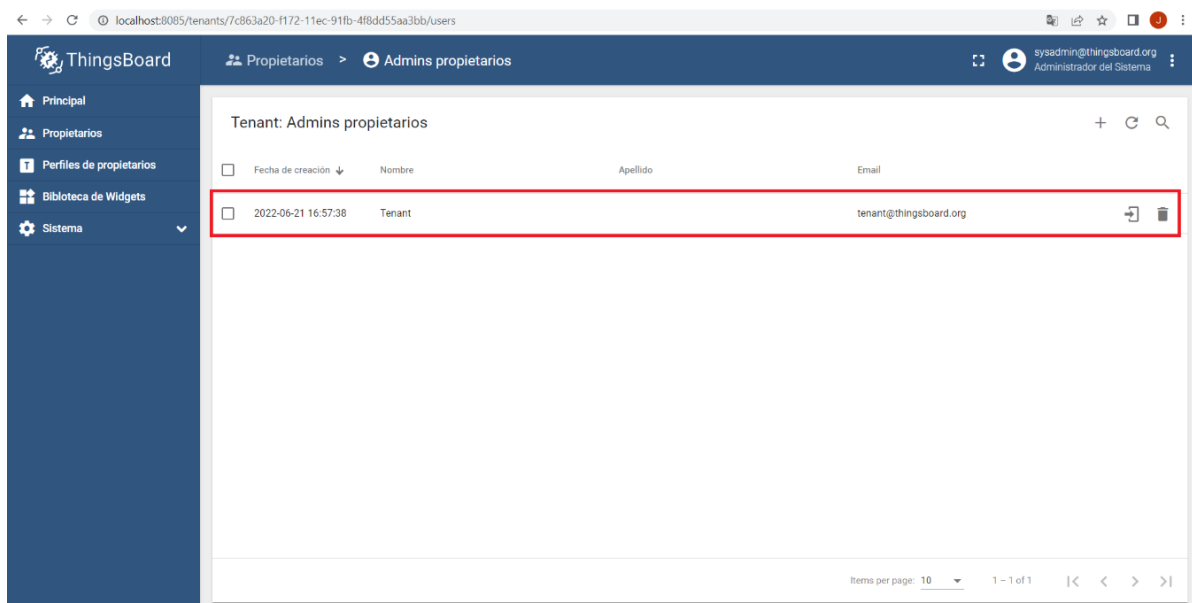


Figura 5.1. Usuario creado (Fuente: elaboración propia)

5.1.3. CREACIÓN DE DISPOSITIVOS

Descripción

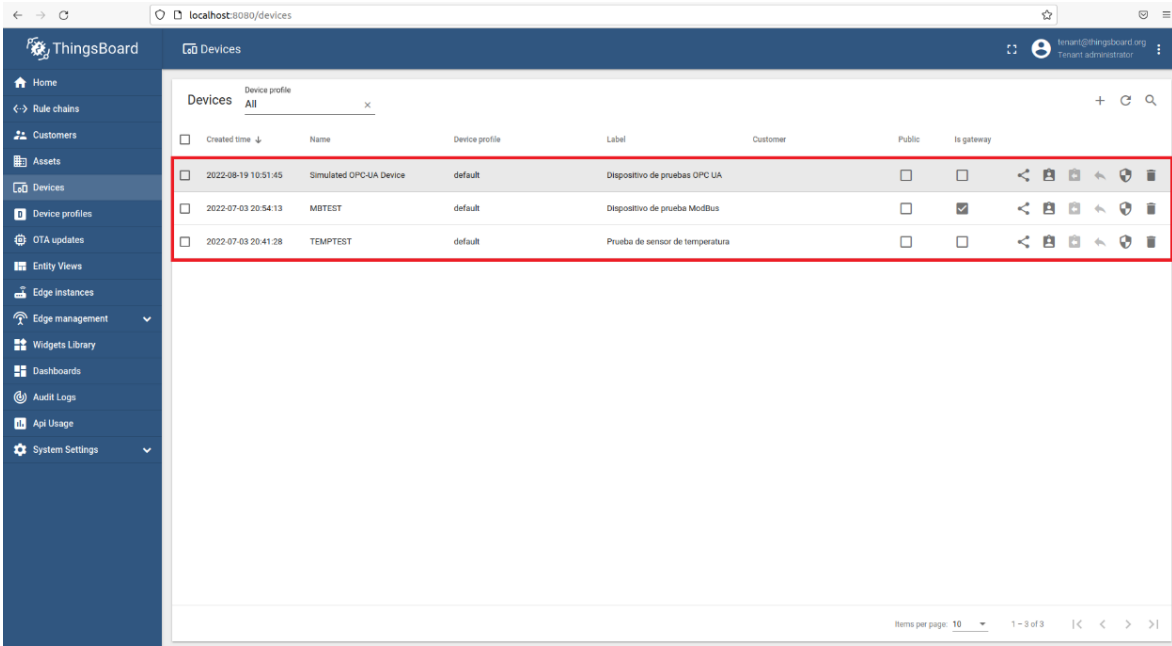
Creación de dispositivos con diferentes características de comunicación. Se crean tres dispositivos cada uno utilizando un protocolo de comunicación distinto: OPC UA; ModBus y API REST.

Objetivos

Demostrar la capacidad de la plataforma para la gestión de diferentes dispositivos que utilizan distintas vías de comunicación con la plataforma.

Resultados

La creación se realiza de manera satisfactoria, asignando a cada dispositivo características diferentes de comunicación y se redirecciona esa información a distintos paneles.



The screenshot shows the ThingsBoard web interface at localhost:8080/devices. A table lists three devices, each with a checkbox, creation time, name, device profile, label, customer, public status, gateway status, and action icons. A red box highlights the three device rows.

<input type="checkbox"/>	Created time ↓	Name	Device profile	Label	Customer	Public	Is gateway	
<input type="checkbox"/>	2022-08-19 10:51:45	Simulated OPC-UA Device	default	Dispositivo de pruebas OPC UA		<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	2022-07-03 20:54:13	MBTEST	default	Dispositivo de prueba ModBus		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	2022-07-03 20:41:28	TEMPTEST	default	Prueba de sensor de temperatura		<input type="checkbox"/>	<input type="checkbox"/>	

Figura 5.2. Dispositivos creados (Fuente: elaboración propia)

5.1.4. CONEXIÓN DE DISPOSITIVO MODBUS

Descripción

Conexión del simulador ModBus a la plataforma con el objetivo de transmitir información.

Objetivos

Demostrar la capacidad del sistema de recibir información e interoperar a través del protocolo ModBus sobre TCP.

Resultados

Se consigue establecer la comunicación a través de ModBus. Sin embargo, se detecta un problema en la comunicación entre IoT Gateway y ThingsBoard. De los tres tipos de comunicación posible, solo funciona el que es por base de datos local.

MODBUS Eth. TCP/IP PLC - Simulator (port: 502)

Connected (1/5) : (received/sent) (472/472) Serv. write data. Rcv: Tx:

Address: H D I/O Holding Regs (400000) Fmt: decimal +/- Prot: MODBUS TCP/IP

Address	0	+1	+2	+3	+4	+5	+6	+7	+8	+9
400001-400010	70	0	0	0	0	0	0	0	0	0
400011-400020	0	0	0	0	0	0	0	0	0	0
400021-400030	0	0	0	0	0	0	0	0	0	0
400031-400040	0	0	0	0	0	0	0	0	0	0
400041-400050	0	0	0	0	0	0	0	0	0	0
400051-400060	0	0	0	0	0	0	0	0	0	0
400061-400070	0	0	0	0	0	0	0	0	0	0
400071-400080	0	0	0	0	0	0	0	0	0	0
400081-400090	0	0	0	0	0	0	0	0	0	0
400091-400100	0	0	0	0	0	0	0	0	0	0
400101-400110	0	0	0	0	0	0	0	0	0	0
400111-400120	0	0	0	0	0	0	0	0	0	0
400121-400130	0	0	0	0	0	0	0	0	0	0
400131-400140	0	0	0	0	0	0	0	0	0	0
400141-400150	0	0	0	0	0	0	0	0	0	0
400151-400160	0	0	0	0	0	0	0	0	0	0
400161-400170	0	0	0	0	0	0	0	0	0	0
400171-400180	0	0	0	0	0	0	0	0	0	0
400181-400190	0	0	0	0	0	0	0	0	0	0
400191-400200	0	0	0	0	0	0	0	0	0	0
400201-400210	0	0	0	0	0	0	0	0	0	0
400211-400220	0	0	0	0	0	0	0	0	0	0
400221-400230	0	0	0	0	0	0	0	0	0	0
400231-400240	0	0	0	0	0	0	0	0	0	0
400241-400250	0	0	0	0	0	0	0	0	0	0
400251-400260	0	0	0	0	0	0	0	0	0	0
400261-400270	0	0	0	0	0	0	0	0	0	0
400271-400280	0	0	0	0	0	0	0	0	0	0
400281-400290	0	0	0	0	0	0	0	0	0	0
400291-400300	0	0	0	0	0	0	0	0	0	0
400301-400310	0	0	0	0	0	0	0	0	0	0

Ver: 8.21.2.7

Figura 5.3. ModRSim2 con valor ModBus a enviar a ThingsBoard (Fuente: elaboración propia)

Created time ↓	Name	Device profile
2022-08-19 10:51:45	Simulated OPC-UA Device	default
2022-07-03 20:54:13	MBTEST	default
2022-07-03 20:41:28	TEMPTEST	default

Last update time	Key ↑	Value
2022-08-21 20:54:24	t6uint_read	70
2022-08-21 20:02:14	eventsProduced	163
2022-08-21 20:02:14	eventsSent	163
2022-08-21 20:02:14	modbusconnectorEventsProduced	163
2022-08-21 20:02:14	modbusconnectorEventsSent	163
2022-08-21 20:02:14	opc-uaconnectorEventsProduced	0
2022-08-21 20:02:14	opc-uaconnectorEventsSent	0

Figura 5.4. Valor ModBus recibido en plataforma ThingsBoard (Fuente: elaboración propia)

5.1.5. CONEXIÓN DE DISPOSITIVO OPC UA

Descripción

Conexión del simulador OPC al sistema ThingsBoard para transmitir información. Se pretende demostrar la capacidad de la plataforma de interoperar utilizando OPC UA.

Objetivos

Demostrar la capacidad del sistema de recibir información e interoperar a través del modelo de datos OPC UA.

Resultados

Se establece la conexión y se consigue ver los valores configurados desde el simulador.

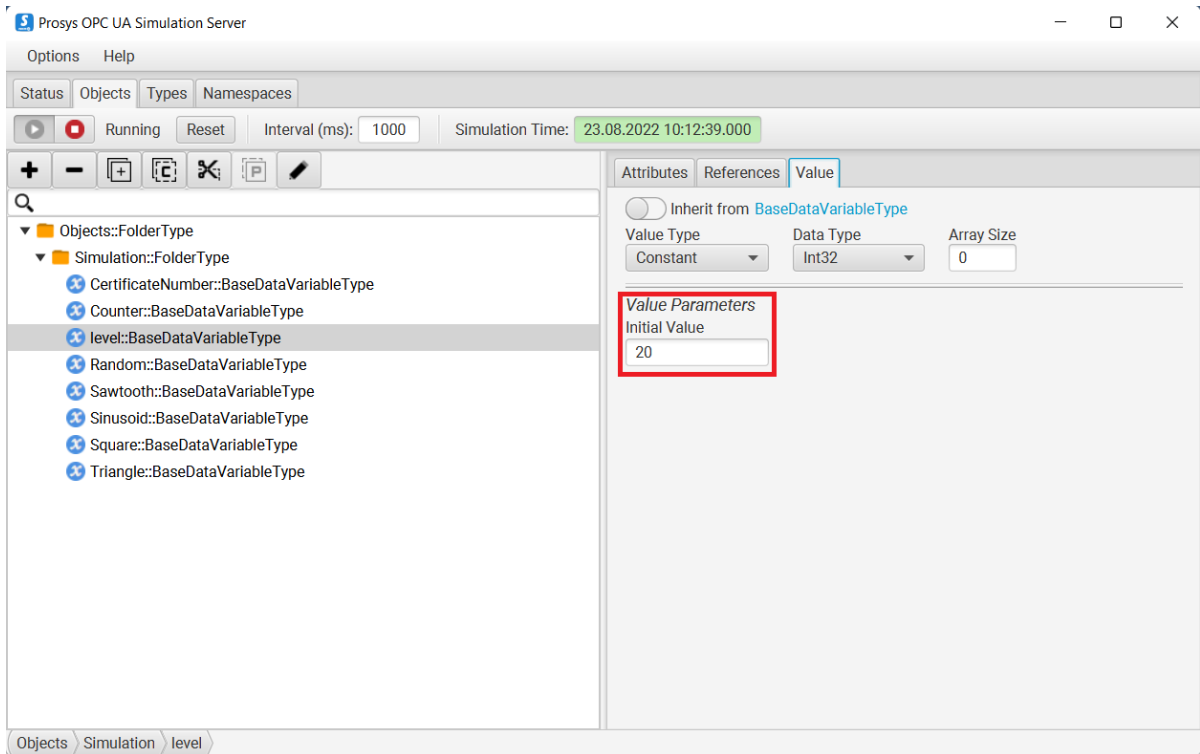


Figura 5.5. Valor OPC a enviar en OPC Simulation Server (Fuente: elaboración propia)

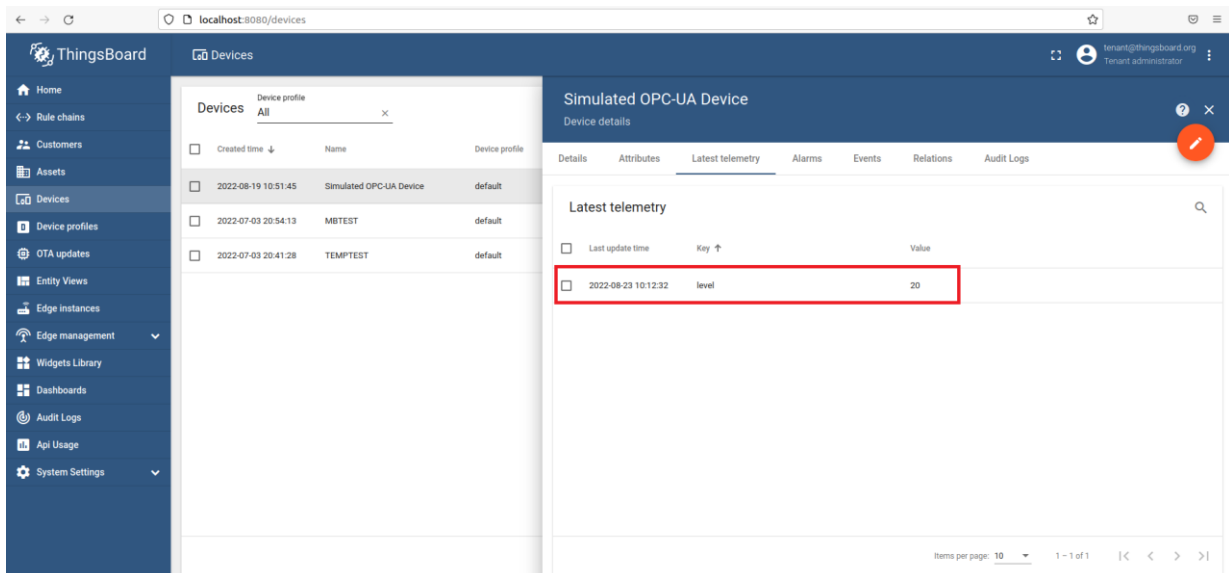


Figura 5.6. Valor OPC recibido en ThingsBoard (Fuente: elaboración propia)

5.1.6. VISUALIZACIÓN DE INFORMACIÓN EN DASHBOARDS

Descripción

Redireccionar la información recibida a diversos paneles para visualizar la información actual y anterior de manera gráfica.

Objetivos

Visualizar la información recibida de los dispositivos en diferentes paneles gráficos en la plataforma.

Resultados

Se muestra la información recibida en diferentes paneles en forma de gráfica, tabla o cartulina mostrando los valores dentro de una escala y el timestamp de recepción de cada uno.

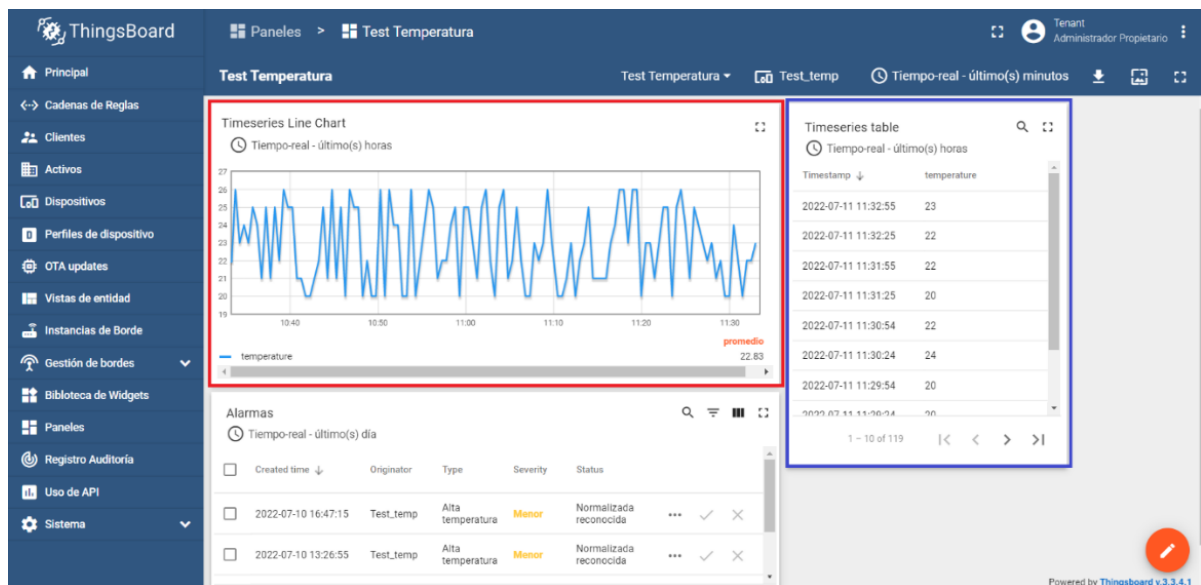


Figura 5.7. Paneles de visualización de datos (Fuente: elaboración propia)

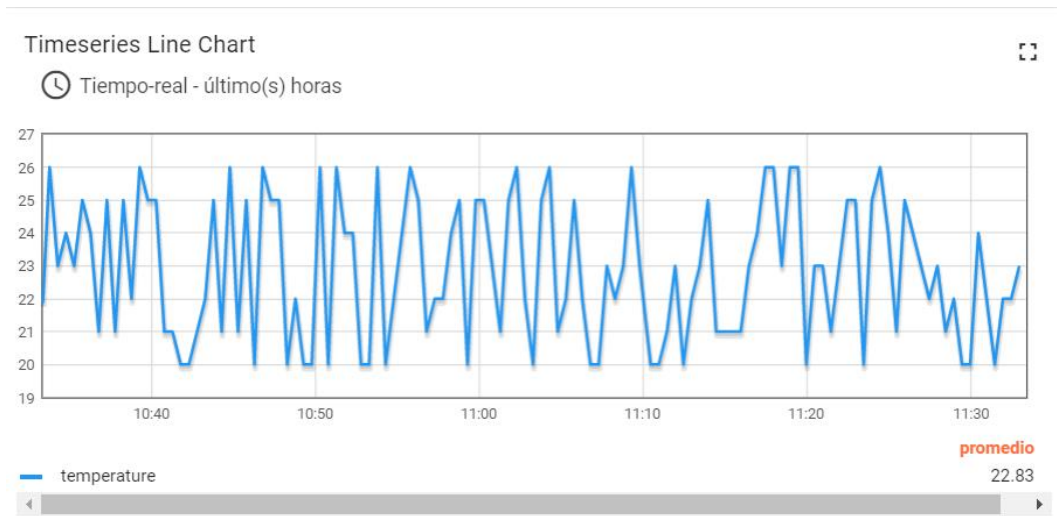


Figura 5.8. Detalle gráfico valores de temperatura (Fuente: elaboración propia)

Timeseries table

🕒 Tiempo-real - último(s) horas

Timestamp ↓	temperature
2022-07-11 11:32:55	23
2022-07-11 11:32:25	22
2022-07-11 11:31:55	22
2022-07-11 11:31:25	20
2022-07-11 11:30:54	22
2022-07-11 11:30:24	24
2022-07-11 11:29:54	20
2022-07-11 11:29:24	20

1 - 10 of 119

Figura 5.9. Detalle tabla valores temperatura (Fuente: elaboración propia)

5.1.7. GENERACIÓN DE ALARMAS

Descripción

Configuración de una regla para la generación de alarmas en función del valor de entrada, comprobación de que cuando la regla se cumple esa alarma se dispara y posterior reconocimiento de la lectura de la alarma por parte del operador.

Objetivos

Establecer reglas en la plataforma a partir de las cuáles generar alarmas para comunicar al operador algún comportamiento anómalo o fuera de los parámetros configurados.

Resultados

Mediante el simulador ModBus se emula el comportamiento de un sensor de temperatura ambiente. Se establece una regla de temperatura superior a 25 °C. Al superarse esa temperatura, se genera una alarma que se notifica a través del interfaz de usuario.

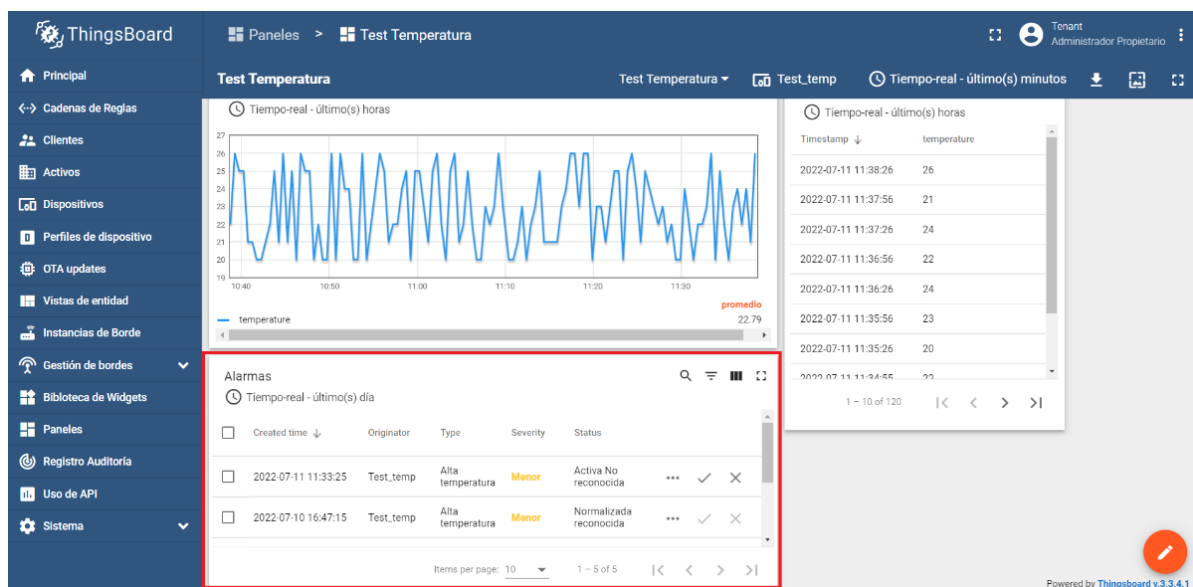


Figura 5.10. Panel de alarmas (Fuente: elaboración propia)

Alarmas

🕒 Tiempo-real - último(s) día

<input type="checkbox"/>	Created time ↓	Originator	Type	Severity	Status			
<input type="checkbox"/>	2022-07-11 11:33:25	Test_temp	Alta temperatura	Menor	Activa No reconocida	...	✓	✗
<input type="checkbox"/>	2022-07-10 16:47:15	Test_temp	Alta temperatura	Menor	Normalizada reconocida	...	✓	✗

Items per page: 10 1 - 5 of 5

Figura 5.11. Detalle panel de alarmas (Fuente: elaboración propia)

The screenshot shows the ThingsBoard interface with a sidebar on the left containing navigation options like 'Principal', 'Cadenas de Reglas', 'Clientes', 'Activos', 'Dispositivos', 'Perfiles de dispositivo', 'OTA updates', 'Vistas de entidad', 'Instancias de Borde', 'Gestión de bordes', 'Biblioteca de Widgets', 'Paneles', 'Registro Auditoría', 'Uso de API', and 'Sistema'. The main area is split into two panels: 'Dispositivos' and 'Test_temp'. The 'Dispositivos' panel shows a list of devices with columns for 'Fecha de creación', 'Nombre', and 'Perfil de dispositivo'. The 'Test_temp' panel shows details for a specific device, with a sub-panel for 'Alarmas' (alarms) displaying a table of alarm records. The alarm records table has columns for 'Hora de creación', 'Origen', 'Tipo', 'Gravedad', 'Estado', and 'Detalle...'. The records show various temperature-related alarms with severity levels like 'Menor' and statuses like 'Activa No reconocida' or 'Normalizada reconocida'.

Figura 5.12. Registro alarmas (Fuente: elaboración propia)

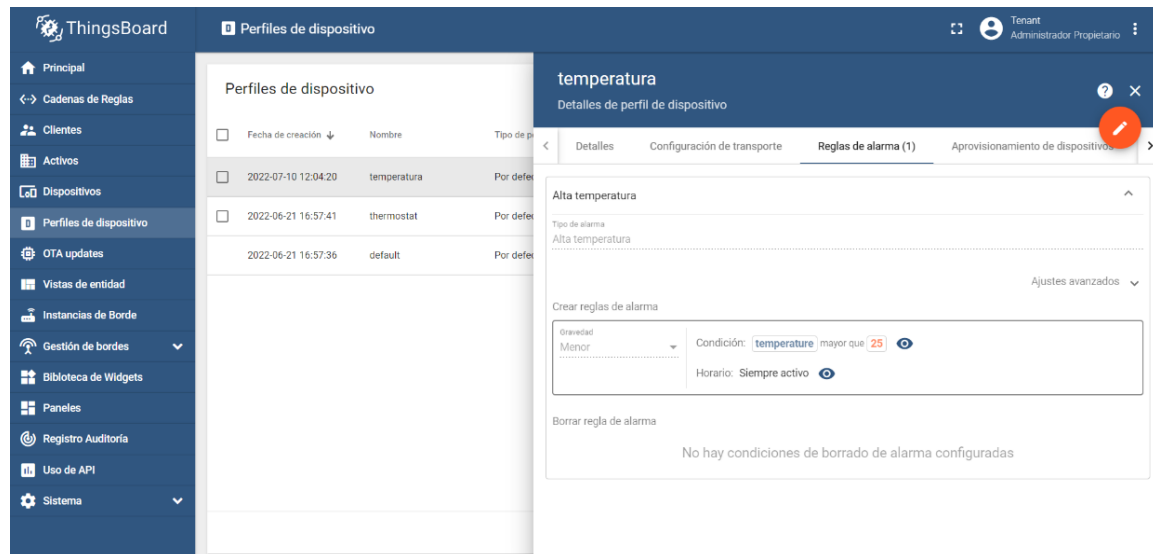


Figura 5.13. Regla disparo de alarma alta temperatura (Fuente: elaboración propia)

5.1.8. PROCESADO BÁSICO DE LA INFORMACIÓN

Descripción

Realizar un cálculo básico con la información recibida en alguno de los dispositivos configurados.

Objetivos

Demostrar la capacidad de la plataforma de tratar la información a nivel básico.

Resultados

Se realiza el valor promedio de los valores de temperatura simulados recibidos desde el simulador ModBus. Se muestra el valor calculado en uno de los paneles.

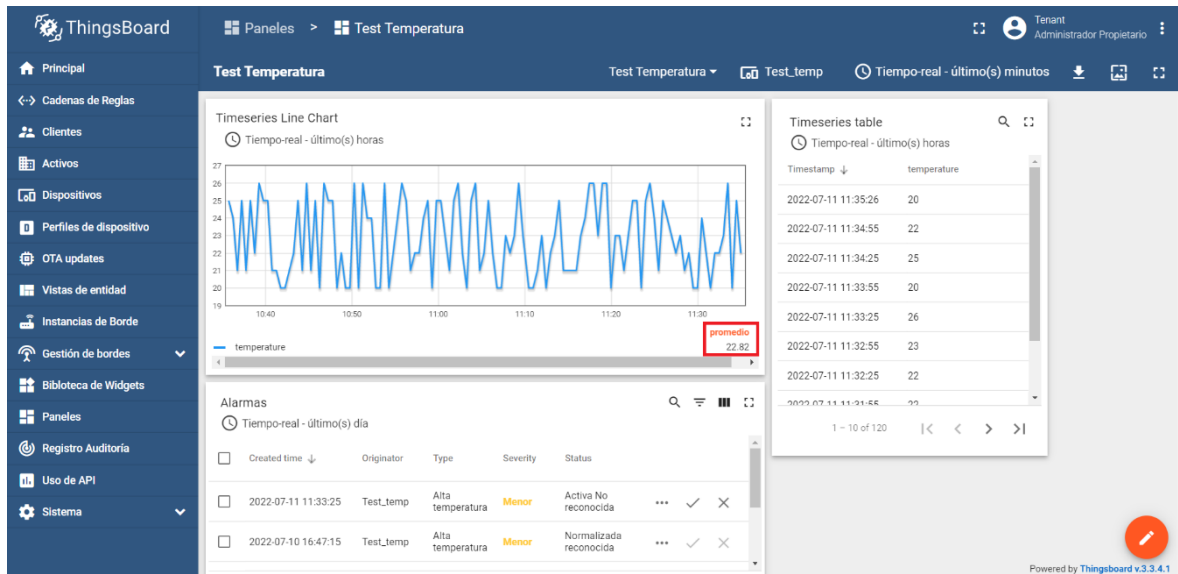


Figura 5.14. Valor promedio de temperatura (Fuente: elaboración propia)

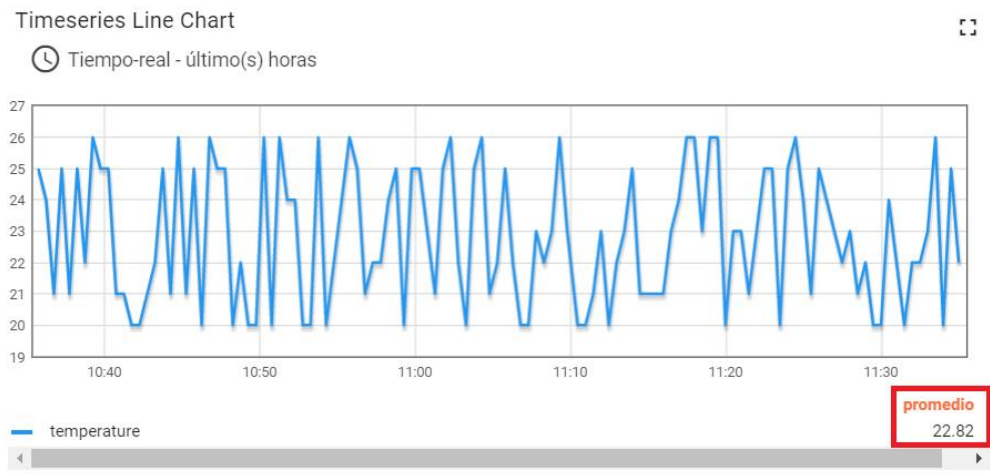


Figura 5.15. Detalle valor promedio (Fuente: elaboración propia)

5.1.9. CONEXIÓN REMOTA A LA PLATAFORMA

Descripción

Acceder remotamente a la plataforma desde una red externa a la de la instalación.

Objetivos

Ver que las tecnologías utilizadas en ThingsBoard permiten habilitar un acceso remoto a la plataforma.

Resultados

Aprovechando la capacidad de la maqueta de conexión desde el exterior utilizando una VPN instalada, se accede a la plataforma desde una red exterior. Se consigue conexión y operar normalmente en ThingsBoard.

5.2. PROBLEMAS DETECTADOS

A lo largo de la realización de las pruebas con la plataforma se han detectado ciertos problemas relacionados principalmente con la comunicación entre ThingsBoard e IoT Gateway.

Desde el punto de vista únicamente de la plataforma ThingsBoard, se puede concluir que es un software maduro y estable, no se han detectado problemas y errores durante las pruebas realizadas.

En el caso de IoT Gateway, hay que tener en cuenta que no es un componente definitivo y se encuentra bajo desarrollo continuo. Presenta problemas de estabilidad y algunas de sus características no funcionan.

Uno de los problemas detectado surge a la hora de instalar en determinados sistemas operativos, el componente no es capaz de arrancar. Este problema está debido incompatibilidades con librerías Python como protobuf. IoT Gateway no es compatible con las versiones de las librerías que vienen con el sistema operativo. La forma de solucionarlo identificada es la reinstalación de las librerías con versiones anteriores para solventar esos problemas.

Otro de los problemas detectados está en la comunicación Gateway-ThingsBoard. Hay disponibles tres formas de configuración de esta comunicación: en memoria, a través de fichero o por base de datos interna. Las dos primeras no funcionan siendo un error en la plataforma [30].

6. Conclusiones

El aumento de la cantidad de información generada y consumida en las instalaciones industriales es una realidad, y la necesidad de obtener conclusiones útiles a partir de ellos es un requisito indispensable para que las empresas puedan mejorar y sobrevivir en un mercado tan competitivo como es el actual.

En los entornos industriales modernos, los términos Industria 4.0 e IoT son cada vez más comunes. Desde el punto de vista práctico, se traducen en dispositivos, elementos o equipos con una capacidad de generación de datos, procesado y, principalmente, comunicación mucho más avanzada. Aumentando la disponibilidad de esta información en lugares y tiempos hasta ahora inimaginables.

Una de las labores principales de los ingenieros es proveer de soluciones a las empresas para que puedan adaptarse a y aprovecharse de las nuevas tecnologías que van apareciendo. A través de este proyecto, se ha presentado y probado una plataforma para la recepción, gestión y almacenamiento centralizado de información proveniente de instalaciones industriales. Con esta herramienta, se pretende que una industria tenga mayor capacidad de obtención y organización de información ofreciendo un acceso rápido y sencillo para los usuarios interesados.

La plataforma seleccionada, ThingsBoard, está desarrollada bajo software de código abierto y centrada en la gestión de dispositivos del ámbito IoT que permite la recepción, visualización y almacenamiento de información.

Además, esta plataforma permite que no solo dispositivos IoT se conecten al sistema. Ofrece también la posibilidad de que sistemas de generaciones previas aumenten sus capacidades siendo acoplados a ThingsBoard. Mediante la capacidad de integración de protocolos de comunicación industrial (como son OPC UA o ModBus) en ThingsBoard, equipos industriales cuyo ámbito de trabajo está limitado a la instalación industrial pueden hacer llegar sus datos a niveles superiores. De esta manera, nueva información, en contacto directo con el proceso, estará disponible para ser utilizada en la mejora de la supervisión o del desempeño global de la instalación.

Todo ello se gestiona desde una interfaz web clara y sencilla que permite visualizar los valores deseados de la instalación, así como la supervisión mediante el establecimiento de reglas o la generación de alarmas.

No menos importante es la sencillez de despliegue de la plataforma que permite ser instalada en dispositivos con un hardware modesto y con un proceso de instalación relativamente sencillo y asequible para un usuario con conocimientos reducidos de informática.

ThingsBoard tiene ventajas y alguna de ella también es la causa de alguno de sus inconvenientes. El hecho de ser una plataforma de código abierto hace que su documentación no sea todo lo completa que debería o que no haya un soporte directo que pueda ayudar en la resolución de problemas. Además, alguno de sus componentes presenta problemas de madurez como se ha podido ver a lo largo de este Trabajo Fin de Máster.

Finalmente, este proyecto presenta muchas oportunidades y actividades de seguir avanzando en un futuro puesto que son gran cantidad de aplicaciones que tiene en la industria. En el siguiente subapartado se recogen algunas de las más importantes.

6.1. LÍNEAS FUTURAS

Tras la realización de este Trabajo Fin de Máster y de haber conocido con cierto detalle la plataforma ThingsBoard, son muchas las opciones que esta posibilita y plantea tareas interesantes a realizar para poder aprovechar sus capacidades al máximo y lograr un conocimiento más completo de la plataforma.

Algunas de estas actividades serían:

- Conectar la plataforma a una instalación industrial real.
- Una de las partes más potentes dentro de ThingsBoard es el denominado *Rule Engine*. Tiene multitud de acciones y opciones posibles, como el envío de email o diferentes filtrados. Sería interesante realizar más pruebas para lograr un conocimiento mayor de las opciones que ofrece.
- Aumentar el número de protocolos de comunicación de interés industrial conectados a la plataforma. También sería interesante realizar el desarrollo

y configuración del *Custom Connector* del IoT Gateway para acoplar protocolos de comunicación industriales actualmente no soportados.

- Probar y utilizar las interfaces de extracción de datos, tanto la suscripción a WebSockets o las peticiones API REST. Esta parte es interesante para poder tener acceso a la información almacenada y poder aplicar diferentes análisis.
- Realizar el proceso de comunicación inverso e insertar las conclusiones extraídas a la plataforma y transmitir órdenes a diferentes entidades de campo conectadas a ThingsBoard.
- Acceder de manera remota a la instalación sin VPN, aumentando el control sobre los usuarios creados en la plataforma y utilizando técnicas de seguridad más avanzadas que permitan protegerse de amenazas externas [31].

Bibliografía

- [1] K. Schwab, La cuarta revolución industrial, Debate, 2016.
- [2] Deloitte. "¿Qué es la Industria 4.0?"
<https://www2.deloitte.com/es/es/pages/manufacturing/articles/que-es-la-industria-4.0.html>. (accedido 11 jul., 2022).
- [3] IoT Oracle. "¿Qué es el IoT?" <https://www.oracle.com/es/internet-of-things/what-is-iot/>. (accedido 11 jul., 2022).
- [4] B. H. J. C. T. W. Hugh Boyes, «The industrial internet of things (IIoT): An analysis framework,» *Computers in Industry*, vol. 101, pp. 1-12, 2018.
- [5] Power BI (Microsoft). <https://powerbi.microsoft.com/es-es/>. (accedido 12 jul., 2022).
- [6] Azure IoT. <https://azure.microsoft.com/es-es/solutions/iot/#products>. (accedido 12 jul., 2022).
- [7] Azure. "IoT Edge arquitectura" <https://docs.microsoft.com/es-es/azure/iot-edge/iot-edge-as-gateway?view=iotedge-2020-11>. (accedido 12 jul., 2022).
- [8] IoT AWS. <https://aws.amazon.com/es/iot/>. (accedido 12 jul., 2022).
- [9] AWS. "IoT Core". <https://aws.amazon.com/es/iot-core/?c=i&sec=srv>. (accedido 12 jul., 2022).
- [10] IoT Oracle. <https://www.oracle.com/es/internet-of-things/>. (accedido 12 jul., 2022).
- [11] IBM Watson. <https://www.ibm.com/es-es/watson>. (accedido 12 jul., 2022).
- [12] IBM Watson. "IoT, IBM Cloud". <https://internetofthings.ibmcloud.com/>. (accedido 12 jul., 2022).
- [13] Mind Sphere Siemens
<https://new.siemens.com/es/es/productos/software/mindsphere.html>. (accedido 12 jul., 2022).
- [14] EcoStruxure Schneider Electric
<https://www.se.com/es/es/work/campaign/innovation/overview.jsp>. (accedido 12 jul., 2022).

- [15] IoT Gateway ABB
<https://new.abb.com/products/robotics/es/controladores/iot>. (accedido 12 jul., 2022).
- [16] Arduino Cloud. <https://docs.arduino.cc/arduino-cloud/>. (accedido 13 jul., 2022).
- [17] Zetta. <https://www.zettajs.org/>. (accedido 13 jul., 2022).
- [18] ThingsBoard. <https://thingsboard.io/>. (accedido 10 mar., 2022).
- [19] AMQP. <https://www.amqp.org/>. (accedido 11 jul., 2022).
- [20] MQTT. "The standard of IoT messaggin". <https://mqtt.org/>. (accedido 15 feb., 2022).
- [21] CoAp. "Constrained Application Protocol". <https://coap.technology/>. (accedido 11 jul., 2022).
- [22] HTTP. "Hypertext Transfer Protocol (Mozilla)".
<https://developer.mozilla.org/es/docs/Web/HTTP>. (accedido 11 jul., 2022).
- [23] ModBus. "ModBus Organization". <https://modbus.org/>. (accedido 20 abril, 2022).
- [24] M. ORG, ModBus Application Protocol Specification v1.1b3, ModBus Organization, 2012.
- [25] «OPC-UA - OPC Unified Architecture,» <https://opcfoundation.org/about/opc-technologies/opc-ua/>. (accedido 20 abr., 2022).
- [26] UNE, UNE-CLC IEC/TR 62541. Aquitectura Unificada OPC, 2021.
- [27] IoT Gateway. <https://thingsboard.io/docs/iot-gateway/>. (accedido 10 mar., 2022).
- [28] ModRSim2. "Simulador ModBus"
<https://sourceforge.net/projects/modrssi2/>. (accedido 15 mar., 2022).
- [29] Prosys. "OPC UA Server". <https://www.prosysopc.com/products/opc-ua-simulation-server/>. (accedido 5 may., 2022).
- [30] GitHub. "Incidencia almacenado IoT Gateway".
<https://github.com/thingsboard/thingsboard-gateway/issues/856>. (accedido 21 jun., 2022).

- [31] P. S. L. H. Tibor Horak, «The Vulnerability of the Production Line Using Industrial IoT Systems under DDoS Attack,» *Electronics (Switzerland)*, vol. 10, nº 4, pp. 1-32, 2021.
- [32] CAN. "Controller Area Network". <https://www.can-cia.org/>. (accedido 11 jul., 2022).
- [33] LwM2M. "Light weight M2M". <https://omaspecworks.org/what-is-oma-specworks/iot/lightweight-m2m-lwm2m/>. (accedido 11 jul., 2022).
- [34] SNMP. "Secure Network Management Protocol". <https://www.snmp.com/>. (accedido 11 jul., 2022).
- [35] J. C. R. M. P. Ismael Essop, «Generating Datasets for Anomaly-Based Intrusion Detection Systems in IoT and Industrial IoT Networks,» *Sensors*, vol. 21, nº 4, pp. 1-31, 2021.
- [36] B. Z. M. B. Paul Pop, «The FORA Fog Computing Platform for Industrial IoT,» *Information Systems*, vol. 98, 2021.
- [37] Oasis, MQTT Specification v5.0, Oasis foundation, 2019.

Anexos

ANEXO 1 - FICHERO DE CONFIGURACIÓN THINGSBOARD

“thingsboard.conf”

```
# Copyright © 2016-2022 The Thingsboard Authors
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#   http://www.apache.org/licenses/LICENSE-2.0
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
export JAVA_OPTS="$JAVA_OPTS -Dplatform=deb -
Dinstall.data_dir=/usr/share/thingsboard/data"
export JAVA_OPTS="$JAVA_OPTS -
Xlog:gc*,heap*,age*,safepoint=debug:file=/var/log/thingsboard/gc.log:time,uptime,l
evel,tags:filecount=10,filesize=10M"
export JAVA_OPTS="$JAVA_OPTS -XX:+IgnoreUnrecognizedVMOptions -
XX:+HeapDumpOnOutOfMemoryError"
export JAVA_OPTS="$JAVA_OPTS -XX:-UseBiasedLocking -XX:+UseTLAB -
XX:+ResizeTLAB -XX:+PerfDisableSharedMem -XX:+UseCondCardMark"
export JAVA_OPTS="$JAVA_OPTS -XX:+UseG1GC -XX:MaxGCPauseMillis=500
-XX:+UseStringDeduplication -XX:+ParallelRefProcEnabled -
XX:MaxTenuringThreshold=10"
export LOG_FILENAME=thingsboard.out
export
LOADER_PATH=/usr/share/thingsboard/conf,/usr/share/thingsboard/extensions
export SQL_DATA_FOLDER=/usr/share/thingsboard/data/sql

# DB Configuration
export DATABASE_TS_TYPE=sql
export
SPRING_JPA_DATABASE_PLATFORM=org.hibernate.dialect.PostgreSQLDialect
export SPRING_DRIVER_CLASS_NAME=org.postgresql.Driver
export
SPRING_DATASOURCE_URL=jdbc:postgresql://localhost:5432/thingsboard
export SPRING_DATASOURCE_USERNAME=postgres
export SPRING_DATASOURCE_PASSWORD=psql
# Specify partitioning size for timestamp key-value storage. Allowed values: DAYS,
MONTHS, YEARS, INDEFINITE.
export SQL_POSTGRES_TS_KV_PARTITIONING=MONTHS
```

ANEXO 2 - FICHERO DE CONFIGURACIÓN IOT GATEWAY

“tb_gateway.yaml”

```
thingsboard:
  host: localhost
  port: 1883
  remoteShell: false
  remoteConfiguration: false
  statistics:
    enable: false
    statsSendPeriodInSeconds: 60
    configuration: statistics.json
  minPackSendDelayMS: 10
  checkConnectorsConfigurationInSeconds: 60
  handleDeviceRenaming: true
  checkingDeviceActivity:
    checkDeviceInactivity: false
    inactivityTimeoutSeconds: 120
    inactivityCheckPeriodSeconds: 10
  security:
    accessToken: q61heI7Y1AaG0PV5nEtq
  qos: 1
storage:
  type: sqlite
  data_file_path: ./data/data.db
  messages_ttl_check_in_hours: 1
  messages_ttl_in_days: 7
grpc:
  enabled: false
  serverPort: 9595
  keepaliveTimeMs: 10000
  keepaliveTimeoutMs: 5000
  keepalivePermitWithoutCalls: true
  maxPingsWithoutData: 0
  minTimeBetweenPingsMs: 10000
  minPingIntervalWithoutDataMs: 5000
connectors:
  -
    name: Modbus Connector
    type: modbus
    configuration: modbus.json
```

ANEXO 3 - FICHERO DE CONFIGURACIÓN CONECTOR MODBUS IOT GATEWAY

“modbus.json”

```
{
  "master": {
    "slaves": [
      {
        "host": "192.168.1.56",
        "port": 502,
        "type": "tcp",
        "method": "socket",
        "timeout": 35,
        "byteOrder": "BIG",
        "wordOrder": "BIG",
        "retries": true,
        "retryOnEmpty": false,
        "retryOnInvalid": false,
        "pollPeriod": 10000,
        "unitId": 1,
        "deviceName": "MBTEST",
        "deviceType": "default",
        "sendDataOnlyOnChange": true,
        "connectAttemptTimeMs": 5000,
        "connectAttemptCount": 5,
        "waitAfterFailedAttemptsMs": 300000,
        "timeseries": [
          {
            "tag": "16int_read",
            "type": "16uint",
            "functionCode": 3,
            "objectsCount": 1,
            "address": 1
          }
        ]
      }
    ]
  }
}
```

ANEXO 4- FICHERO DE CONFIGURACIÓN CONECTOR OPC UA IOT GATEWAY

opcua.json:

```
{
  "server": {
    "name": "OPC-UA Test Server",
    "url": "192.168.1.33:53530/OPCUA/SimulationServer/",
    "timeoutInMillis": 5000,
    "scanPeriodInMillis": 5000,
    "disableSubscriptions": false,
    "subCheckPeriodInMillis": 100,
    "showMap": false,
    "security": "Basic128Rsa15",
    "identity": {
      "type": "anonymous"
    }
  },
  "mapping": [
    {
      "deviceNodePattern": "Root\\Objects\\Simulation",
      "deviceNamePattern": "Simulated OPC-UA Device",
      "attributes": [
        {
          "key": "CertificateNumber",
          "path": "${ns=3;i=1007}"
        }
      ],
      "timeseries": [
        {
          "key": "level",
          "path": "${ns=3;i=1008}"
        }
      ]
    }
  ]
}
```