



Ingeniaritza Goi Eskola Teknikoa
Escuela Técnica Superior de Ingeniería
Bilbao

eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

TRABAJO DE FIN DE MASTER DE

Diseño y Despliegue de arquitectura
para la recogida y presentación de
medidas de sensores IoT

Alumno: Mikel De Prado Liceaga
Curso: 2016-2017

Directora: María Victoria Higuero Aperribay

Contenido

Abstract	1
1. Introducción	2
2. Contexto	4
3. Objetivos y Especificaciones del Trabajo	6
3.1. Objetivo Principal.....	6
3.2. Especificaciones generales	7
4. Beneficios.....	8
4.1. Beneficios Técnicos	8
4.2. Beneficios Sociales.....	8
4.3. Beneficios Económicos	8
5. Estado del Arte	10
6. Análisis de Alternativas	13
6.1. Tecnologías de comunicación de los dispositivos.....	13
6.1.1. Sigfox.....	13
6.1.2. LoRa/LoRaWAN.....	14
6.1.3. Criterios de selección	15
6.1.4. Selección	15
6.2. Recogida de información a través de la red	16
6.2.1. Red LoRaWAN propia.....	16
6.2.2. LORIOT	16
6.2.3. The Things Network	17
6.2.4. Criterios de selección	17
6.2.5. Selección	18
6.3. Protocolos de comunicación.....	18
6.3.1. MQTT	18
6.3.2. CoAP.....	19
6.3.3. Criterios de selección	20
6.3.4. Selección	21
6.4. Almacenamiento y representación de datos	21
6.4.1. Base de datos relacional.....	21
6.4.2. Elasticsearch, Logstash y Kibana (ELK)	22
6.4.3. Criterio de selección.....	22
6.4.4. Selección	23
7. Análisis de Riesgos	24
7.1. Identificación de Riesgos	24
7.1.1. Riesgos externos para el desarrollo de la maqueta	24

7.1.2.	Riesgos de la implementación.....	24
7.1.3.	Riesgos en Planificación	24
7.2.	Análisis de Riesgos	25
7.3.	Planificación de la respuesta	25
8.	Metodología	26
8.1.	Descripción de las Tareas	26
8.1.1.	Equipo de Trabajo	26
8.1.2.	Paquetes de trabajo y Tareas	26
8.2.	Diagrama de Gantt.....	28
9.	Descripción de la solución	29
9.1.	Diseño de la arquitectura.....	29
9.2.	Diseño y Desarrollo de la maqueta.....	32
9.2.1.	Arquitectura de recogida, almacenamiento y representación de datos	34
9.2.2.	Recogida de datos mediante dispositivos LoRa a través de TTN.....	34
9.2.2.1.	Configuración en TTN y gateway	35
9.2.2.2.	Configuración sensores TTN.....	37
9.2.3.	Lectura de datos contador eléctrico mediante Modbus.....	37
9.2.4.	Lectura de datos contador eléctrico mediante Modbus TCP	38
9.2.5.	Presentación de datos mediante Kibana.....	39
9.2.5.1.	Instalación de Plugins para Kibana	39
9.2.5.2.	Diseño de paneles de control.....	41
9.3.	Seguridad.....	43
9.3.1.	Separación de equipos generadores y consumidores de datos	43
9.3.2.	Disponibilidad del servicio	44
9.3.3.	Confidencialidad e Integridad de los datos	44
9.3.4.	Mecanismos de cifrado empleados en la maqueta	44
10.	Coste del TFM	46
10.1.	Coste Recursos Materiales.....	46
10.1.1.	Coste equipos	46
10.1.2.	Material amortizable.....	47
10.1.3.	Gastos	47
10.2.	Coste Recursos Humanos	47
10.3.	Costes Totales	48
11.	Conclusiones	49
	Fuentes de Información.....	50
	ANEXO 1 Pliego de condiciones. Normativa aplicable	51
	ANEXO 2 Planos y esquemas	52

Diagrama de la arquitectura	52
Diagrama de la maqueta.....	53
ANEXO 3 Código y ficheros de configuración	55

Índice de Figuras

Figura 1 Esquema caso de uso AWS IoT	10
Figura 2 Esquema de Azure IoT services.....	11
Figura 3 Esquema servicios IoT Google Cloud Platform	12
Figura 4 Esquema de funcionamiento MQTT	19
Figura 5 Diagrama Gantt.....	28
Figura 6 Diagrama arquitectura general.....	30
Figura 7 Diagrama de la arquitectura en base a las tecnologías seleccionadas en el análisis de alternativas	31
Figura 8 Clúster Elasticsearch y Shards.....	32
Figura 9 Diagrama de la maqueta	33
Figura 10 Diagrama recogida de datos LoRaWAN	35
Figura 11 Diagrama Infraestructura TTN	35
Figura 12 Esquema reenvió TTN.....	36
Figura 13 Lectura datos mediante Modbus.....	38
Figura 14 Lectura datos mediante ModbusTCP	39
Figura 15 Plugin Gauge	40
Figura 16 Plugin custom-gauge	40
Figura 17 Plugin traffic	40
Figura 18 Panel de control nivel de CO2	41
Figura 19 Panel de control temperatura	41
Figura 20 Panel de control balizas GPS.....	42
Figura 21 Panel de control medidor eléctrico	42
Figura 22 Panel de control estadísticas LoRaWAN	43
Figura 23 Claves cifrado LoRaWAN	45
Figura 24 Diagrama de arquitectura de recogida, almacenamiento y presentación.....	52
Figura 25 Diagrama de la maqueta	54
Figura 26 Diagrama de ficheros del TFM.....	55

Índice de Tablas

Tabla 1 Evaluación de tecnologías de comunicación SigFox vs LoRa/LoRaWAN	15
Tabla 2 Evaluación de alternativas para recogida de información a través de la red.....	18
Tabla 3 Evaluación alternativas protocolo de comunicación.....	20
Tabla 4 Evaluación de alternativas de almacenaje y representación de datos	23
Tabla 5 Análisis cualitativo-cuantitativo delos riesgos del proyecto	25
Tabla 6 Equipo de Trabajo	26
Tabla 7 Lista de acrónimos	26
Tabla 8 Hitos	27
Tabla 9 Coste Equipos.....	46
Tabla 10 Coste Material Amortizable	47
Tabla 11 Gastos	47
Tabla 12 Coste Recursos Humanos.....	47
Tabla 13 Costes Totales.....	48

Diseño y Despliegue de arquitectura para la recogida y presentación de medidas de sensores IoT

Abstract

Actualmente, un número creciente de dispositivos y equipos se conectan a Internet, cada vez es más común encontrarse con sensores que envían información a través de la red o empresas que quieren monitorizar su producción remotamente. Por ello, en la Industria 4.0 y en el ámbito de las Smartcities es cada vez más importante la recogida de los datos de sensores del denominado IoT.

Debido a ello, en este contexto, este Trabajo Fin de Master (TFM) propone un diseño de una arquitectura que permita recoger y almacenar los datos de sensores IoT de diversos ámbitos, comerciales e industriales, y proporcione herramientas para una presentación sencilla y comprensible por cualquier usuario de estos datos. Adicionalmente en este documento se analizará la viabilidad de esta arquitectura mediante el diseño y la implementación de una maqueta, que haga uso de este diseño utilizando algunas de las tecnologías que comienzan a cobrar importancia en el ámbito del IoT.

Nowadays a growing number of devices and equipment connect to the Internet; it is increasingly common to find sensors that send information through the network or companies that want to monitor their production remotely. For this reason, in the Industry 4.0 and in the scope of Smartcities, it is increasingly important to collect the data of sensors of the so-called IoT.

Because of this, in this context, this End of Master work proposes an architecture design that allows the gathering and storage of IoT sensor data from various commercial and industrial environments, and provides tools for a simple and comprehensible presentation of the data for any user. Additionally, in this document will be analysed the feasibility of this architecture, through the design and implementation of a model that makes use of this design by means of some of the technologies that begin to acquire importance in the scope of the IoT.

Gaur egun, gailu eta ekipo kopuru handiagoa konektatzen da Internetera, gero eta ohikoagoa da topatzea sarearen bidez informazioa bidaltzen duten sentsoreekin edo urrunetik haien ekoizpenaren jarraipena egin nahi duten enpreseekin. Ondonoz, 4.0 Industrian eta Smartcity-en eremuan gero eta garrantzitsuagoa da IoT sentsoreen datuen bilketa.

Hori dela eta, testuinguru honetan, Master Amaierako Lan (MAL) honek eremu ezberdinetako (komertziala eta industrial) IoT sentsoreen datuen bilketa eta biltegiketa ahalbideratzen diseinu honek tresna ezberdinak eskaini behar ditu edozein erabiltzailek datu horitako aurkezpen soil eta ulergarria jaso dezan. Horrez gain, dokumento honetan arkitektura honen bideragarritasuna aztertuko da. Horretarako diseinu hau erabiltzen duen maketa bat diseinatu eta garatuko da, IoT eremuan gero eta garrantzi handiago duten teknologiak erabiliz.

1. Introducción

Durante los últimos años, se ha producido un crecimiento espectacular del número de dispositivos que se conectan a Internet. Estos dispositivos han aumentado en cantidad y variedad; desde televisores hasta los denominados smartwatches. Esto ha propiciado que se acuñe el término de Internet de las Cosas (IoT), en inglés Internet of Things, que hace referencia a la interconexión de multitud de objetos a Internet.

Esta interconexión de dispositivos a Internet, no se limita solo a dispositivos de usuarios particulares; en realidad, este uso no sería el mayoritario cuando se habla del IoT, sino que se extiende a las industrias y a las Smartcities. De hecho, el verdadero despliegue del IoT se observaría en las ciudades inteligentes (Smartcities) y en la denominada Industria 4.0, donde las máquinas se conectan a Internet para enviarse información entre ellas. Este concepto de comunicaciones de máquinas con máquinas, M2M (Machine to Machine), no es nuevo, pero vuelve a cobrar importancia en el contexto del IoT.

Tanto en las Smartcities como en la Industria 4.0 los sensores adquieren una gran importancia. Estos sensores deben poder recabar información y enviarla a través de Internet. La Industria 4.0 y las Smartcities mediante estos sensores serán capaces de funcionar de una manera más autónoma y permitirán una monitorización más eficaz de todos los procesos y situaciones que se den gracias a que van a poder disponer de toda la información de forma centralizada y accesible de forma simple a través de Internet. Esto reportará grandes beneficios tanto económicos, como sociales a las empresas y a la sociedad en general.

Partiendo de este escenario el IoT está cobrando cada vez más importancia en el entorno empresarial y en el de las telecomunicaciones, como se puede apreciar por las nuevas empresas que han surgido o aquellas que ya existían que empiezan a invertir en las tecnologías asociadas al concepto de IoT. Si se atiende, por ejemplo, a las últimas ediciones del Mobile World Congress [15] se puede ver que hay una tendencia creciente hacia el IoT.

Por otro lado, como cabría esperar, la información generada por esta gran cantidad de sensores es masiva por lo que se convertirá en una necesidad en algunos casos la aplicación de técnicas de análisis y procesamiento de datos basadas en BigData, para procesar y analizar los datos que se generen a lo largo del tiempo por una Smartcity o por una empresa Industrial.

Así mismo, los nuevos sensores y dispositivos del IoT, hacen uso y propulsan el desarrollo de nuevas tecnologías, sobre todo en el campo de las telecomunicaciones: protocolos, modulaciones, etc. Esto ocurre en particular en dispositivos de difícil acceso o que no pueden estar cableados. Algunas de las más conocidas son tecnologías de comunicación como Zigbee, LTE, LoRa, etc. y protocolos de comunicación como MQTT.

A su vez, los datos generados por estos dispositivos y sensores serán en algunos casos sensibles o críticos por lo que será necesario utilizar mecanismos de seguridad adecuados, que se encarguen de mantener la confidencialidad e integridad de los mismos. Por ejemplo, datos de consumo eléctrico mediante los cuales se pueda inferir cuándo hay personas en una localización, como podría ser el caso de los contadores eléctricos utilizados por una empresa de distribución de energía, así como los datos relacionados con el contexto del eHealth, conjunto de Tecnologías de la Información y la Comunicación (TICs) que, a modo de herramientas, se emplean en el entorno sanitario en materia de prevención, diagnóstico, tratamiento, seguimiento, etc.

En este contexto se enmarca este Trabajo de Fin de Master (TFM) que se centrará precisamente en el diseño y despliegue de una maqueta de un sistema de recogida de datos de IoT desarrollada en la empresa Sarnet S.A.U, que permitirá recoger datos de diversos sensores y los enviarán utilizando diferentes tecnologías para su almacenamiento y representación de forma sencilla e intuitiva, asegurando la confidencialidad y la integridad de los datos.

Este TFM ha sido desarrollado en la empresa Sarenet S.A.U, que es una empresa proveedora de conectividad a Internet, telefonía IP y servicios Cloud y se enmarca en sus trabajos de constante innovación en la oferta de servicios a sus clientes, en particular, empresas relacionadas con las TIC y la industria entre otras.

2. Contexto

Tal y como se acaba de indicar en el apartado anterior, existe una creciente necesidad de desarrollar plataformas, sistemas, protocolos y tecnologías para permitir la comunicación y recolección de datos del creciente número de dispositivos conectados a Internet. Por ello en la actualidad diversas empresas y entidades están desarrollando tecnologías y protocolos enfocados a aquellos servicios relacionados con el Internet de las Cosas (IoT) y muchas empresas comienzan a invertir en equipos y sistemas que utilicen dichas tecnologías. No solo para las denominadas Smartcities sino para la que se ha denominado como la 4ª revolución industrial (Industria 4.0), que será una de las mayores impulsoras y principal nicho de negocio para muchas empresas.

De la misma forma, muchas empresas de telecomunicaciones comienzan a ofrecer servicios de IoT. En particular, los operadores ofrecen principalmente servicios de comunicaciones basados en la red de telefonía móvil (3G/4G) para IoT. Sin embargo, esta tecnología no es muy apropiada para estos escenarios debido al consumo de potencia elevado que implica, que para los dispositivos que pueden estar en lugares de difícil acceso o sin acceso a la red eléctrica es una gran desventaja. Y además la red móvil es un servicio tarifado, lo que provoca un aumento del coste de los servicios que utilicen la red móvil. En la actualidad se está desarrollando un estándar de telefonía móvil para IoT por parte del grupo de organizaciones de telecomunicaciones conocido como 3GPP, denominado NB-IoT, para hacer frente al problema de consumo de batería.

Paralelamente, otras organizaciones han desarrollado otras tecnologías de baja potencia y área de cobertura grande (LPWAN). Entre ellas destacan SIGFOX [2] que es una modulación de Banda Ultra Estrecha, desarrollada por la compañía del mismo nombre, y LoRaWAN [1] que es otra modulación basada en espectro expandido y un protocolo de nivel de acceso para esta modulación, desarrollada por LoRa-Alliance [11].

Adicionalmente, no solo las tecnologías de comunicación orientadas al IoT están avanzando, también los protocolos de comunicación extremo a extremo, como MQTT [3], cobran gran importancia y se vuelven herramientas esenciales al ofrecer una forma de transportar la información entre productores de información (dispositivos y maquinas) y consumidores (Bases de datos, Interfaces de monitorización, etc.) de forma segura, aislando a unos de otros. Esto impide que desde uno de los elementos se pueda atacar al otro directamente, lo cual en industria podría suponer pérdidas económicas si se consiguiera atacar a equipos de producción o a aquellos que controlan la producción.

Como se ha mencionado en el párrafo anterior, la seguridad es un factor muy importante, principalmente en el sector industrial, el cual es muy receloso de emplear tecnologías que puedan volver vulnerable su infraestructura de producción o que puedan facilitar la obtención de información confidencial. Por ello, es necesario aplicar mecanismos de seguridad como encriptación de datos y aislar los elementos que se comunican entre sí, para que en caso de producirse un ataque al sistema no se comprometan todos los elementos.

En este contexto cobra también una gran importancia el análisis y almacenamiento de la información recogida y generada por los diferentes dispositivos. La gran cantidad de información con características y estructuras diversas requiere bases de datos escalables y distribuidas. Entre las tecnologías de bases de datos actuales cabe destacar como una buena posibilidad las herramientas ELK [4] (ElasticSearch, Logstash y Kibana), ya que son de código abierto y ofrecen una serie de funcionalidades que facilitan mucho la representación de los datos.

La empresa Sarnet, proveedora de conectividad a Internet, telefonía IP y servicios Cloud, ha comenzado a comercializar soluciones de IoT basadas en Software Libre con una infraestructura IoT gestionada y securizada para las comunicaciones y el alojamiento orientado a la Pyme industrial. En base a este nuevo

servicio se está desarrollando una maqueta de servicios IoT para analizar nuevas posibilidades de este nuevo servicio IoT.

En el seno de esta empresa, es donde se desarrolla este Trabajo de Fin de Master (TFM) en el que se diseñará y desarrollará una arquitectura mediante la cual se recabarán, almacenarán y presentarán los datos de diversos sensores. Además se diseñarán y realizarán una serie de prueba de campo sobre una maqueta real mediante medidas de calidad de aire en oficinas, consumo eléctrico y control de flotas mediante balizas GPS.

3. Objetivos y Especificaciones del Trabajo

En este apartado se expondrán los objetivos y especificaciones que se deberán cumplir en este TFM.

3.1. Objetivo Principal

El objetivo principal de este trabajo es el diseño y despliegue de una arquitectura para la recogida de datos, almacenamiento y la presentación de medidas realizadas por sensores en el ámbito del IoT. Esta arquitectura está orientada a la recogida de datos en el entorno del IoT, así como en la denominada Industria 4.0, y por ello la arquitectura deberá de ser capaz de recoger, almacenar y presentar datos tanto de sensores comerciales, como de sensores y maquinaria industrial. Como objetivo parcial de este proyecto se deberá analizar la viabilidad de la arquitectura sugerida.

Para llevar a cabo este objetivo principal se establecen una serie de objetivos secundarios que deberán cumplirse para el éxito del TFM. Estos objetivos se refieren al diseño de la arquitectura general:

- Un diseño de arquitectura segura para la recogida de los datos, asegurando la integridad y confidencialidad de los datos, así como y más importante, aislar a los equipos productores de datos de los consumidores de estos.
- El sistema de almacenamiento para los datos y el sistema de presentación de los datos deberán permitir presentar los datos almacenados de una forma sencilla y comprensible para los interesados.

Con el fin de cumplir el objetivo del diseño de una arquitectura segura para la recogida de datos, se deberán de cumplir los siguientes objetivos parciales relacionados con la seguridad en el diseño:

- Se deberá asegurar la disponibilidad del servicio proporcionado por esta arquitectura.
- La arquitectura deberá implementar mecanismos que aseguren la integridad de los datos en su envío hacia la arquitectura.
- También se deberá asegurar la confidencialidad de los datos en el envío.
- Por último, se deberán aplicar mecanismos que aseguren que los datos recibidos, se hayan generado recientemente y no sean datos antiguos reenviados.

Para el cumplimiento del objetivo principal y para el análisis de la viabilidad de la arquitectura se diseñará y desplegará una maqueta en la que se analizarán las capacidades y funcionalidades de la arquitectura, la cual tiene que cumplir los siguientes objetivos parciales:

- Diseño de maqueta para la recogida de datos de diferentes dispositivos y fuentes del ámbito del IoT, tanto equipos comerciales a través de una red **LPWAN (Low Power Wide Area Network)**, como equipos industriales a través de un bus de comunicación industrial.
- Despliegue de infraestructura para la recogida, almacenamiento y representación de los datos de los distintos dispositivos de la maqueta.
- Diseño de paneles de control, simples y comprensibles, que permitan a usuarios poco familiarizados con el ámbito de los datos representados ser capaces de monitorizar y percibir la situación actual con el fin de detectar problemas en el funcionamiento o situaciones anómalas.
- Adicionalmente, analizar la capacidad de la arquitectura de recoger datos a través de un protocolo de comunicación industrial mediante un equipo industrial, en particular con el protocolo Modbus, y su envío a través de la red IP, para su almacenamiento y monitorización. Se deberá hacer uso de un protocolo que permita aislar al equipo industrial que recoge los datos de la infraestructura que los almacenará y los visualizará, para que en caso de que uno de ellos este comprometido no pueda atacarse a través del otro.

3.2. Especificaciones generales

A continuación se presentarán las especificaciones generales de la arquitectura diseñada y desplegada en este TFM en la maqueta, que se deben cumplir para el éxito del mismo:

- El sistema de almacenamiento de los datos debe permitir almacenar cualquier tipo de dato que se le envíe sin requerir modificar el diseño de la base de datos y debe permitir realizar búsquedas con rapidez entre una gran cantidad de información.
- La representación de los datos debe ser simple e intuitiva, es decir, que un usuario pueda entender la situación sin necesidad de ser un experto.
- El backend que almacenará y representará los datos debe estar aislado de los sensores y equipos que produzcan la información para minimizar y asegurar la integridad de ambos.
- El Gateway LPWAN utilizado deberá ser configurable y capaz de conectarse a una red de la tecnología LPWAN utilizada.
- La arquitectura debe de ser capaz de recoger datos tanto de equipos comerciales como de equipos de carácter industrial. Ellos deberán contar con:
 - Sensores para realizar medidas o la capacidad para comunicarse con equipos industriales para la recogida de datos.
 - Interfaz de comunicación Ethernet o de forma inalámbrica con una tecnología LPWAN.
 - Ser configurables o programables.
- Asegurar la integridad y confidencialidad de los datos recogidos al enviarlos a través de la red mediante la encriptación de los mismos.

4. Beneficios

Este trabajo proporcionará una serie de beneficios técnicos, sociales y económicos. Estos beneficios serán de interés tanto para empresas como para la sociedad en general.

4.1. Beneficios Técnicos

Este trabajo proporciona un excelente punto de partida para futuros desarrollos, por parte de Sarenet, de otros servicios y proyectos para IoT, habiendo ya identificado los factores más relevantes para plataformas o sistemas similares acerca de aspectos como los dispositivos involucrados, los mecanismos de comunicación de dichos dispositivos y la recolección y análisis de los datos.

Adicionalmente se podrá conocer el comportamiento y las peculiaridades de estas tecnologías facilitando la utilización, desarrollo y mantenimiento de las mismas, con lo que se conseguirá agilizar y mejorar futuros desarrollos que las utilicen, como por ejemplo, en la recolección, almacenamiento y presentación de los datos mediante tecnologías como ELK [4], las cuales pueden tener usos muy diversos no solo en el entorno del IoT sino por ejemplo para almacenar y visualizar datos de monitorización de una red de telecomunicaciones.

A su vez, la familiarización con estas tecnologías permitirá contribuir al desarrollo y mejora de las mismas, con lo que se logrará en el futuro crear plataformas y sistemas más avanzados y eficaces.

4.2. Beneficios Sociales

Este trabajo proporciona beneficios de carácter social, ya que permitirá disfrutar de servicios que repercutan en su bienestar o salud, a aquellos interesados (empleados, inspectores, etc.) monitorizar la calidad del aire en su lugar de trabajo y tomar las medidas oportunas para mejorarlo, por ejemplo, abriendo una ventana para ventilar si hay una gran concentración de CO₂ en el ambiente, evitando la fatiga asociada a una elevada concentración de CO₂.

También proporcionará beneficios para la salud de los empleados y su seguridad por ejemplo, al monitorizar el CO en lugares cerrados donde haya motores de combustión o generadores, y se puedan producir acumulaciones peligrosas de CO.

Además podría ser aplicado en otros escenarios como eHealth, con las que se podrían realizar tareas de prevención y diagnóstico de posibles problemas relacionados con la salud y la seguridad laboral.

4.3. Beneficios Económicos

Este proyecto reportará una serie de beneficios económicos algunos de los cuales pueden ser inferidos de los subapartados anteriores.

Se podrá obtener beneficios económicos del desarrollo de plataformas IoT a partir de este proyecto y la posterior comercialización los servicios a otras empresas, así como del ahorro de tiempo y coste de desarrollo de plataformas que aprovechen parte o todas las tecnologías y herramientas utilizadas en este trabajo.

Por otro lado, las pruebas realizadas en la maqueta para el control de flotas permitirán a aquellos interesados monitorizar la utilización de recursos comunes de la empresa y su localización. Como por ejemplo, visualizar la localización de un coche de empresa para conocer su ubicación aproximada y conocer su disponibilidad.

Adicionalmente, las empresas o asociaciones que empleen plataformas de monitorización como esta o similares, obtendrán beneficios económicos del ahorro al disponer de este tipo de sistemas que les

ayudarán a mejorar la eficiencia de sistemas como el aire acondicionado o permitir detectar anomalías en la red eléctrica gracias a las medidas de contadores eléctricos, lo cual en caso de avería puede ayudar a detectar el fallo antes de que ocurra.

Adicionalmente las medidas obtenidas de los contadores eléctricos podrán ser contrastadas con las presentadas en la factura de la compañía eléctrica, con lo que se podrá verificar su correcta medida y además permitirá optimizar el gasto al poder planificar la utilización de recursos que consuman una mayor potencia a franjas horarias de menor coste. También habrá beneficios al aplicar a otros escenarios la arquitectura diseñada en este TFM.

5. Estado del Arte

El objetivo principal que se persigue en este TFM es el diseño y despliegue de una arquitectura para la recogida de datos mediante sensores, y su posterior representación, en escenarios IoT. A partir de esta arquitectura se desarrollará una maqueta que será utilizada posteriormente para la demostración de este tipo de servicios relacionados con el IoT por la empresa Sarenet, por lo cual en este estado del arte se estudiarán otras plataformas de servicios IoT existentes.

En cuanto a las tecnologías empleadas para este tipo de servicios, existen diversas tecnologías y plataformas que se han desarrollado o se están desarrollando en la actualidad. De algunas de estas tecnologías se hablará más en detalle en el análisis de alternativas de este documento. Los grandes grupos de tecnologías del IoT son los dispositivos, la tecnología de comunicación que emplean, los protocolos de comunicación, las plataformas de almacenamiento empleadas y las herramientas de representación empleadas.

En la actualidad existen varias plataformas que ofrecen un backend, es decir, un sistema de almacenamiento, análisis de los datos y en algunos casos la representación de ellos. Entre las empresas que ofrecen estos servicios se puede observar cómo los gigantes de Internet poseen sus propias plataformas.

- AWS IoT [8] de Amazon: AWS IoT es una plataforma de nube que permite a los dispositivos conectados interactuar con facilidad y seguridad con las aplicaciones en la nube y otros dispositivos. AWS IoT facilita la utilización de otros servicios propietarios de Google como AWS Lambda, Amazon Kinesis, Amazon S3, Amazon Elasticsearch service, Amazon Machine Learning, etc. Entre otras cosas estos servicios ofrecen la posibilidad de almacenar, analizar y visualizar los datos proporcionados por dispositivos y sensores IoT conectados a AWS IoT.

Esta plataforma de Amazon es un servicio tarificado en función del número de mensajes recibidos; cuantos más dispositivos y una frecuencia de envío mayor provocará un mayor coste de operación en cualquier proyecto que la utilice.

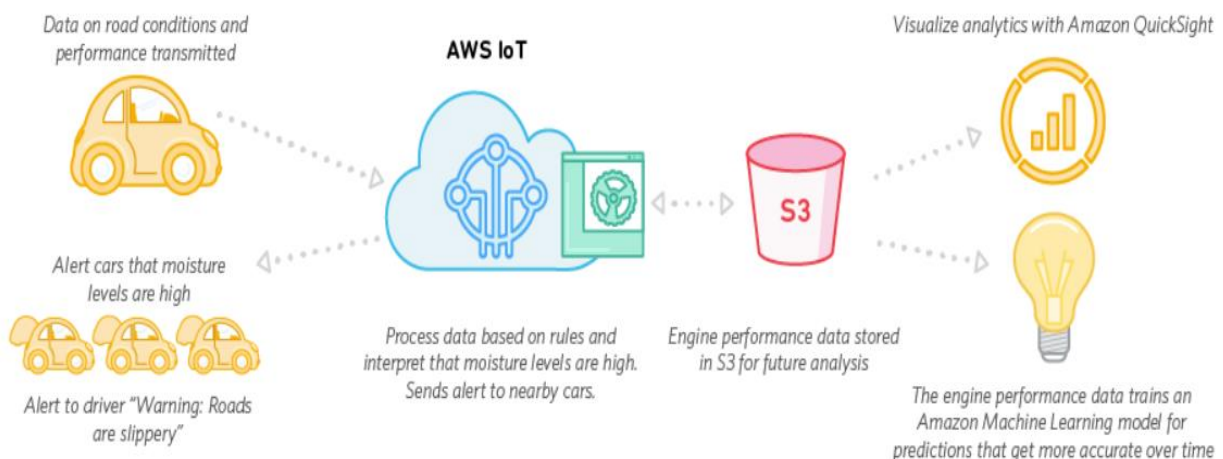


Figura 1 Esquema caso de uso AWS IoT

En la Figura 1 se puede observar un diagrama de los componentes de esta plataforma y su interacción para la solución AWS IoT. En ella se puede observar cómo sensores colocados en vehículos envían la información a AWS IoT, el cual los recibe y los almacena en el S3 de Amazon, y posteriormente estos pueden ser visualizados mediante Amazon QuickSight y se pueden obtener predicciones de los datos mediante un modelo de predicciones de Amazon basado en Machine Learning.

- Servicios IoT de Azure [9]: Azure de Microsoft ofrece varias funcionalidades diferentes orientadas al IoT, entre ellas recopilación de datos de dispositivos, análisis de flujos de datos, almacenamiento y consulta de los datos, representación de los datos, etc. Microsoft ofrece soluciones preconfiguradas que utilizan los diferentes servicios que componen la solución, o una solución personalizada con componentes y servicios específicos. Por defecto los componentes comunes de las soluciones son los siguientes:
 - Azure IoT Hub: Ofrece capacidades de mensajería entre los dispositivos y la nube, y sirve como puerta de enlace a los dispositivos con la nube. Adicionalmente permite la administración de dispositivos.
 - Azure Streaming Analytics: Permite realizar análisis de datos en movimiento, procesar la telemetría entrante, realizar la agregación y detectar eventos.
 - Azure Storage y Azure Cosmos DB: ofrece funcionalidades para almacenar la información.
 - Azure Web Apps y Microsoft Power BI: ofrecen funcionalidades de representación de datos.

La plataforma es configurable por el usuario seleccionando aquellos componentes que desea utilizar, y el coste de utilización es en base a las horas de utilización de los componentes y a las capacidades necesarias, así como al número de mensajes y dispositivos que envíen los datos. Por ello, como en la plataforma anterior, el coste aumentará cuanto mayor sea el número de dispositivos y la utilización de los componentes.

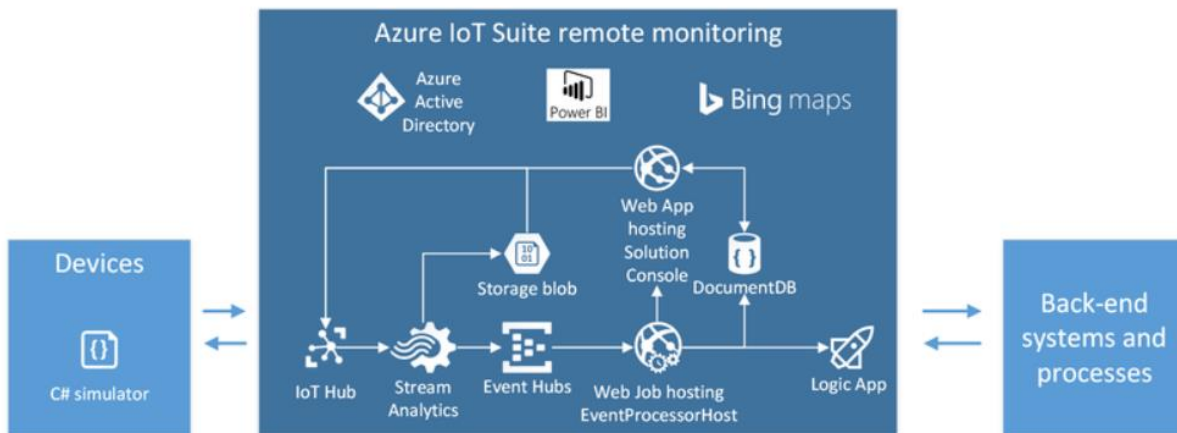


Figura 2 Esquema de Azure IoT services

En la Figura 2 se puede observar un diagrama con los componentes comunes de Azure IoT, así como la interacción entre ellos.

- Google Cloud Platform [10]: Google ofrece soluciones para el IoT desde su plataforma en la nube, aunque a diferencia de las dos plataformas anteriores no tiene un nombre propio como servicio, sino que ofrece los diferentes servicios de su plataforma Cloud para proporcionar el servicio. Ofrece servicios de adquisición de datos, servicios de almacenamiento, servicios analíticos y servicios de aplicación y presentación para proveer los servicios de backend necesarios para el IoT.

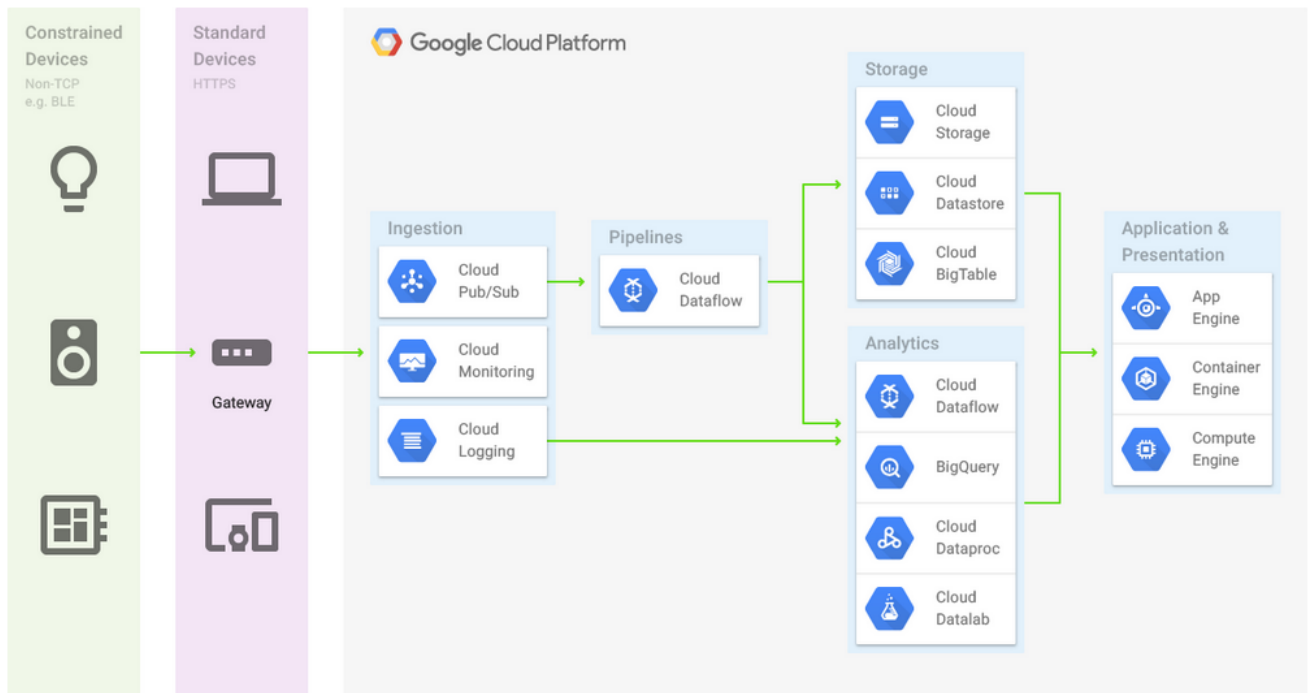


Figura 3 Esquema servicios IoT Google Cloud Platform

En la Figura 3 se puede observar un diagrama con los componentes de la plataforma cloud de Google para IoT.

Como en el caso de Azure y AWS el coste de operación de la plataforma de Google es en función de su utilización: a mayor volumen de dispositivos y mensajes, mayor coste.

6. Análisis de Alternativas

En este apartado se presentan los factores y tecnologías en las que se han valorado diferentes alternativas, y que se han considerado de interés y se han tenido en cuenta para el desarrollo de este TFM. Se explican las diferentes opciones y se comparan a fin de concluir qué alternativa se ha considerado la más adecuada para este TFM.

Principalmente este análisis se centrará en las tecnologías y los protocolos de comunicación y las diferentes formas de recoger, almacenar y presentar los datos obtenidos por esta maqueta de IoT.

6.1. Tecnologías de comunicación de los dispositivos

En el entorno de IoT existen diversas tecnologías para la comunicación de los dispositivos. Estas tecnologías son muy variadas, y en función del tipo de dispositivo y el tipo de información que se vaya a enviar algunas tecnologías serán más adecuadas que otras. Se podrían comunicar los dispositivos utilizando una gran variedad de tecnologías, desde la utilización de un simple cable Ethernet hasta la utilización de tecnologías inalámbricas como pueden ser Zigbee o Bluetooth, o incluso tecnologías móviles como LTE.

Principalmente si el dispositivo que se desea comunicar está conectado al suministro eléctrico y no se encuentra en un lugar de difícil acceso, como por ejemplo una montaña, lo habitual es utilizar tecnologías cableadas para comunicarlo. No obstante, en IoT no solo hay dispositivos de este tipo, como pueden ser los dispositivos clásicos que se conectan a Internet, sino que también hay pequeños gadgets o como en el caso de este proyecto, sensores que por su ubicación o debido a otros factores, requieren de tecnologías inalámbricas.

Muchos de estos sensores se alimentan de una batería, por lo que para comunicarse, y a fin de lograr que la batería dure lo máximo posible, requieren de tecnologías que supongan un consumo bajo de energía. En este proyecto se han tenido en cuenta estas tecnologías, entre las que podemos encontrar Zigbee y Bluetooth, pero ambas tienen un alcance limitado o requieren de una red de dispositivos para aumentar el alcance. Por ello este proyecto se ha concentrado más en las tecnologías denominadas **LPWAN (Low-Power Wide-Area Network)**. Las LPWAN son tecnologías inalámbricas que permiten comunicación a través de largas distancias con un consumo de energía pequeño. Las principales tecnologías LPWAN que se analizarán para este TFM son **Sigfox [2]** y **LoRa o LoRaWAN (Long Range WAN) [1]**, debido a que son las más extendidas en la actualidad.

6.1.1. Sigfox

Sigfox [2] es una tecnología de modulación de banda ultra-estrecha, y su ancho de banda del canal es de 200Hz. Utiliza un estándar de transmisión de radio BPSK, que hace uso de pequeñas franjas muy estrechas del espectro de radio y modifica la fase de la onda portadora para codificar la información. Esto permite que el receptor solo escuche en una pequeña franja del espectro, lo que mitiga el efecto del ruido.

Esta tecnología utiliza bandas de frecuencia de uso libre por lo que no es necesaria una licencia para transmitir y recibir mediante esta tecnología. Pero estas bandas son utilizadas por otras tecnologías, y por tanto tendrán un nivel de ruido mayor que las bandas reservadas. En particular, Sigfox utiliza la banda de 902 MHz en América y la banda de 868 MHz en Europa.

El largo alcance se consigue como resultado de enviar poca información y con un bajo bitrate. Para ilustrar esta afirmación, según los datos proporcionados por Sigfox la carga útil de un paquete puede tener un tamaño máximo de 12 bytes y se pueden transmitir un máximo de 140 mensajes al día, lo que supone la capacidad de enviar como máximo 1680 bytes al día. El alcance de transmisión que proporciona es de entre 3 y 10 Km. Adicionalmente, el consumo de potencia es muy reducido por lo que la duración de la batería del dispositivo se alarga considerablemente.

Debido a todas las características citadas, se puede decir que esta tecnología es adecuada para aplicaciones que necesiten enviar poca información o pequeñas ráfagas infrecuentes de datos, y por tanto, se considera adecuada para el envío de alarmas, y mediciones simples. Esto la convierte en una buena opción para la comunicación con sensores.

Para comunicarse mediante Sigfox los dispositivos requieren un chip compatible, el cual está libre de derechos y royalties, cualquier empresa puede fabricarlos. Esto ha provocado una reducción del precio de los dispositivos al permitir que cualquier fabricante pueda fabricar sus propios dispositivos sin necesidad de comprar los chips a un fabricante o pagar una licencia.

No obstante, no cualquier empresa puede proporcionar servicios de operador Sigfox ya que esto solo lo puede realizar la empresa Sigfox. Si se desea proporcionar cobertura a los dispositivos funcionando como operador será necesario llegar a un acuerdo con esta empresa. Y hay que tener en cuenta que se tarifica el servicio ofrecido por la red Sigfox, aunque el coste es mucho menor que el de un operador móvil.

6.1.2. LoRa/LoRaWAN

LoRa (Long Range) es un esquema de modulación de espectro expandido desarrollado por la LoRa Alliance, el cual permite un área grande de cobertura con un consumo de potencia reducido, a costa de un bajo bitrate. Esta modulación utiliza las bandas de libre uso de 915MHz en América y 868 MHz en Europa, bandas ISM [12] de uso libre iguales a las utilizadas por Sigfox.

LoRaWAN [1] (Long Range Wide Area Network) es un estándar de protocolo de nivel de enlace que utiliza el esquema de modulación LoRa. Este protocolo está optimizado para dispositivos alimentados por batería que pueden estar en movimiento o en un lugar de difícil acceso.

La comunicación entre los dispositivos y los gateways se realiza a través de la banda de 868 MHz (Europa), y mediante la utilización de diferentes data rates se puede ajustar el área de cobertura y la duración de la transmisión del mensaje.

Al utilizarse espectro expandido, las comunicaciones que utilizan data rates diferentes no se interfieren entre sí. Los data rates pueden ir desde 0.3 kbps hasta 50 kbps. Para maximizar la vida de la batería y la capacidad de la red el servidor de red LoRaWAN gestiona los data rates de los dispositivos, mediante la opción de un data rate adaptativo en los mismos.

Adicionalmente esta tecnología limita el uso de los canales para evitar sobrepasar los niveles de emisión y un nivel de ocupación del canal que evite interferencias en exceso; por ello en el documento del estándar, se define un ciclo de servicio de 1% por canal. En cuanto al área de cobertura LoRa ofrece un área de cobertura de aproximadamente 1 km en entorno urbano y de hasta 25 Km en el campo.

Las redes LoRaWAN tienen normalmente una topología de star-of-stars, en la que los gateways reenvían los mensajes entre los dispositivos finales y los servidores de la red en el back-end. Los gateways se conectan mediante conexiones IP, mientras que los dispositivos lo hacen mediante un único salto LoRa o FSK con uno o más Gateways. Todas las comunicaciones son bidireccionales aunque el canal de subida del dispositivo al gateway es el tráfico predominante.

El chip necesario para utilizar esta tecnología es fabricado solo por Semtech actualmente. No obstante el precio de los dispositivos con esta tecnología es bastante reducido. Adicionalmente, cualquier empresa o entidad puede montar su propia red LoRaWAN y proporcionar cobertura sin necesidad de licencias o permisos. Un ejemplo, de ello es **The Things Network (TTN)** [7] una red gratuita y libre en la que cualquier

persona puede agregar un gateway y utilizar su infraestructura para recibir datos de sus dispositivos LoRaWAN.

6.1.3. Criterios de selección

Los criterios de selección que se han tenido en cuenta para la selección de la tecnología de comunicación inalámbrica a utilizar han sido los siguientes:

- Capacidad de transmisión, área de cobertura y seguridad (40 %)
 - Bajo este criterio se evaluarán las características propias de cada una de las tecnologías.
- Complejidad de implementación y mantenimiento (30 %)
 - Bajo este criterio se valorará la complejidad de llevar a cabo el despliegue de esta solución, y más importante, la complejidad de mantenerla para su funcionamiento continuado.
- Coste dispositivos y equipos (30 %)
 - Se han de tener en cuenta los costes que conlleva cada una de las dos tecnologías, tanto a la hora del despliegue inicial como a la hora de mantener el sistema o el coste de los propios dispositivos que soportan estas tecnologías.

En cuanto a la capacidad y área de cobertura de las dos tecnologías ambas son similares. No obstante, LoRaWAN proporciona un data rate mayor que Sigfox, llegando a ser varias veces superior. En cuanto al área de cobertura que ofrecen las dos tecnologías es similar, aunque un poco superior en el caso de Sigfox. En cuanto a la seguridad LoRaWAN en su protocolo implementa medidas de seguridad para cifrar la información asegurando la integridad y la confidencialidad.

La complejidad de implementación y mantenimiento, se verá afectada por la aceptación y utilización de estas tecnologías en la actualidad, dado que cuanto más extendidas estén habrá acceso a un mayor número de fuentes y ejemplos a los que recurrir como referencia. En este caso, parece que LoRaWAN está cobrando mucha fuerza y la complejidad de implantación y mantenimiento es mucho menor.

Si se atiende a los costes se puede apreciar que ambas tecnologías tienen costes de equipos y módulos transmisores para los dispositivos parecidos. En particular, SigFox tiene un coste algo inferior debido a que hay más empresas fabricantes de los chips necesarios, aunque es una diferencia marginal.

Tabla 1 Evaluación de tecnologías de comunicación SigFox vs LoRa/LoRaWAN

Criterio	Ponderación	SigFox	LoRa/LoRaWAN
Capacidad de transmisión, área de cobertura y seguridad	40	15	25
Complejidad de implementación y mantenimientos	30	10	20
Costes dispositivos y equipos	30	20	10
Total	100	45	55

6.1.4. Selección

Tras este análisis se concluye que LoRa/LoRaWAN es una solución más adecuada para el desarrollo propuesto en este trabajo, ya que proporciona unas capacidades ligeramente superiores a la otra

alternativa, su complejidad de implementación y mantenimiento es menor que la otra y el coste de ambas es similar.

6.2. Recogida de información a través de la red

Partiendo de la selección realizada en el punto anterior se ha de analizar la forma más adecuada de recoger la información de los dispositivos LoRaWAN para la maqueta. Esto se puede llevar a cabo mediante una red LoRaWAN propia, o utilizando alguna plataforma preexistente.

En la actualidad existen dos plataformas que utilizan LoRaWAN y se pueden utilizar abiertamente. Por un lado **LORIoT [13]**, que ofrece la posibilidad de registrar un Gateway LoRaWAN y recibir datos de hasta diez dispositivos de forma gratuita.

Y por otro lado esta **TTN [7]** (**The Things Network**), la cual es una plataforma libre que te permite registrar gateways LoRaWAN y utilizar cualquier Gateway para recibir datos de un dispositivo de forma gratuita.

6.2.1. Red LoRaWAN propia

Esta alternativa consistiría en el despliegue de todos los elementos necesarios para proporcionar una red para recoger la información de los dispositivos LoRaWAN. Para ello será necesario disponer de al menos un gateway LoRaWAN para proporcionar cobertura.

Si se atiende al estándar LoRaWAN además de un gateway será necesario disponer de un servidor de aplicaciones, el cual según el estándar registra los dispositivos y recoge la información recibida de los dispositivos. Adicionalmente realiza funciones de gestión de claves de cifrado.

Según el estándar existen 3 claves de cifrado en el protocolo LoRaWAN:

- una clave de aplicación (**AppKey**) AES-128 específica para el dispositivo, que es asignada por el dueño de la aplicación. Cada vez que un dispositivo se une a la red mediante una asociación (o Join) se utiliza para generar una **NwkSKey** y **AppSKey**.
- La clave de sesión de red, **NwkSKey (Network Session Key)**, es una clave utilizada entre el servidor de la red y el dispositivo para cifrar y descifrar la información de un mensaje MAC.
- La clave de sesión de aplicación, **AppSKey (Application Session Key)**, es una clave de sesión de aplicación específica para el dispositivo final utilizada por la red y el dispositivo, para cifrar y descifrar información del mensaje.

Por defecto, los gateways comerciales llevan integrada una aplicación que realiza las funciones de servidor de aplicación, pero si se quiere dar cobertura con una red en la que los dispositivos puedan mandar los datos a cualquier gateway, este servidor deberá estar separado y los gateways funcionar como IP forwarders hacia él.

6.2.2. LORIoT

Loriot [13] es una empresa que proporciona una infraestructura LoRaWAN para que sus clientes puedan enviar y recibir mensajes a través de cualquier gateway conectado a su red. Adicionalmente ofrece una plataforma web con la que gestionar los gateways que posea el usuario registrado, las aplicaciones y los dispositivos asociados a ellas.

Por otro lado, Loriot ofrece una forma simple de configurar los gateways y gestionarlos para poder desplegar un servicio LoRaWAN de forma rápida. Esta empresa ofrece diferentes servicios y ofertas

relacionadas con IoT y la tecnología LoRaWAN. Adicionalmente dispone de un modo gratuito con el cual se puede registrar un único gateway y hasta 10 dispositivos, solo requiriendo un registro previo del usuario.

Permite recibir los datos de la aplicación a través de un servicio HTTP, con el cual se pueden recibir los datos de los servidores de Lorient. Si se dispone de una cuenta de pago, se proporcionan servicios adicionales, como la posibilidad de la descarga y almacenaje de los datos de la aplicación.

6.2.3. The Things Network

TTN (The Things Network) es una red IoT basada en LoRaWAN global, abierta y financiada mediante crowd-funding, la cual ofrece cobertura proporcionada por los gateways de los usuarios de la red. Un usuario se conecta a la red y puede registrar un Gateway; desde gateways comerciales hasta gateways hechos mediante una raspberry y un módulo de comunicaciones LoRaWAN por ejemplo. Actualmente, TTN está desarrollando su propio gateway LoRaWAN, el cual se encuentra en fase de pruebas.

TTN proporciona un back-end de servidores de aplicación que gestionan los gateways, y una plataforma que permite a los usuarios crear sus propias aplicaciones, asociar sus dispositivos y recibir los datos.

Además ofrece muchas formas de recibir la información de la red, desde mediante servicios HTTP, servicios REST, o incluso un broker MQTT al que se puede subscribir un cliente para recibir los datos.

TTN proporciona una forma sencilla de desplegar un servicio IoT sin requerir una gran infraestructura, y de forma gratuita. Solo se requiere tener cobertura de un gateway perteneciente a la red, propio o de un tercero y algún dispositivo que envíe información a la aplicación.

6.2.4. Criterios de selección

Los criterios de selección que se han tenido en cuenta para la selección de la solución para la recogida de información a través de la red han sido los siguientes:

- Gestión de aplicaciones y recogida de datos (40 %)
 - Bajo este criterio se valora la importancia que se dará a las facilidades que ofrece esta solución con respecto a la gestión de aplicaciones y a la recogida de los datos.
- Complejidad de implementación y mantenimiento (30 %)
 - Bajo este criterio se valorará la complejidad de llevar a cabo el despliegue de esta solución, y más importante, el mantenimiento necesario para su funcionamiento continuado.
- Costes (30 %)
 - Se han de tener en cuenta los costes que conllevan cada una de las tres tecnologías, tanto a la hora del despliegue inicial como a la hora de mantener el sistema, o el coste de los propios dispositivos que se usarán en la maqueta.

En cuanto a la gestión de aplicaciones y recogida de datos, la alternativa de crear una red propia proporciona una completa libertad a la hora de crear aplicaciones y añadir dispositivos, pero no ofrece ninguna facilidad para su gestión. Lorient ofrece facilidades para la gestión pero muchas de sus funciones están restringidas si no se tiene contratado el servicio de pago. TTN ofrece muchas facilidades y no pone ningún tipo de límite, permite dispositivos y aplicaciones ilimitadas, y adicionalmente ofrece múltiples medios con los que recoger los datos de las aplicaciones.

La complejidad de implementación y mantenimiento es similar en las alternativas de Lorient y TTN, que permiten implementarlas de forma sencilla. En el caso de la red propia, la implementación se vuelve más compleja y requeriría invertir mucho más tiempo.

En cuanto a los costes, todas las alternativas requerirían en principio la compra de un Gateway LoRaWAN. En el caso de la red propia podrían ser necesarios equipos adicionales para funcionar como servidor de aplicaciones LoRaWAN, o si se necesita más de un Gateway para dar cobertura. Las alternativas de Lorient y TTN permiten ambas la utilización de su infraestructura de forma gratuita, aunque en el caso de Lorient con funciones limitadas.

Tabla 2 Evaluación de alternativas para recogida de información a través de la red

Criterio	Ponderación	Red Propia	LORIENT	TTN
Gestión de aplicaciones y recogida de datos	40	5	15	20
Complejidad de implementación y mantenimientos	30	5	10	15
Costes	30	10	10	10
Total	100	20	35	45

6.2.5. Selección

Tras este análisis se concluye que TTN es una solución más adecuada para el desarrollo propuesto en este trabajo, ya que proporciona facilidades para su implementación, proporciona herramientas para gestionar las aplicaciones y dispositivos de forma sencilla, no tiene ningún tipo de límite de dispositivos y supone unos costes similares si se utiliza la versión con capacidades limitadas de Lorient.

6.3. Protocolos de comunicación

Con el auge del IoT ha sido necesario el desarrollo de protocolos de comunicación que se adecúen a las necesidades de este tipo de aplicaciones. Dichos protocolos permitirán enviar a través de Internet la información recogida por los sensores o equipos de una forma segura, simple y eficazmente.

Muchos de estos protocolos son protocolos basados en envío de mensajes, como los utilizados en aplicaciones de mensajería instantánea. Entre ellos podemos destacar los protocolos **MQTT/MQTT-SN [3]** (**Message Queuing Telemetry Transport/MQTT For Sensor Networks**) o **CoAP [5]** (**Constrained Application Protocol**), ambas soluciones proveen mecanismos asíncronos de comunicación, trabajan sobre IP y son más adecuados para entornos con recursos limitados que otras soluciones basadas en HTTP.

A continuación se analizarán los dos protocolos mencionados anteriormente, el protocolo MQTT y el protocolo CoAP y se procederá a seleccionar la alternativa más adecuada para este TFM.

6.3.1. MQTT

MQ Telemetry Transport [3] (MQTT) es un protocolo de mensajería abierto, simple y ligero, basado en publicaciones y suscripciones a mensajes en formato JSON, entre un cliente y un broker. El protocolo trabaja sobre TCP/IP, pero existen otras variantes como MQTT-SN que funcionan sobre redes no TCP/IP. Este protocolo en particular ha sido diseñado para localizaciones remotas con ancho de banda limitado.

Las entidades involucradas en la comunicación son las siguientes:

- El **cliente** puede ser desde un microcontrolador hasta un servidor, es decir, sensores, dispositivos móviles, servidores, etc. Esta entidad se encarga de enviar datos hacia el bróker (generar datos)

mediante mensajes de tipo publish, o a solicitar que el bróker le envíe datos (consumir datos) mediante mensajes de tipo subscribe.

- El **Broker** es el núcleo del protocolo, recibe todos los mensajes de los clientes, los filtra y se los reenvía a los clientes interesados en un determinado topic. Se encarga de la autenticación y autorización de clientes.

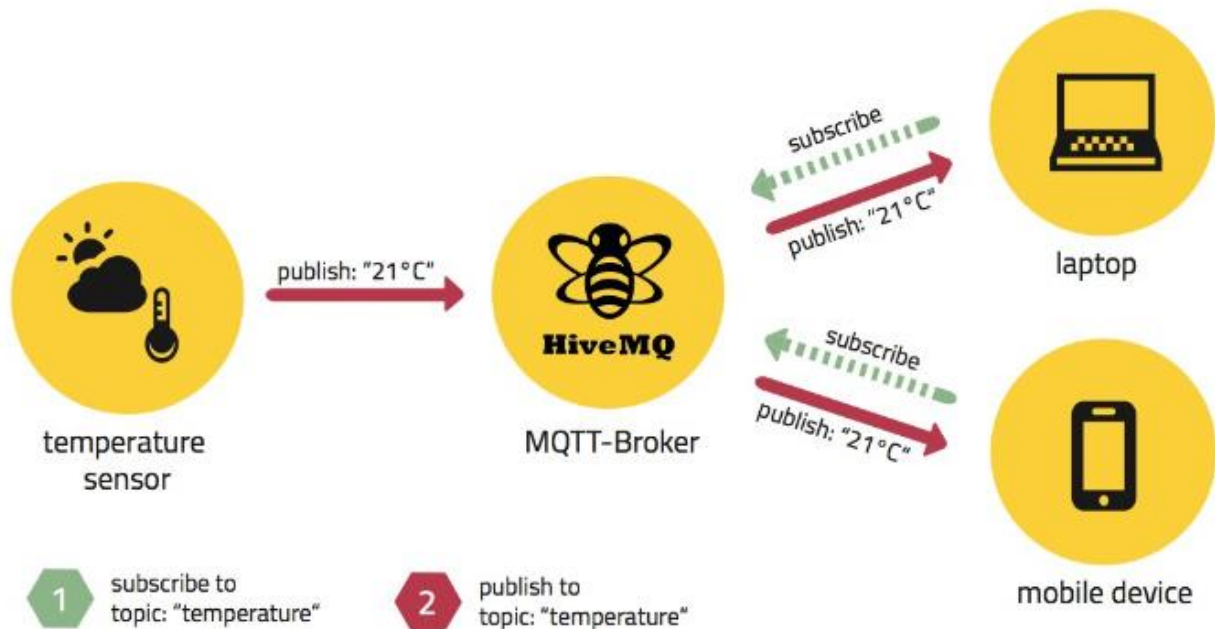


Figura 4 Esquema de funcionamiento MQTT

El bróker MQTT separa al publicador del suscriptor, los clientes se conectan siempre con él, los clientes son publicadores y/o suscriptores. Un cliente envía un mensaje (Publish) y uno o más clientes reciben el mensaje, si previamente se han suscrito a ese topic. Cuando un cliente se suscribe a un topic cada vez que alguien publica algo en ese topic el bróker reenvía esos mensajes a todos los suscriptores. Esto se puede observar en la Figura 4

MQTT se basa en los topics para decidir qué cliente recibe qué mensaje. Los topics son cadenas de texto jerárquicas separadas por /. Un ejemplo podría ser: "eventos/logs/activaciones". En este caso los mensajes de activaciones estarían dentro del topic logs y todos los logs dentro del topic eventos. Un cliente se puede suscribir a un topic en particular o una serie de topics.

6.3.2. CoAP

CoAP [5] (**C**onstrained **A**pplication **P**rotocol) es un protocolo de la capa de aplicación para la utilización en dispositivos con recursos limitados, tales como las redes de sensores. CoAP está diseñado para que pueda ser traducido fácilmente a HTTP para simplificar su integración en la web, y también cumple con requisitos como capacidad de multicast, cabeceras muy pequeñas y simplicidad.

Estas características mencionadas anteriormente son muy importantes para dispositivos IoT y para la comunicación **M2M** (**M**achine-**t**o-**M**achine), debido a que suelen ser dispositivos embebidos y tienen menor memoria y fuentes de alimentación que los dispositivos tradicionales que se conectan a Internet.

Otras Características de CoAP:

- Complejidad baja de cabeceras y parseo.
- Soporte de URI y Content-type de HTTP.
- Soporte para el descubrimiento de recursos mediante servicios CoAP.
- Subscripciones a recursos y envío de notificaciones simples.

6.3.3. Criterios de selección

Los criterios de selección que se han tenido en cuenta para la selección del protocolo de comunicación han sido los siguientes:

- Compatibilidad con las otras tecnologías utilizadas (30 %)
 - Bajo este criterio se evaluará la importancia que se dará a la compatibilidad y facilidades que ofrecen las otras tecnologías utilizadas para esta solución.
- Complejidad de implementación y mantenimiento (30 %)
 - Bajo este criterio se valorará la complejidad de llevar a cabo el despliegue de esta solución, y más importante, el mantenimiento necesario de la misma para su funcionamiento continuado.
- Seguridad (40 %)
 - Bajo este criterio se tendrá en cuenta la seguridad que ofrece cada alternativa, atendiendo a criterios de confidencialidad de la información, como al aislamiento de los dispositivos que forman parte de la comunicación y otras cuestiones referentes a la seguridad de los elementos involucrados.

En cuanto a la compatibilidad con las otras tecnologías utilizadas, la alternativa MQTT cuenta con ciertas ventajas frente a CoAP, TTN implementa una solución para recoger datos mediante la subscripción a un bróker MQTT. Mientras que con CoAP haría falta adquirir estos datos con alguna solución alternativa y convertirlos para poder enviarlos mediante CoAP.

En cuanto a la complejidad de implementación y mantenimiento, ambas alternativas cuentan con documentación extensa, aunque MQTT está más extendida, con lo que se facilita la labor de encontrar ejemplos y documentación. Adicionalmente existen librerías para estos protocolos para diferentes equipos y lenguajes de programación.

En cuanto a la seguridad, MQTT ofrece la capacidad de cifrar los datos mediante TLS y permite aislar a los dispositivos que producen los datos de los receptores de estos, mediante la utilización de un bróker que se encarga de reenviar los datos. Esto reduce las probabilidades de que se pueda atacar a la red de sensores si se tuviera acceso a un consumidor de los datos. CoAP ofrece funcionalidades similares pero como trabaja sobre UDP y es más vulnerable a determinados ataques informáticos.

Tabla 3 Evaluación alternativas protocolo de comunicación

Criterio	Ponderación	MQTT	CoAP
Compatibilidad con las otras tecnologías utilizadas	30	25	5
Complejidad de implementación y mantenimientos	30	15	15
Seguridad	40	25	15
Total	100	60	40

6.3.4. Selección

Tras este análisis se concluye que MQTT es una solución más adecuada para el desarrollo propuesto en este trabajo, ya que proporciona una mayor facilidad para integrarla con otras de las alternativas tecnológicas utilizadas, ofrece una complejidad de implementación y mantenimiento adecuada, y es una alternativa muy buena para cumplir con los requisitos de seguridad en la comunicación.

6.4. Almacenamiento y representación de datos

A raíz de la expansión del número de dispositivos que se conectan a las redes en escenarios IoT, el almacenamiento y análisis posterior de estos datos cobra mucha importancia. Es por ello, que un aspecto importante del análisis de alternativas de este TFM, trata de cómo se almacenan y analizan estos datos.

El almacenamiento de los datos en los sistemas informáticos tradicionales se ha llevado a cabo y todavía se realiza mediante bases de datos relacionales. En este tipo de bases de datos se establecen tablas con campos para la información esperada y se relacionan estas tablas entre sí, siguiendo la relación existente entre los datos. La mayoría de este tipo de bases de datos están basadas en SQL. Esta será una de las alternativas que se analizarán en este análisis aportado. Junto con este almacenamiento será necesario un programa o una página web que permita analizar y visualizar los datos.

Por otro lado, con el auge del IoT y el denominado BigData han comenzado a surgir nuevos tipos de bases de datos que se han denominado **NoSQL (Not only SQL)**, y adicionalmente existen varios subtipos; en base a grafos, documentales, etc. Este tipo de bases de datos son capaces de absorber datos sin tener un formato o diseño específico, por ejemplo, las bases de datos documentales almacenan documentos, un documento podría tener unos campos en particular y otro del mismo índice podría tener parte o ninguno de los campos que tiene el anterior, cada documento sería como una entrada diferente en una tabla que no tiene que tener el mismo formato que las anteriores. En particular en este análisis se estudiará un sistema de almacenamiento basado en Elasticsearch que permite almacenar datos como ficheros, y una búsqueda rápida de estos. Adicionalmente, cuenta con dos complementos Logstash y Kibana que permiten adquirir datos de múltiples fuentes y visualizar los datos, respectivamente.

A continuación se analizan estas dos alternativas, una base de datos relacional más un front-end web para analizar y visualizar los datos, y lo que comúnmente se conoce como la pila **ELK[4]** (Elasticsearch, Logstash y Kibana).

6.4.1. Base de datos relacional

Una base de datos relacional es una base de datos que organiza los datos en base al modelo relacional. Este modelo organiza los datos en una o más tablas de columnas y filas, con un identificador único en cada fila. Cada fila representa una entidad o relación (como por ejemplo, un producto o cliente). Las filas de diferentes tablas se pueden unir añadiendo a una de las tablas una columna con el identificador único de esa fila.

Existen diferentes softwares de bases de datos relacionales, tanto gratuitas como de pago, por ejemplo, MySQL; un software de gestión de bases de datos de código abierto, que ofrece un servidor y un programa para gestionar tu base de datos basada en SQL. Otro ejemplo sería Oracle database que es otro software de gestión de bases de datos que ofrece muchas herramientas y permite simplificar la gestión de bases de datos. En este caso es un software de pago.

Para visualizar y analizar los datos de este tipo de bases de datos, sería necesario utilizar algún software para buscar la información deseada en la base de datos y analizarla. Para ello hace falta un servidor que

recoja esta información y la muestre mediante una página web. Para ello, se podrían programar los servicios REST necesarios mediante tecnología Java EE, y generar las páginas web para la representación de los datos.

6.4.2. Elasticsearch, Logstash y Kibana (ELK)

Elasticsearch [4] es un servidor de búsqueda basado en Lucene (Servidor de búsquedas de alto rendimiento de Apache), que provee un motor de búsqueda de texto completo, desarrollado sobre Java y de código abierto. Elasticsearch permite buscar todo tipo de documentos, de forma escalable, con búsqueda casi en tiempo real. Es capaz de escalar horizontalmente, lo que permite aumentar su capacidad de forma simple mediante la utilización de un cluster de nodos Elasticsearch.

Logstash [4] es un software que funciona junto a Elasticsearch, y permite adquirir datos de múltiples fuentes con diferentes formatos, simultáneamente, transformándolos, e inyectarlos a Elasticsearch. El software recoge esos datos los parsea, identificando los nombres de los campos, los estructura y los transforma para convertirlos a un formato común para su posterior análisis. Esto permite que el análisis se realice de forma más rápida y simple.

Por último, **Kibana [4]** ofrece una interfaz con la que se puede realizar un análisis y representación de los datos almacenados en Elasticsearch. Kibana permite buscar información rápidamente y filtrar los datos en base a los diferentes campos existentes, y proporciona una forma sencilla de generar gráficas y un panel de control con el que se puede monitorizar el estado de la información recogida.

6.4.3. Criterio de selección

Los criterios que se han tenido en cuenta para la selección del sistema de almacenamiento y representación de los datos han sido los siguientes:

- Complejidad de implementación y mantenimiento (40 %)
 - Bajo este criterio se valorará la complejidad de llevar a cabo el despliegue de esta solución, así como el mantenimiento de la misma.
- Herramientas, funcionalidades y seguridad (30 %)
 - Bajo este criterio se evaluarán las herramientas y funcionalidades ofrecidas por cada alternativa.
- Escalabilidad (30 %)
 - Bajo este criterio se tendrá en cuenta la capacidad de escalado que las alternativas ofrecen para su posible ampliación en el futuro.

En cuanto a la complejidad de la implementación y mantenimiento, la alternativa de ELK tiene una complejidad de implementación baja, existen muchas guías y tutoriales que facilitan esta tarea y la configuración es relativamente sencilla. La alternativa de una base de datos relacional posee una complejidad de implementación y mantenimiento más elevada, el despliegue del servidor y la base de datos es sencillo pero requiere un diseño adecuado de la base de datos. En caso de cambiar el tipo de datos que se van a recibir será necesario rediseñar la base de datos para almacenarlos.

Las herramientas y funcionalidades que ofrece ELK son muy útiles para el análisis de los datos y su representación, mientras que con la otra alternativa, haría falta programar y desplegar un servicio web. Adicionalmente también proporciona herramientas de seguridad que permitirán realizar un control de acceso a la base de datos.

La escalabilidad de una base de datos del modelo relacional, es muy compleja y requiere aumentar la capacidad del equipo que la contiene, si es centralizada, o añadir un nuevo equipo si es una base de datos distribuida lo que conlleva una mayor complejidad en la configuración. La alternativa de ELK proporciona

escalabilidad de forma más simple, con añadir un nuevo nodo de Elasticsearch al cluster se lograría aumentar la capacidad del sistema.

Tabla 4 Evaluación de alternativas de almacenaje y representación de datos

Criterio	Ponderación	Base de datos relacional	ELK
Complejidad de implementación y mantenimientos	40	15	25
Herramientas, funcionalidades y seguridad	30	10	20
Escalabilidad	30	10	20
Total	100	35	65

6.4.4. Selección

Tras este análisis se concluye que ELK es una solución más adecuada para el desarrollo propuesto en este trabajo, ya que proporciona facilidades para su implementación, proporciona muchas funcionalidades y herramientas, y adicionalmente el escalado de la solución se puede realizar de una forma simple.

7. Análisis de Riesgos

En este apartado se recogerán algunos de los riesgos valorados en la definición de este TFM los cuales podrían afectar a su transcurso o al resultado del mismo. Estos riesgos se pueden clasificar en riesgos externos, técnicos o de planificación.

7.1. Identificación de Riesgos

7.1.1. Riesgos externos para el desarrollo de la maqueta

En este TFM se llevará a cabo una demostración de las funcionalidades de la arquitectura diseñada mediante el despliegue de una maqueta que utilizará equipos comerciales e industriales. Estos equipos fabricados por terceros serán vitales para el desarrollo de la maqueta y cualquier inconveniente en su transporte o funcionamiento afectaría negativamente al TFM.

Durante el despliegue de la maqueta se pueden producir retrasos e inconvenientes que afecten a la planificación planteada para el desarrollo de la maqueta derivados de la existencia de algún defecto o problema con algunos de los equipos específicos utilizados en la maqueta.

Adicionalmente, para el desarrollo de la maqueta se utilizará una red LoRa abierta, TTN [7], esta red es ajena al proyecto y no se tienen ningún tipo de control sobre ella. Por este motivo en caso de ocurrir cualquier problema con su funcionamiento este repercutirá negativamente en la recopilación de los datos enviados por los dispositivos LoRa.

7.1.2. Riesgos de la implementación

Elasticsearch y Kibana son un software que se encuentra en continuo desarrollo y por ello algunos de los plugins y funcionalidades utilizadas en este TFM podrían no funcionar con versiones diferentes de aquellas en las que han sido desarrollados.

Adicionalmente para el desarrollo de la maqueta se utiliza la red TTN y se recogerán los datos de esta red de un bróker MQTT mediante la utilización de un software propio que los reenviará a la arquitectura diseñada. Cualquier modificación en la forma en que se han de solicitar los datos al bróker MQTT afectará negativamente al desarrollo y será necesario modificar el software.

Para el despliegue del sistema de recogida de datos y el despliegue de la maqueta es necesario el diseño y desarrollo de varios componentes y como en cualquier desarrollo software existen ciertos riesgos inherentes presentes en este tipo de proyectos, derivados de la utilización de librerías de terceros que podrían afectar al desarrollo.

7.1.3. Riesgos en Planificación

Como todo proyecto, durante su ejecución éste es susceptible de cambios en sus especificaciones iniciales aunque este riesgo es muy poco probable teniendo en cuenta la duración del proyecto. Estos cambios pueden estar motivados por razones externas, como podría ser la variación de las funcionalidades requeridas de manera general por sistemas de terceros utilizados en el desarrollo. Asimismo, estos cambios podrían ser resultado de la comprobación de la imposibilidad del desarrollo de determinadas sensores pensados para la maqueta, debido a la inexistencia de medios, la falta de conocimientos o por la complejidad de su implementación.

Estos riesgos podrían influir significativamente en la duración del proyecto, pudiendo derivar en la posible continuidad del mismo a través de un nuevo proyecto necesario para mejorar la aplicación resultante.

7.2. Análisis de Riesgos

Una vez detectados y establecidos los posibles riesgos que podrían afectar al TFM, se procederá a analizar su probabilidad e impacto, presentados en la Tabla 5.

Tabla 5 Análisis cualitativo-cuantitativo de los riesgos del proyecto

ID	Riesgo	Probabilidad	Impacto	Causa
R1	Problemas con los equipos seleccionados para el despliegue de la maqueta	Posible (0,5)	Alto (0,4)	Defectos de fabricación o problemas con el envío de los equipos
R2	Fallo en el funcionamiento de la red TTN debido a problemas técnicos ajenos al proyecto	Difícil (0,3)	Alto (0,4)	Fallo de la red TTN por motivos desconocidos
R3	Incapacidad para recoger los datos de TTN	Difícil (0,3)	Alto (0,4)	Variación en la forma o requisitos para la obtención de los datos del bróker MQTT de TTN
R4	Conocimientos insuficientes o imposibilidad para desarrollar uno de los sistemas medidos en la maqueta	Posible (0,5)	Moderado (0,2)	Tiempo insuficiente para el desarrollo, o problemas no previstos en un principio que imposibilite el despliegue

7.3. Planificación de la respuesta

En el caso del riesgo R1 cuando se trate de problemas derivados del transporte de los equipos o un defecto de fabricación de los mismos, lo que sería un riesgo externo no controlable, se propone la aceptación del riesgo y se proponen las siguientes respuestas:

- Posponer aquellas tareas que dependan de estos equipos cuando estas tareas puedan verse afectadas.
- Ponerse en contacto con los responsables, fabricantes o empresa de transporte, para tratar de encontrar una solución efectiva al problema.

En cuanto al riesgo R2 un fallo de la red LoRaWAN TTN, se propone aceptar el riesgo dado que es un riesgo externo con una probabilidad de que ocurra baja y en caso de producirse, la probabilidad de que el fallo persista durante mucho tiempo antes de ser corregido es muy baja, por lo que se procederá a realizar otras tareas hasta que el fallo se haya corregido.

Cuando se dé el riesgo R3, una incapacidad de recoger los datos de TTN debidos a una variación de la forma de recogerlos o por un fallo en el programa en cargado, se propone revisar la documentación de TTN para estudiar las variaciones que se hayan podido producir que provocarán el fallo y proceder a adaptar el código a la nueva situación.

Finalmente para el riesgo R4 se propone una reducción del riesgo aumentando las horas invertidas al estudio de las tecnologías y la revisión de las funcionalidades planificadas para la maqueta para asegurar su viabilidad a fin de evitar retrasos innecesarios en otras tareas debido a estas funcionalidades no esenciales para el cumplimiento de los objetivos.

8. Metodología

En este punto del TFM se describen los aspectos metodológicos y de desarrollo del trabajo realizado. Se expondrá la hoja de ruta seguida para alcanzar los objetivos de este TFM, junto con una justificación del motivo por el cual ha sido elegida.

Para el desarrollo de este TFM se ha seguido una metodología tradicional de desarrollo, donde inicialmente se analiza el problema y las diferentes soluciones alternativas posibles, tras lo cual se selecciona la mejor alternativa.

Inicialmente se presentará una descripción breve de las tareas y el equipo de trabajo involucrado en el TFM y para terminar, un diagrama Gantt para una mejor visualización de esta planificación.

8.1.Descripción de las Tareas

En este apartado se presenta un resumen de las tareas que se han desempeñado a lo largo del TFM, así como las personas que colaboran en la realización del mismo.

8.1.1. Equipo de Trabajo

En la Tabla 6 se muestra el equipo de trabajo que ha realizado este proyecto.

Tabla 6 Equipo de Trabajo

Identificador	Nombre	Empresa	Responsabilidad
L1	María Victoria Higuero Aperribai	UPV/EHU	Directora del Trabajo
L2	Borja Marcos	Sarenet	Ingeniero Sénior
L3	Mikel De Prado	UPV/EHU	Ingeniero Junio

- L1: Directora del TFM perteneciente al grupo de investigación I2T cuya labor es de dirigir y realizar un seguimiento del proyecto.
- L2: Ingeniero Sénior de la empresa Sarenet cuya tarea consiste en supervisar y aportar su experiencia a la realización del TFM.
- L3: Ingeniero Junior cuya labor es elaborar el proyecto siguiendo las pautas definidas por la directora del proyecto.

8.1.2. Paquetes de trabajo y Tareas

En este apartado se presenta una descripción de los paquetes de trabajo y las tareas que son necesarias para la realización de este TFM. A continuación en la Tabla 7 **Lista de acrónimos** se muestra una lista de acrónimos para facilitar la interpretación de este apartado.

Tabla 7 Lista de acrónimos

Acrónimo	Definición
PTX	Paquete de trabajo número X
TXY	Tarea del paquete de trabajo X número Y
HX	Hito número X

A continuación se identifican y describen los paquetes y tareas del TFM:

PT0 – Gestión y Documentación del Proyecto: Este paquete se extiende a lo largo de toda la vida del proyecto. Hay varias tareas implicadas: Gestión de recursos, seguimiento del desarrollo del TFM, Acciones correctoras, control del proyecto, revisión de entregables y otras tareas que tienen como fin el alcance de los objetivos del proyecto. También están incluidas en este paquete todas las reuniones, inicio y cierre del proyecto y la documentación del mismo.

PT1 – Definición de objetivos y especificaciones: En este paquete de trabajo se definen los objetivos y especificaciones del TFM. Para ello se define el objetivo principal que es el diseño y despliegue de una arquitectura para la recogida, almacenamiento y presentación de medidas de sensores IoT, así como sub-objetivos que llevarán al cumplimiento del objetivo principal y del proyecto y la maqueta realizada.

PT2 – Diseño de la arquitectura: En este paquete de trabajo se identificarán y analizarán las alternativas consideradas para la realización del TFM. Y se propondrá un diseño de la arquitectura para la recogida y presentación de medidas de sensores IoT.

- **T21:** Identificación de alternativas.
- **T22:** Análisis y estudio de alternativas.
- **T23:** Diseño de la arquitectura.

PT3 – Diseño de la maqueta: En este paquete de trabajo se diseñará una maqueta que tenga como fin demostrar el funcionamiento de la arquitectura. También estarán presentes en este paquete todas las tareas relacionadas con la selección y adquisición de equipos para la maqueta.

PT4 – Despliegue y Realización de la maqueta: En este paquete de trabajo se encuentran las tareas relacionadas con el despliegue y montaje de la maqueta diseñada. Este paquete consta de las siguientes tareas:

- **T41:** Despliegue de arquitectura de recogida, almacenamiento y presentación de datos.
- **T42:** Despliegue y configuración de recogida de datos LoRaWAN.
- **T43:** Despliegue de recogida de medidas de contadores eléctricos mediante Modbus y ModbusTCP.
- **T44:** Desarrollo de paneles de control para la presentación de los datos.

PT5 – Análisis de funcionamiento de la maqueta y conclusiones: En este paquete de trabajo se comprobará el funcionamiento de la maqueta, los datos obtenidos de ella y se obtendrán unas conclusiones globales del desarrollo del TFM.

A continuación en la Tabla 8 se muestran los hitos de este TFM.

Tabla 8 Hitos

Identificador	Hito
H1	Inicio de proyecto
H2	Entrega de diseño de la arquitectura
H3	Entrega de diseño de maqueta
H4	Finalización de despliegue de maqueta
H5	Fin de proyecto

8.2. Diagrama de Gantt

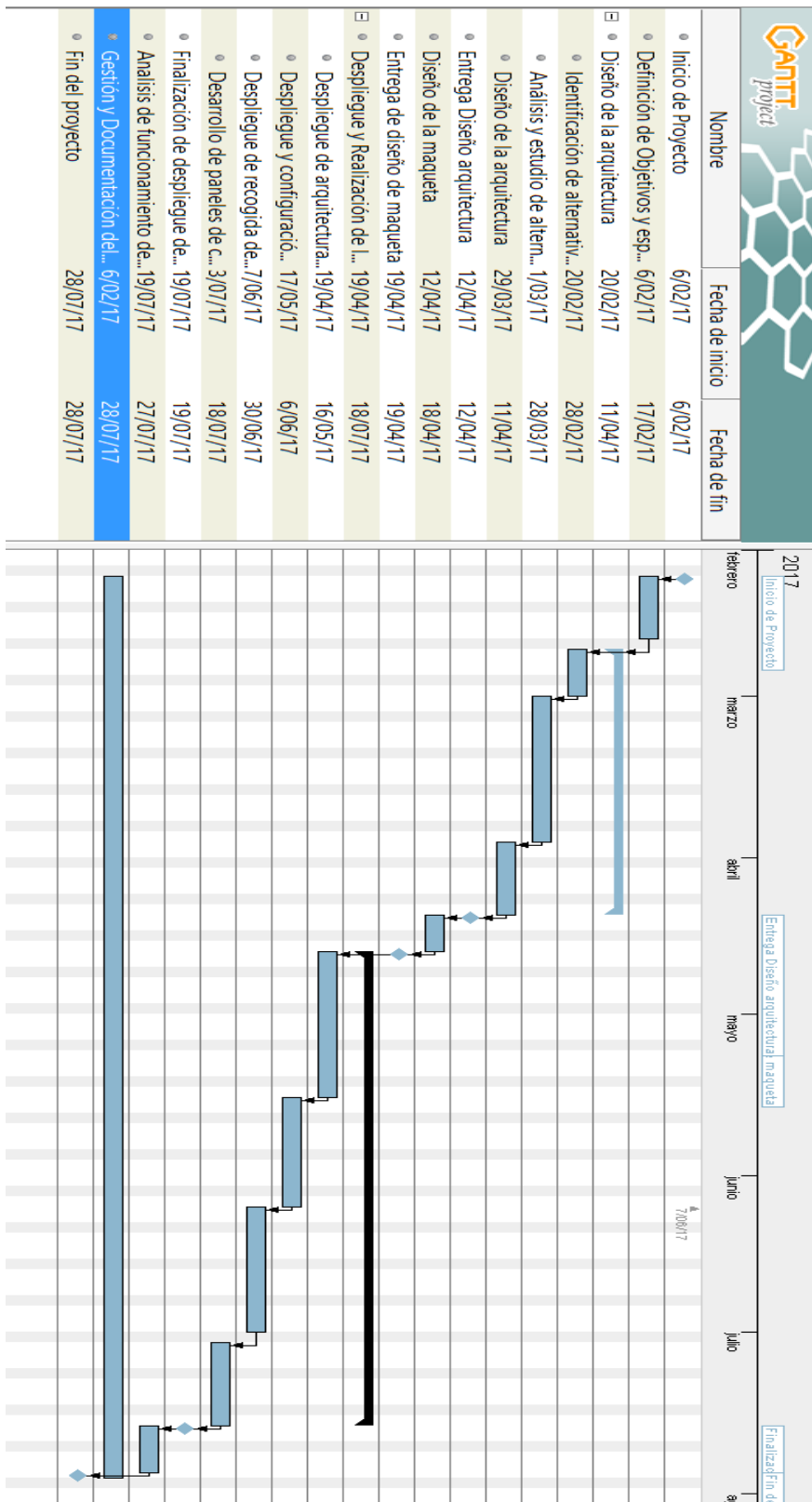


Figura 5 Diagrama Gantt

9. Descripción de la solución

En este TFM se diseñará y desarrollará una arquitectura de IoT que recogerá información de múltiples fuentes y la almacenará en una base de datos para su posterior análisis y representación. Para la realización de este proyecto será necesaria tanto la implementación de un sistema de almacenamiento y representación de los datos, y el envío a través de Internet de estos, como la configuración de los sensores que recogerán la información.

Para comprobar el funcionamiento de la arquitectura se desplegará una maqueta, compuesta por un backend que almacenará y visualizará los datos, así como una serie de sensores. Los sensores y dispositivos utilizados enviarán o les serán solicitados los datos. En la maqueta existirán dos tipos de dispositivos: por un lado dispositivos que utilizarán la tecnología LoRaWAN para enviar la información y por otro lado, unos dispositivos industriales que se comunican a través de un bus estándar RS-485 y mediante la utilización del protocolo Modbus, dada su implantación en industria y por requisitos del equipo del que se recogen los datos, un contador eléctrico Circutor CVM-C10. Dado que para obtener datos con los que realizar un análisis adecuado de las medidas del contador, las medidas deberían tener una frecuencia de muestreo de 1 minuto, sería imposible la utilización de LoRa dado que se sobrepasa el 1% de ciclo de trabajo definido para la banda ISM [12] de 868 MHz, por lo que se utilizará una conexión Ethernet.

Para describir la solución en profundidad se han tenido en cuenta los siguientes puntos:

- **Diseño de la arquitectura:** Se explicará el diseño de la arquitectura para la recogida, almacenamiento y representación de los datos.
- **Diseño y Desarrollo de la maqueta**
 - **Sensores y Dispositivos de la maqueta:** Se describirán los sensores y dispositivos utilizados en la maqueta y su configuración.
 - **Conexión a TTN y configuración requerida para la comunicación:** Se explicará la información relativa para la conexión y configuración de la red TTN.
 - **Recogida de datos y almacenamiento en Elasticsearch:** En este punto se tendrá en cuenta todo lo necesario para la recogida de los datos, la recogida de datos de TTN y la solicitud de los datos al medidor de eléctrico, y su posterior envío e inserción en la base de datos.
 - **Representación de datos mediante Kibana:** Se explicarán los datos que se analizarán en la maqueta y cómo se representarán gráficamente, además de la configuración de cualquier plugin necesario para su representación.
- **Seguridad:** En este punto se expondrán las medidas de seguridad proporcionadas por el diseño y las herramientas utilizadas para ello, presentes en la maqueta.

9.1. Diseño de la arquitectura

A continuación se describe el diseño de la arquitectura que se ha realizado, para la recogida, almacenamiento y representación de los datos.

Esta arquitectura recogerá datos de cualquier fuente mediante la solicitud de datos a un bróker de mensajes, a través de un cliente que inyectará los datos al módulo de almacenamiento de la arquitectura mediante un módulo de recogida e inserción de datos.

Estos datos serán visualizados a través de Internet mediante un módulo de presentación, el cual solicitará los datos al módulo de almacenamiento y realizará representaciones visuales de los mismos.

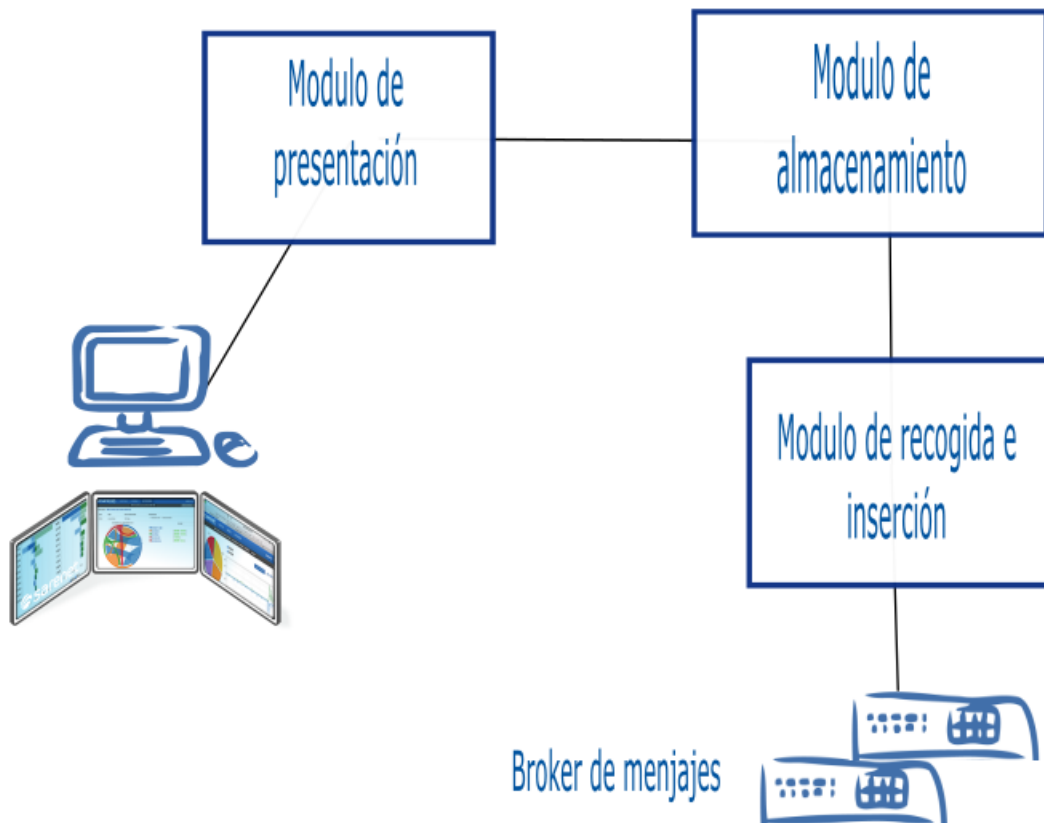


Figura 6 Diagrama arquitectura general

Como se observa en la Figura 6 la arquitectura está compuesta por:

- Un módulo de almacenamiento que procesará búsquedas de datos solicitadas por el módulo de presentación, o cualquier búsqueda solicitada por otra fuente, y almacenará los datos recogidos.
- Un módulo de presentación de datos, con el cual se podrán generar visualizaciones graficas de los datos, y se podrán diseñar paneles de control con las visualizaciones generadas.
- Un equipo que ejecute un módulo de recogida e inserción de datos, el cual contiene un programa cliente, que recibe los datos publicados en el bróker de mensajes. Este módulo se encarga de almacenar los datos en el módulo de almacenamiento.
- Un bróker mensajes redundado implementado mediante un software que recibirá los datos publicados por sus clientes y los reenviará a los clientes que soliciten estos datos.

El módulo de almacenamiento de datos de la arquitectura deberá tener capacidades para hacer búsquedas de los datos de una forma rápida entre una gran cantidad de datos almacenados. Deberá ofrecer una arquitectura distribuida en lugar de una centralizada ya que la forma de escalar de esta última es mediante un equipo más potente mientras que en las arquitecturas distribuidas el escalado se realizaría añadiendo un nuevo equipo. El tipo de base de datos serán NoSQL las cuales son mucho más flexibles ya que se pueden modificar la base de datos sin parar su funcionamiento y ofrecen una mayor optimización para realizar búsquedas de grandes cantidades de datos.

El módulo de presentación deberá ser capaz de solicitar los datos almacenados en el módulo de almacenamiento utilizado, por lo que deberá ser compatible u ofrecer soporte para obtener datos del utilizado. A partir de estos datos, el modulo deberá permitir diseñar y presentar los datos mediante visualizaciones gráficas y el diseño de paneles de control en los cuales se podrá analizar la situación actual

de los datos. Adicionalmente, la implementación de un sistema de monitorización y alerta en el módulo de presentación permitirá realizar la monitorización de los datos de una forma más simple para los usuarios.

El módulo de recogida e inserción deberá de ser compatible o tener soporte para el módulo de almacenamiento utilizado en la arquitectura para poder insertar los datos en él. Adicionalmente se deberá permitir parsear los datos para evitar problemas en el módulo de almacenamiento. Este módulo deberá implementar un programa cliente que permita recoger los datos recibidos por el bróker de mensajes.

El bróker de mensajes deberá utilizar un software para el envío y recepción de mensajes ligero el cual pueda ser implementado en dispositivos embebidos y que no consuma excesivos recursos. Este bróker deberá de estar redundado en esta arquitectura para asegurar la disponibilidad del servicio.

A continuación se presenta un diseño de la arquitectura en base a las tecnologías seleccionadas en el análisis de alternativas para la realización de la maqueta, y así permite obtener una imagen más definida de cómo debería ser una solución basada en esta arquitectura para la recogida, almacenamiento y presentación de los datos de sensores IoT.

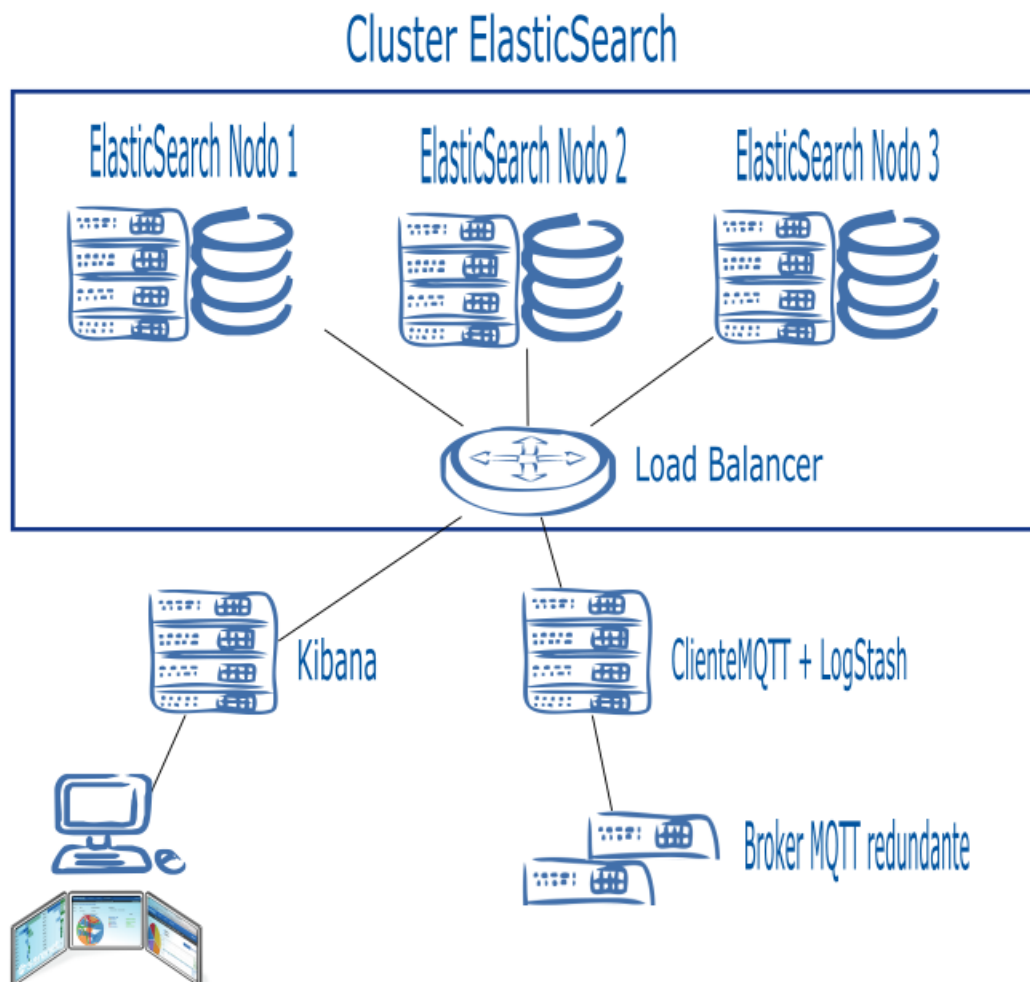


Figura 7 Diagrama de la arquitectura en base a las tecnologías seleccionadas en el análisis de alternativas

Como se ha comentado en el apartado de análisis de alternativas, el módulo de almacenamiento y el módulo de presentación de los datos se harán mediante Elasticsearch y Kibana. El motivo de peso para seleccionar esta alternativa es el escalado que ofrece esta solución. El clúster Elasticsearch podría sustituirse por un único nodo Elasticsearch o como en el diseño utilizar un clúster formado por varios nodos, el escalado consistiría en añadir nuevos nodos al clúster.

El funcionamiento de un clúster de Elasticsearch consiste en que uno de sus nodos funcionará como maestro y los demás como esclavos. En una base de datos Elasticsearch, una base de datos NoSQL, los datos se almacenan como documentos en un índice que los agrupa. Adicionalmente, estos índices se subdividen en fragmentos denominados Shards los cuales se distribuyen entre los nodos. Estos Shards tienen copias en otros nodos del clúster y uno de ellos se denomina primario. Cuando se realiza una búsqueda, se busca en todos los Shards relevantes para la búsqueda y las réplicas de estos, acelerando la búsqueda. En la Figura 8 se puede observar esta estructura.

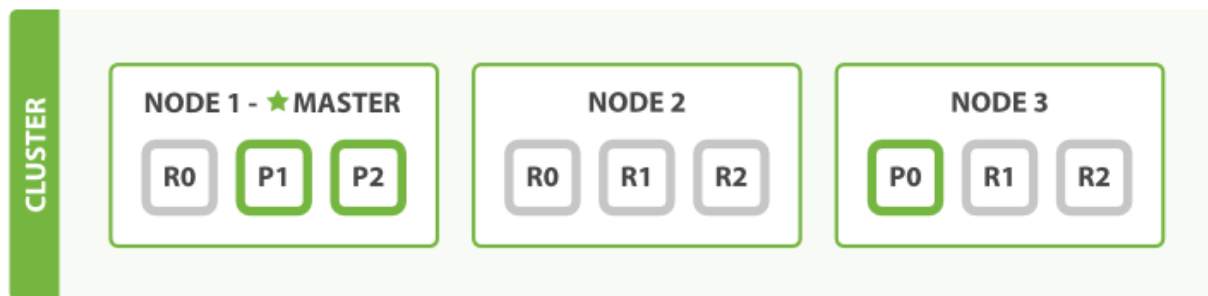


Figura 8 Clúster Elasticsearch y Shards

El módulo de presentación basado en Kibana permite diseñar visualizaciones y paneles de control a partir de búsquedas realizadas en el clúster Elasticsearch y realizar agregaciones con los datos, calcular medias, máximos, mínimos, etc. Adicionalmente, permite añadirle visualizaciones diferentes mediante la instalación de plugins para mejorar las opciones para la representación de los datos. Estas visualizaciones se pueden ver mediante una interfaz web del propio Kibana o mediante su integración en una página web con HTML.

El módulo de recogida e inserción utilizando el software Logstash formatea los datos, indica los índices en los que se almacenan e inserta los datos en el clúster Elasticsearch. Además, en este equipo se ejecuta un script de Python que se suscribe a un bróker MQTT y recibe los datos, pasándoselos al servicio Logstash.

El bróker de mensajes en base a MQTT se podrá implementar mediante la ejecución de un software como Mosquitto, que realiza funciones de bróker MQTT y recoge los datos. Este equipo se redunda y mediante un balanceo de carga esta acción es transparente para todos los equipos que envíen datos al bróker.

9.2. Diseño y Desarrollo de la maqueta

El diseño de arquitectura anterior se utilizará en una forma simplificada para la realización de una maqueta en la cual se demostrará la viabilidad, y se analizará el funcionamiento de la arquitectura, mediante la recogida de datos de varios sensores y dispositivos.

La arquitectura simplificada consistirá en implementar un único nodo Elasticsearch, junto al software Kibana y Logstash en una única máquina. Esta máquina integrará los módulos de almacenamiento, presentación y recogida e inserción de datos. Este equipo también ejecutará el programa cliente de MQTT que recibe los datos del bróker MQTT y que en el diseño se ejecuta en el equipo Logstash.

Para analizar el funcionamiento de esta arquitectura se recogerán medidas de varios sensores y dispositivos. En particular se utilizarán sensores LoRaWAN que midan la temperatura y humedad, y adicionalmente según el modelo medirán CO2 o CO. También se utilizarán balizas GPS que enviarán su posición a través de LoRaWAN. Estos sensores LoRaWAN enviarán la información a través de la red TTN.

Por otro lado, se recibirán datos de contadores eléctricos a través del protocolo Modbus mediante tecnología Ethernet, para ilustrar mejor las múltiples formas de adquirir los datos de dispositivos Modbus. Se utilizará un equipo programable que realice las lecturas mediante Modbus y un programa que realice las medidas mediante un Gateway ModbusTCP-Modbus a través de la red con ModbusTCP.

Por último, se enviarán los datos obtenidos de los dispositivos a un bróker MQTT mediante los propios programas de lectura en el caso de los dispositivos Modbus, y mediante un script para recoger los datos de TTN y reenviarlos a la arquitectura definida.

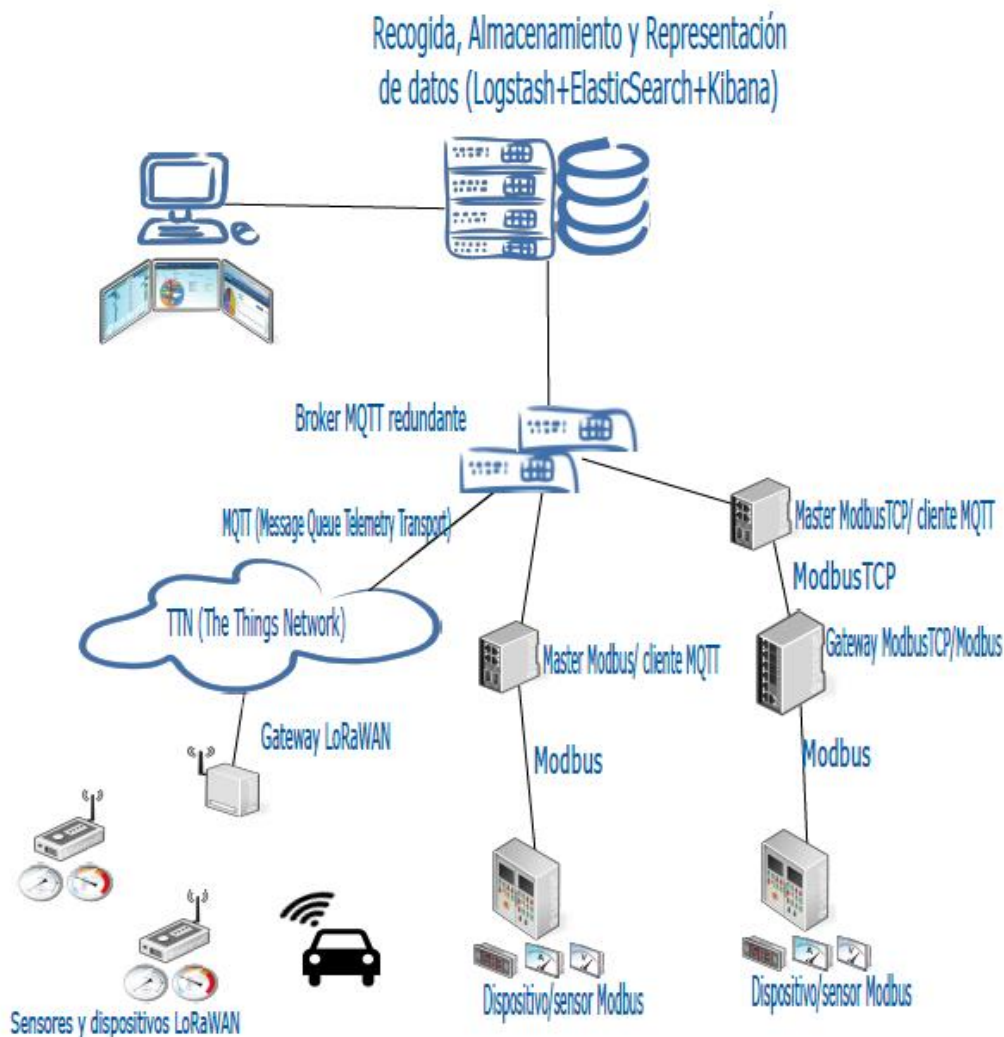


Figura 9 Diagrama de la maqueta

En la Figura 9 se puede observar un diagrama de la maqueta utilizada en este TFM. Esta maqueta se puede dividir en los siguientes módulos:

- La arquitectura de recogida, almacenamiento y representación de los datos.
- La recogida de datos mediante dispositivos LoRa a través de TTN.

- La lectura de datos del contador eléctrico mediante Modbus.
- La lectura de datos del contador eléctrico mediante ModbusTCP.
- Representación de los datos medidos con Kibana.

9.2.1. Arquitectura de recogida, almacenamiento y representación de datos

Para el desarrollo de esta maqueta se aplicará un diseño simplificado de la arquitectura presentada en el punto 9.1 Diseño de la arquitectura. En este diseño simplificado una única maquina alojará los servicios Elasticsearch, Logstash y Kibana, además de un programa cliente de MQTT, que recibirá los datos. Con un bróker MQTT en otra máquina.

Tal como se ha explicado en el punto anterior, la maqueta recogerá datos de medidas de calidad del aire, balizas GPS y contadores eléctricos. Cada una de estos tres tipos de medidas se almacenará en un índice de Elasticsearch diferente, con los siguientes identificadores:

- airq: Para medidas de calidad del aire.
- gps: Para medidas realizadas por balizas GPS.
- emeter: Para los datos obtenidos de contadores eléctricos.

Estos índices seguirán el siguiente formato [Tipo]-[Fecha], por ejemplo, airq-16-06-2017. Cada día se genera un índice para los datos almacenados. Para evitar errores con la identificación del tipo de dato de los campos por parte de Kibana, será necesario definir una plantilla donde se haga un mapeo de ellos.

El programa cliente de MQTT que se ejecuta en la maquina Logstash recibirá los datos del bróker MQTT, subscribiéndose a los siguientes topics MQTT, AirQ/+/+, GPS/+ y Emeter/+/+. Los topics siguen la siguiente estructura [tipo]/[Ubicación]/[Identificador de dispositivo], excepto el tipo GPS que no requiere ubicación. Un ejemplo seria AirQ/Edificio103/C0202. En función del tipo en el topic se enviará a un puerto diferente del servicio Logstash.

En la configuración de Logstash se define que en función del puerto por el que entre se le asignará un tipo que se utilizará para indicar el índice en el que se introducirán los datos en Elasticsearch, puertos 9401, 9402 y 9403 se utilizarán para calidad de aire(airq), balizas GPS(gps) y contadores eléctricos(emeter), respectivamente.

Estos puertos se hayan definidos en la configuración de Logstash y el programa cliente MQTT que recibe los datos del bróker MQTT.

Como se ha mencionado al comienzo de este punto, también se instalará Kibana en la maquina utilizando una configuración por defecto. En un punto más adelante se explicarán los plugins adicionales que se han utilizado para representar los datos.

9.2.2. Recogida de datos mediante dispositivos LoRa a través de TTN

Para lograr comunicar los dispositivos LoRaWAN con los equipos de almacenamiento hará falta una red a través de la cual transmitirlos. Para simplificar este aspecto se utilizará una red preexistente de uso libre, TTN (The Things Network) y unos sensores LoRaWAN configurados para transmitir con esta red.

En la Figura 11 se puede observar el diagrama de la infraestructura de TTN. Los gateways se comunican con un router regional, el cual envía los datos a un bróker, un servidor de red gestiona los gateways y el resto de parámetros de la red. El bróker se comunica con un handler, que envía los mensajes de los dispositivos a la aplicación que descifra y decodifica los datos recibidos. A continuación se puede acceder a ellos desde diferentes sistemas.

Para configurar el gateway para estas funciones, se deberá registrar el equipo en la plataforma de gestión que provee TTN a sus usuarios. Adicionalmente, será necesario instalar un servicio en el gateway que se encargará de reenviar los mensajes. Para ello se ejecutará un script de instalación que creará el servicio y permitirá configurarlo, además de deshabilitar la aplicación LoRaWAN que viene por defecto en el firmware. Tras esto, el gateway podrá conectarse a TTN y los mensajes de cualquier dispositivo LoRaWAN registrado en TTN que transmita a su alcance serán reenviados a través de TTN y se enviará a la aplicación a la que estén asociados los dispositivos. Esto se puede ver ilustrado en la Figura 12.

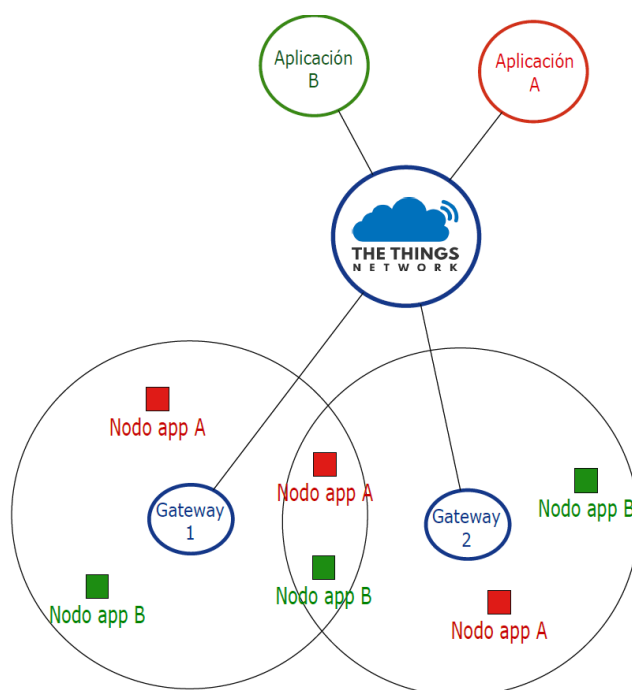


Figura 12 Esquema reenvió TTN

Adicionalmente, para enviar y recibir datos a través de los dispositivos, éstos deberán ser registrados. Para ello, se deberá crear una aplicación LoRaWAN, en la red TTN, a la que se vincularán los dispositivos y que se encargará de descifrar la información enviada por ellos. Para ello, en la plataforma de TTN se creará la aplicación que tendrá un AppEui, un identificador único de la aplicación que deberá ser configurado en los dispositivos para asociarlos.

A esta aplicación se añadirán dispositivos registrando el DevEui del dispositivo (similar a una dirección MAC de IP, única del dispositivo), definiendo si ese dispositivo tendrá una activación manual **ABP (Activation by Personalization)** u **OTAA (Over The Air Activation)**.

La información enviada por los dispositivos deberá ser decodificada del formato con el que llega, un array de bytes a los datos correspondientes. Para ello, TTN permite definir una función de decodificación que permite convertir estos bytes en campos que posteriormente se puedan enviar, como por ejemplo a formato JSON. Esta función será definida para decodificar los datos enviados por los sensores utilizados en la maqueta.

Para recibir los datos de TTN y enviarlos al bróker de la maqueta se empleará un programa de Python que realizará estas funciones.

9.2.2.2. Configuración sensores TTN

Para el desarrollo de la maqueta se han utilizado varios sensores comerciales LoRaWAN, en particular de la compañía Globalsat. A continuación se indicarán los modelos, los datos que miden y su funcionamiento en general.

Para esta maqueta se definirá una solución que tenga en cuenta todos los dispositivos LoRaWAN utilizados para la maqueta. Por ello, se configurará la utilización de puertos LoRaWAN diferentes en función del tipo de dispositivo. Puerto 2 para balizas GPS y Puerto 3 para sensores de Temperatura, Humedad y CO2 o CO. Este último dato se diferenciará en base a un campo que identifica el modelo de sensor.

En la maqueta se dispondrán varios sensores de la compañía Globalsat, en particular los modelos LS-111, LS-112 y LT-100E. El LS-111 es un sensor de temperatura, humedad y CO2 (Dióxido de Carbono), con el cual se pueden ver los valores en su display y son enviados periódicamente, en un intervalo definido por el usuario mediante LoRaWAN.

El LS-112 es un sensor similar al anterior en cuanto a diseño y funcionalidades, al igual que el LS-111. Este sensor es de Temperatura y Humedad pero en lugar de CO2 éste mide CO (Monóxido de Carbono). Igualmente envía los datos mediante LoRaWAN en intervalos definidos por el usuario. Ambos modelos LS-111 y LS-112 se alimentan conectándolos a una fuente de continua de 8-24V o por micro USB. Estos dispositivos se configurarán mediante una conexión en serie con comandos AT para configurar los parámetros de la red LoRa, y para establecer el periodo de envío de la información, que se fijará en esta maqueta en una transmisión cada 20 minutos, para evitar sobrepasar el 1% de ciclo de servicio por canal.

El dispositivo LT-100E, es una baliza GPS, que envía las coordenadas del dispositivo mediante LoRaWAN. El dispositivo se puede configurar para que transmita periódicamente independientemente de si está en movimiento o no, o que transmita cada cierto tiempo definido si no se encuentra en movimiento y con otro periodo diferente cuando está en movimiento. Adicionalmente, el dispositivo transmite cuando cambia de un modo a otro y tiene varias alertas configurables para informar de que su batería se está acabando. En este caso se ha configurado para enviar su posición periódicamente si no está en movimiento cada 20 minutos y si se encuentra en movimiento cada 30 segundos.

También dispone de un botón que lanza una alerta solicitando ayuda cuando se mantiene pulsado. Al pulsarlo, intenta fijar su posición durante 30 segundos y después procede a enviar 3 mensajes solicitando ayuda.

9.2.3. Lectura de datos contador eléctrico mediante Modbus

Para leer datos del contador eléctrico del fabricante Circutor, modelo CVM-C10, mediante Modbus se utilizará la siguiente estructura.

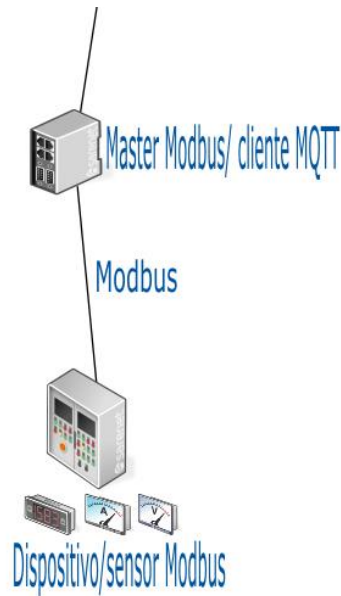


Figura 13 Lectura datos mediante Modbus

Para leer los datos del contador mediante Modbus se utilizará un equipo del fabricante Owasys, Owa3x, que tiene diferentes interfaces para diferentes tipos de señales y se utilizará su puerto RS-485 para leer los datos del contador eléctrico.

Para leer estos datos ha sido necesario desarrollar un programa en C++, utilizando un compilador específico para la arquitectura del Owa3x, utilizando una biblioteca modificada de Modbus en C++ libmodbus, la biblioteca de MQTT de Eclipse Paho en C, una librería de JSON llamada Jansson en C y otra librería de inicialización de Owasys.

El software iniciará la interfaz RS-485 del Owa3x, y se comunicará a través de Modbus a través de esta interfaz. Una vez obtenidos los datos se parsearán y a continuación se enviarán al bróker MQTT en formato JSON, utilizando un fichero campos.txt para obtener los identificadores de los campos.

9.2.4. Lectura de datos contador eléctrico mediante Modbus TCP

En este caso, para leer datos del contador eléctrico del fabricante Circutor, modelo CVM-C10, mediante Modbus TCP se utilizará la siguiente estructura.

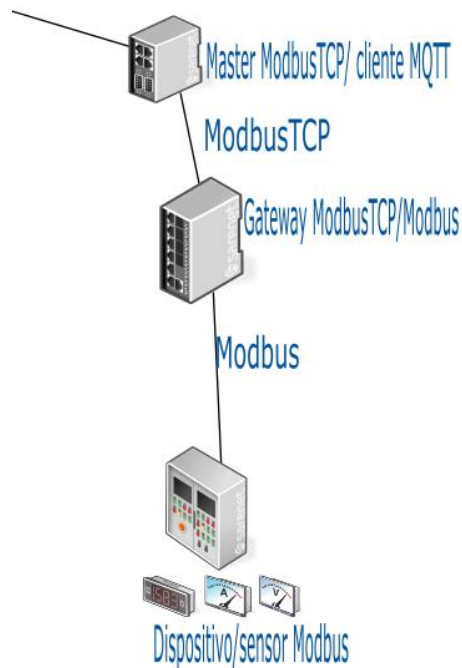


Figura 14 Lectura datos mediante ModbusTCP

Para leer los datos del contador mediante Modbus TCP se utilizará un gateway Modbus TCP-Modbus industrial del fabricante Moxa, MBGate3170, que permite mediante el protocolo Modbus TCP leer datos de un dispositivo Modbus o a la inversa.

En esta forma de leer los datos del contador eléctrico, se ha programado un software similar al del apartado anterior, pero en lenguaje Python y utilizando las librerías pymodbus y MQTT de Eclipse Paho. Adicionalmente, se ha tenido que configurar el gateway indicando que el maestro Modbus está en la interfaz Ethernet y los esclavos en la interfaz RS-485.

9.2.5. Presentación de datos mediante Kibana

Para la presentación de los datos obtenidos, se hará uso de Kibana, una herramienta que permite hacer análisis y búsquedas sobre elementos almacenados en Elasticsearch. Esta herramienta proporciona opciones de búsqueda, filtrado en base a campos de los datos y herramientas que permiten crear visualizaciones de datos. Tiene varias opciones para visualizaciones y se le pueden añadir plugins con más opciones. Entre las básicas hay gráficos de barras, tablas de datos, gráficos de sectores, etc.

Adicionalmente, en este apartado se explicarán los paneles de control generados para la presentación de los datos de la maqueta.

9.2.5.1. Instalación de Plugins para Kibana

La instalación de plugins en Kibana es relativamente sencillo, basta con copiar los ficheros del plugin deseado en la carpeta de plugins de Kibana, indicando en el plugin la versión de Kibana que se utiliza y reiniciar el servicio. Tras unos minutos si el plugin funciona correctamente el servicio Kibana volverá a estar disponible con los plugins instalados.

En particular para presentar los datos de la maqueta se utilizarán los siguiente plugins de visualización:

- Gauge: un medidor que permite medir el valor como si fuera un indicador de nivel, en el cual se indica el valor mínimo y máximo y se muestra el valor medido con diferentes colores en función del valor medido, y también mostrará el valor del porcentaje del indicador.

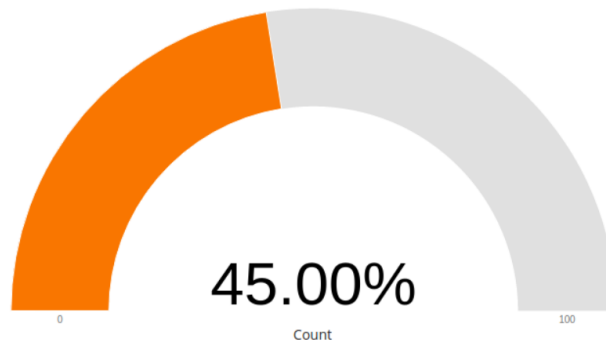


Figura 15 Plugin Gauge

- Custom-gauge: a partir del plugin anterior se ha modificado su código para mostrar en lugar de un porcentaje el valor instantáneo del indicador.

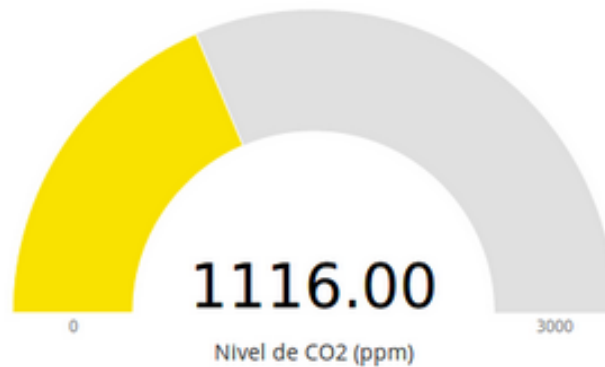


Figura 16 Plugin custom-gauge

- Traffic: Este plugin mostrará un semáforo el cual en función de unos umbrales definidos para el valor de un campo, visualizará un semáforo con una luz roja, verde o naranja.



Max cpu.nice

Figura 17 Plugin traffic

- myTileMap: plugin que modifica una visualización de Kibana, TileMap, la cual muestra la posición de un elemento en un mapa a través de sus coordenadas, o un geoip. En esta modificación se añade una opción para diferenciar por colores las diferentes localizaciones en base a otro parámetro, como por ejemplo el indicador de dispositivo. Por defecto, Tilemap muestra con diferentes colores los posiciones gps en función del número de veces que esta almacenada esa posición para el intervalo de tiempo visualizado.

9.2.5.2. Diseño de paneles de control

Los datos obtenidos de los dispositivos de la maqueta se presentarán mediante paneles que utilicen visualizaciones gráficas para presentar el estado actual de una forma sencilla y clara. Se diseñarán diferentes paneles de control para los diferentes datos medidos en la maqueta:

- Un panel de control para monitorizar la calidad de aire, midiendo temperatura, humedad y CO2.

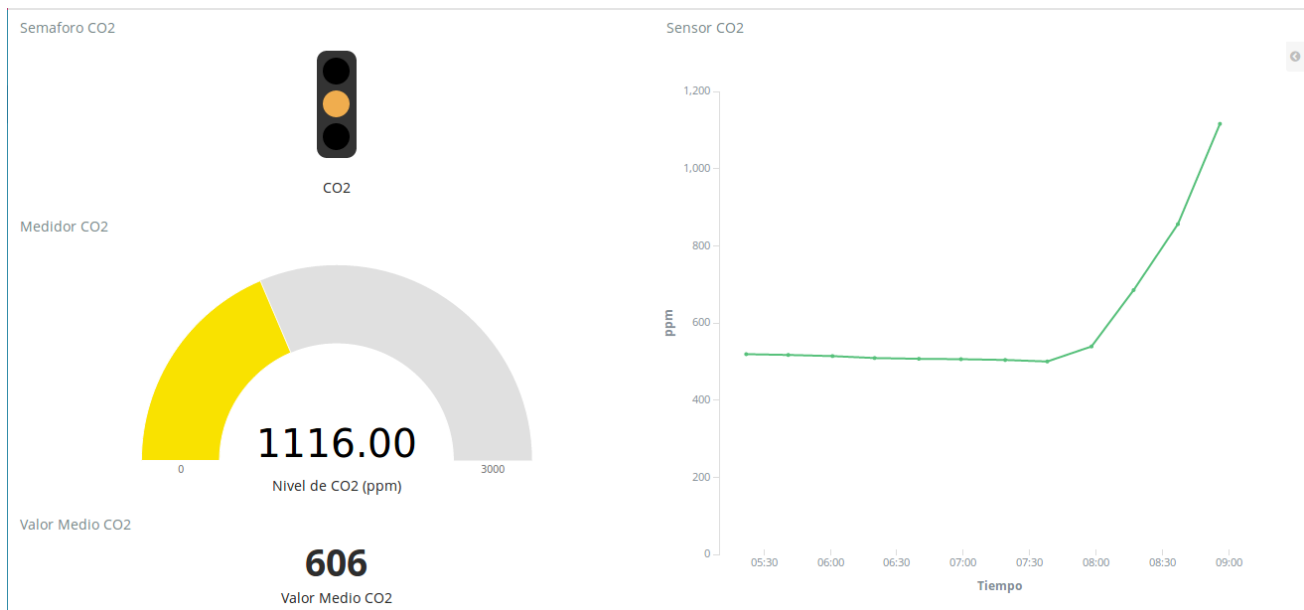


Figura 18 Panel de control nivel de CO2

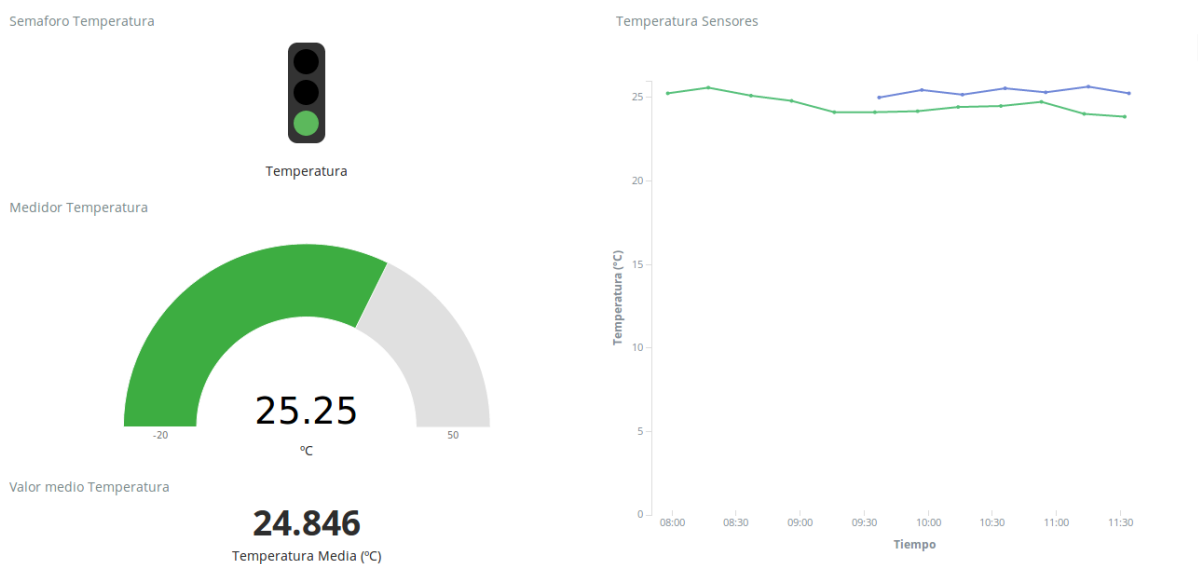


Figura 19 Panel de control temperatura

Como se puede observar se han utilizado los plugins mencionados anteriormente junto con una gráfica que muestra un histórico del nivel de CO2 y la temperatura (en este caso las últimas 4h). Este intervalo de datos es seleccionable desde Kibana.

- Un panel de control que muestre la localización de las balizas GPS en un mapa y su nivel de batería.

Adicionalmente, para la representación de la localización gps en un mapa ha sido necesario modificar el mapa utilizado en la aplicación dado que no ofrecía una precisión suficiente para el zoom. Por ello, se ha configurado los datos de un servidor WMS en la visualización TileMap de Kibana. Este servidor WMS, es parte de Openstreetmaps, proyecto colaborativo para crear mapas libres y editables. En particular el mapa es de OWS Terrestres [14].

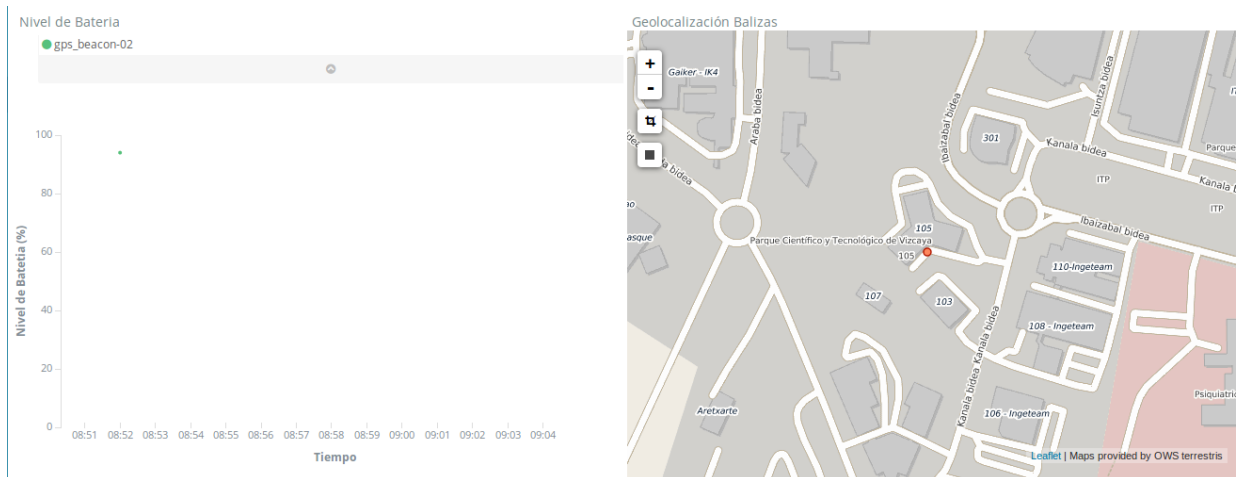


Figura 20 Panel de control balizas GPS

Como se observa en esta Figura 20, se observa en una gráfica una representación del nivel de batería de las balizas en el tiempo, y un mapa en el que se muestra la posición de la baliza. Adicionalmente una pestaña donde se pueden observar todas las entradas referentes a la baliza en el periodo de tiempo especificado.

- Un panel de control que muestre la potencia activa, la potencia reactiva y el factor de potencia del contador eléctrico.

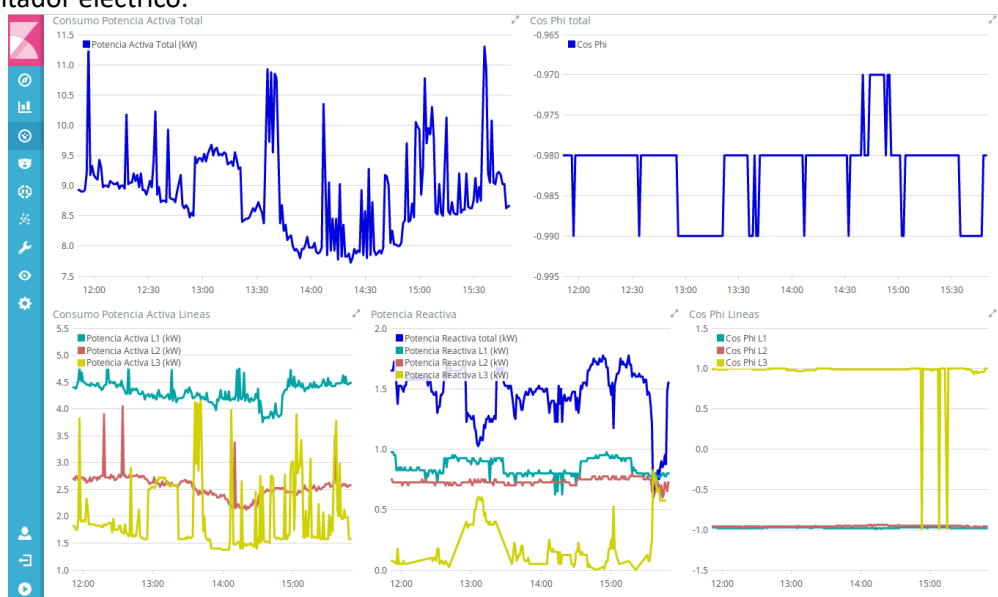


Figura 21 Panel de control medidor eléctrico

En este panel de control se pueden observar varias graficas en las que se muestran la potencia activa, el factor de potencia y la potencia reactiva calculada, para las tres líneas que mide el contador CVM-C10 y el total. Así como el Cos ϕ de todas las líneas y el total.

La potencia reactiva se ha calculado a partir de los valores de potencia inductiva y capacitiva medidos por el contador.

- Adicionalmente, se pueden generar gráficos estadísticos de LoRaWAN, mostrando la utilización de los canales por dispositivos, los niveles de señal recibidos y la relación señal a ruido. A continuación se pueden observar algunos posibles paneles de control para las diferentes medidas de LoRaWAN.

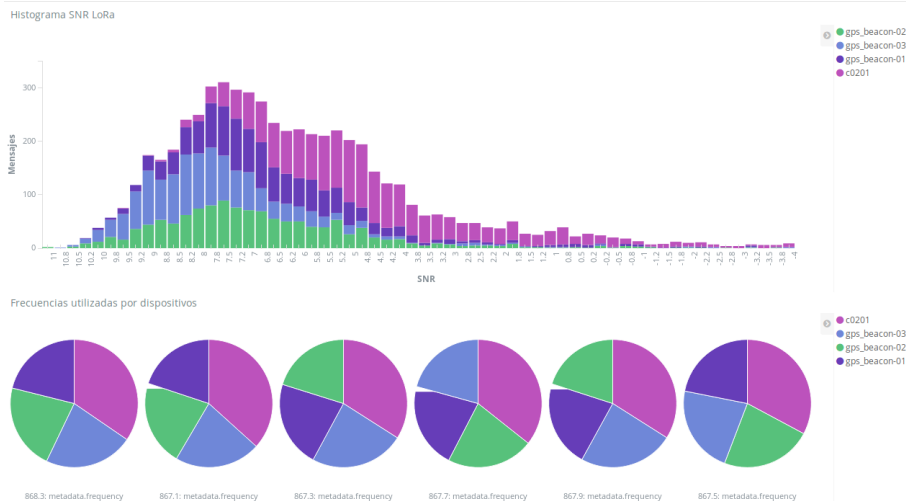


Figura 22 Panel de control estadísticas LoRaWAN

9.3.Seguridad

Como se mencionó anteriormente en los objetivos, la seguridad es un factor de gran importancia en este TFM. Por ello la arquitectura diseñada y la maqueta desarrollada deben asegurar que los compromisos de seguridad se cumplan. A continuación se explicarán los objetivos de seguridad y los mecanismos empleados en el diseño de la arquitectura empleados.

9.3.1. Separación de equipos generadores y consumidores de datos

Como ya se ha mencionado en el contexto, este proyecto se enfoca hacia el sector industrial. Es por ello que aislar a los equipos que recogen los datos de aquellos que los almacenan es tan importante. Dado que en el caso de Industria aquellos equipos de los que se quieren recoger los datos son maquinaria de fabricación o elementos muy sensibles y delicados, un fallo en la seguridad podría afectar al proceso de fabricación y suponer a las empresas afectadas pérdidas económicas e incluso materiales muy elevadas.

También se podrían provocar grandes daños si los datos fueran manipulados o erróneos, cuando se utilizan los mismos para la monitorización de los procesos de fabricación. O si personas sin la autorización debida pudieran acceder a estos datos.

Por este motivo, en esta arquitectura se aísla a los equipos que generan los datos (un equipo en la fábrica o la maquinaria de la misma) de aquellos equipos que la consumen (el sistema de almacenamiento y presentación) para que en caso de que alguno de los equipos sea atacado no ponga en peligro a los demás.

En esta arquitectura se establece un bróker de mensajes que se encarga de enviar los datos entre ellos, de forma que este equipo sirve como una frontera entre ellos. Los equipos solo conocen al bróker con el que se comunican, y desconocen si existe algún otro equipo que realice funciones iguales o similares a las suyas.

9.3.2. Disponibilidad del servicio

La utilización de un bróker supone que hay un único punto de fallo, motivo por el cual en la arquitectura se define un bróker de mensajes redundante para que en caso de producirse algún fallo en uno de los brókers, otro bróker tome su lugar. Al utilizarse equipos redundantes se asegura la disponibilidad del servicio incluso cuando falle un bróker.

9.3.3. Confidencialidad e Integridad de los datos

Como se ha mencionado antes, la Integridad y confidencialidad de la información es muy importante, por lo que se deberán aplicar mecanismos de cifrado, para enviar los datos y utilizar certificados para asegurar que la fuente del mensaje no es un impostor junto con otros mecanismos para lograr asegurar su integridad y confidencialidad.

La confidencialidad de la información es un motivo de gran importancia en el envío de la información, en particular, entre los sensores o dispositivos de medida y la arquitectura de recogida, almacenamiento y presentación. Por ello, se deberán cifrar todos los mensajes intercambiados entre estos sensores y el bróker de mensajes de la arquitectura. Adicionalmente, en el sistema de presentación se debería implementar un sistema de control de acceso para que los usuarios solo puedan acceder a los paneles de control propios y no a los de otros usuarios.

La integridad de los datos en su envío entre los sensores y la arquitectura de recogida, almacenamiento y presentación también es un factor importante, dado que datos erróneos podrían ocasionar problemas a los usuarios de estos servicios si se produce una falsa alarma o se omite una situación peligrosa, por un error o una manipulación del mensaje recibido. Por ello, se deberán aplicar mecanismos de seguridad y tecnologías que comprueben errores en la transmisión de los datos, mediante la utilización de protocolos como TCP para la transmisión que permite detectar errores en el envío de los paquetes y mecanismos de seguridad como la firma de los mensajes para permitir detectar si un mensaje ha sido alterado y asegurar que el remitente es quien ha enviado el mensaje y no un impostor.

Adicionalmente, la utilización de sellos de tiempo para evitar ataques de repetición de mensajes podría ser otra medida de seguridad para evitar la recepción de datos no correctos.

9.3.4. Mecanismos de cifrado empleados en la maqueta

En particular, en la maqueta desarrollada se implementan varios cifrados de la información en diferentes partes de la arquitectura de la maqueta. En la propia especificación de LoRaWAN al enviar los datos se aplica un cifrado de la información y adicionalmente al recoger los datos de TTN y al enviarlos al Bróker MQTT se aplica TLS, lo que permite asegurar la confidencialidad de los datos.

Estos dispositivos se pueden configurar de dos formas definiendo si ese dispositivo tendrá una activación manual **ABP (Activation by Personalization)** u **OTAA (Over The Air Activation)**.

En el caso de configurarse como ABP, se definirán unas claves de cifrado de sesión, AppSKey y NwkSKey, fijadas por el usuario y posteriormente deberán ser configuradas en el dispositivo, junto con la AppEui. Por otro lado, si se registra el dispositivo para realizar OTAA como se ha hecho en la maqueta, se deberá registrar la DevEui y una AppKey en TTN, y posteriormente se deberá configurar la AppKey y AppEui. Una vez hecho esto, cada vez que el dispositivo se encienda enviará un mensaje de Join a la red, mediante el

cual se acordará con el servidor de aplicaciones las claves de cifrado, AppSKey y NwkSKey. Estas claves se explicarán con mayor detalle en el apartado de seguridad.

Los dispositivos LoRaWAN al realizar la activación utiliza una clave **AppKey** única para cada dispositivo a partir de la cual se generan las claves utilizadas en la sesión y para cifrar el intercambio de mensajes durante la negociación de estas claves. Estas claves de sesión son **NwkSKey (Network Session Key)** y **AppSKey (Application Session Key)**, clave de cifrado entre el dispositivo y el servidor de red LoRaWAN y clave de cifrado entre el dispositivo y el servidor de aplicaciones LoRaWAN, respectivamente.

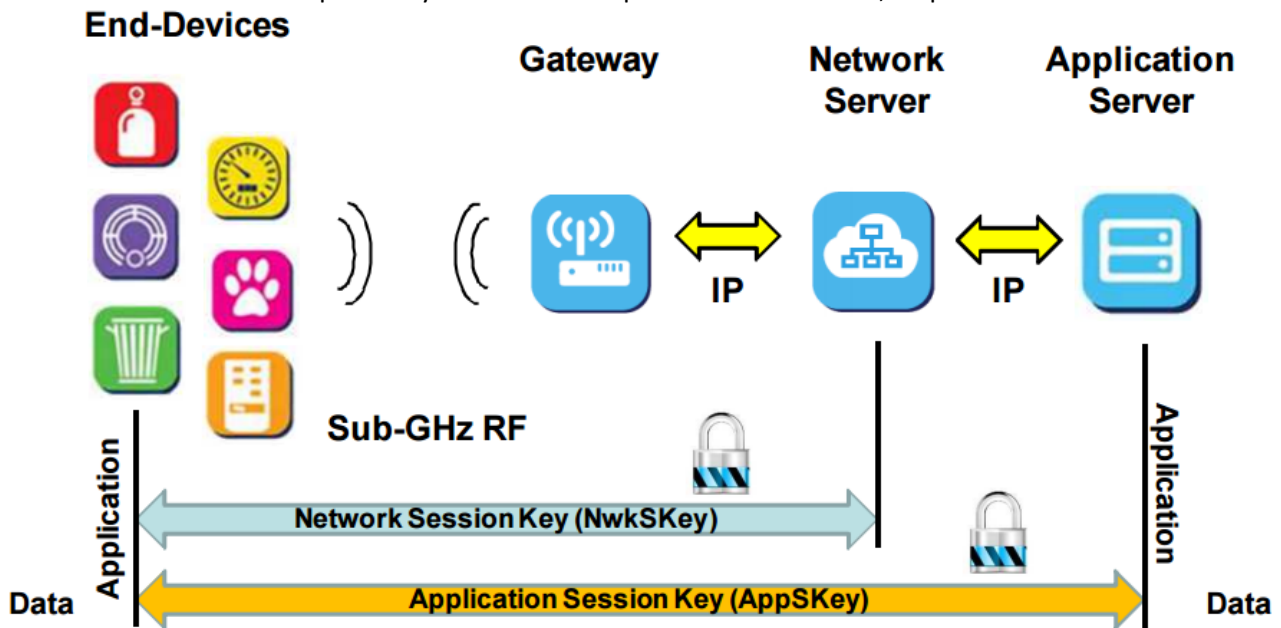


Figura 23 Claves cifrado LoRaWAN

Como se puede apreciar en la Figura 23 la clave NwkSKey se utiliza para cifrar todas aquellas comunicaciones entre el servidor de red y el dispositivo. Por ejemplo para comprobar la validez de un mensaje. La clave AppSKey se utiliza para cifrar la carga útil de los mensajes LoRaWAN. En el caso de la maqueta esta carga se descifrará en los servidores de TTN, a continuación los datos se ponen a disposición de un bróker MQTT al que se le pueden solicitar los datos utilizando TLS, y un certificado proporcionado por TTN. Para solicitar los datos se ha de indicar un usuario definido por el identificador que tiene la aplicación en TTN y una clave definida por el usuario en la aplicación de TTN.

Para enviar los datos al sistema de almacenamiento se vuelve a utilizar MQTT. Se envían utilizando TLS con lo que los datos se mantienen cifrados siempre que se envían a través de la red. Con esto se puede asegurar la confidencialidad e integridad de los datos.

10. Coste del TFM

En este apartado se presenta el descargo de gastos derivado de la realización de este TFM. En él se desglosan los costes de los equipos utilizados en la maqueta, los costes de materiales amortizables utilizados en el proyecto, gastos materiales no reutilizables y los gastos en recursos humanos derivados de la realización del proyecto. Finalmente se mostrará una tabla con todas las partidas y se presentará el coste total del proyecto.

10.1. Coste Recursos Materiales

En este punto se expondrán todos los gastos y costes derivados de los materiales necesarios para llevar acabo la realización del proyecto.

10.1.1. Coste equipos

En este apartado se detallarán los costes de los equipos necesarios para la realización de la maqueta y arquitectura. En particular, sensores, un equipo embebido industrial, un Gateway Modbus TCP y un Gateway LoRaWAN tal y como se puede ver en la Tabla 9 Coste Equipos.

Tabla 9 Coste Equipos

Equipo	Modelo	Precio unitario	Unidades	Total (€)
MultiConnect® Conduit™ Programmable Gateway for the Internet of Things 3G (HSPA+)	MTCDT-H5-210L-US-EU-GB	419,00 €	1	419,00 €
LoRa mcard 868 MHz	MTAC-LORA-868	169,91 €	1	169,91 €
LoRaWAN compliant GPS tracker	LT-100E	65,80 €	3	197,40 €
GlobalSat CO2 and Temperature & Humidity Sensor - LS-111	LS-111	131,60 €	2	263,20 €
GlobalSat CO and Temperature & Humidity Sensor – LS-112	LS-112	114,06 €	1	114,06 €
Owasys owa3x	Owa3x	149,00 €	1	149,00 €
MGate MB3170 -1 Port RS-232-422-485 advanced serial communication Gateway	MB3170	204,00 €	1	204,00 €
Subtotal				1.516,57 €

El subtotal de esta partida es de 1516,57€.

10.1.2. Material amortizable

En este apartado se detallarán los costes del material amortizable utilizado para la realización del proyecto, con un periodo de amortización de 2 años. A continuación en la Tabla 10 Coste Material Amortizable se pueden ver los materiales utilizados.

Tabla 10 Coste Material Amortizable

Concepto	Precio Unidad(€)	Unidades	Total(€)	Periodo de Utilización(meses)	Total(€/mes)	Total(€)2
Pc Sobremesa	1.200,00 €	1	1.200,00 €	9	50,00 €	450,00 €
Servidor	3.800,60 €	1	3.800,60 €	6	158,36 €	950,15 €
Total					208,36 €	1.400,15 €

El subtotal de esta partida son 1400,15€.

10.1.3. Gastos

En este punto se detallarán los gastos realizados durante la ejecución de este proyecto. Se tratan en su mayoría de materiales fungibles que no serán reutilizados para otros proyectos. El desglose de estos gastos se presenta en la Tabla 11 Gastos .

Tabla 11 Gastos

Concepto	Coste (€/mes)	Duración(Meses)	Total(€)
Material de oficina	5,00 €	9	45,00 €
Desplazamiento	50,00 €	9	450,00 €
Otros	60,00 €	9	540,00 €
Conexión a Internet	50,00 €	9	450,00 €
Total			1.485,00 €

El subtotal de esta partida es de 1485€.

10.2. Coste Recursos Humanos

A continuación se desglosan los costes en recursos humanos derivados de la mano de obra durante la realización de este TFM.

Para la realización del proyecto, se ha contado con un Ingeniero en Tecnologías de Telecomunicación, con una tasa horaria estipulada de 20€ la hora, un ingeniero Senior y un director de proyecto, con una tasa horaria de 40€/h y 50€/h, respectivamente. En base a las horas necesarias para completar el proyecto en la Tabla 12 Coste Recursos Humanos se observa a continuación el coste en mano de obra.

Tabla 12 Coste Recursos Humanos

Cargo	Coste(€/h)	Carga Trabajo(h)	Total(€)
Directora Proyecto	50,00 €	70	3.500,00 €
Ingeniero Sénior	40,00 €	150	6.000,00 €
Ingeniero Junior	20,00 €	660	13.200,00 €
Total			22.700,00 €

Como se puede observar el coste total de esta partida es de 22.700€.

10.3. Costes Totales

A continuación se realizará un resumen de todos los costes de la realización del trabajo, y se procederá a calcular el coste total junto con el IVA en la Tabla 13 Costes Totales.

Tabla 13 Costes Totales

Recursos	Total (€)
Equipos	1.516,57 €
Material amortizable	1.400,15 €
Gastos	1.485,00 €
Recursos Humanos	22.700,00 €
Total sin IVA	27.101,72 €
IVA(21%)	5.691,36 €
Total con IVA	32.793,08 €

Como se puede observar el coste de la realización de todo el trabajo es de 27.101€ sin IVA, aplicando un IVA del 21% se obtiene un coste de 32.793,08€.

11. Conclusiones

En este Trabajo Fin de Master se ha llevado a cabo el diseño de una arquitectura para la recogida, almacenamiento y presentación de los datos recogidos por diversos sensores IoT. También se ha diseñado y desplegado una maqueta de esta arquitectura en la que se recogen datos de múltiples fuentes, dispositivos comerciales e industriales que utilizan tanto tecnologías nuevas orientadas al IoT como LoRaWAN, así como tecnologías más extendidas en entornos industriales como Modbus. Los datos de estos sensores son recogidos en la maqueta por una arquitectura de recogida, almacenamiento y presentación, basada en Elasticsearch, Logstash y Kibana.

Como conclusión principal de este TFM, se ha logrado cumplir el objetivo principal que se había establecido para el desarrollo de este proyecto, atendiendo a los plazos planificados y al coste establecido. Se ha diseñado una arquitectura para la recogida, almacenamiento y presentación de los datos de sensores IoT.

Los objetivos secundarios establecidos se han cumplido correctamente, la seguridad de la arquitectura se ha logrado mediante la utilización de un bróker de mensajes que actúa como frontera o intermediario entre el sistema de almacenamiento y presentación, aislando los equipos que envían la información de aquellos en los que se recoge, almacena y presenta. Adicionalmente, la implementación de mecanismos de seguridad como el cifrado y la firma de mensajes garantiza la confidencialidad e integridad de los datos entre los sensores y la arquitectura.

El objetivo del análisis de la viabilidad de la arquitectura planteada se ha completado mediante el diseño y desarrollo de una maqueta de la arquitectura, que ha permitido comprobar su viabilidad. Para este objetivo se habían definido una serie de objetivos parciales los cuales se han alcanzado tal como se puede comprobar en la descripción de la solución en la que se puede analizar el diseño de esta maqueta, así como una descripción del proceso y pasos seguidos para su implementación.

Como conclusión final se puede decir que la arquitectura presentada en este TFM permitirá recoger y almacenar datos de múltiples fuentes y diversos ámbitos, de una manera simple y eficaz. También permitirá consultar estos datos de una forma sencilla y comprensible mediante la utilización de visualizaciones gráficas de los datos. Adicionalmente, esta arquitectura proporciona seguridad tanto de las fuentes de los datos como del sistema de almacenamiento al aislar las dos partes, así como mecanismos de seguridad que permitirán garantizar la integridad y la confidencialidad de los datos en su transmisión a través de Internet.

Fuentes de Información

- [1] LoRaWAN specification versión 1.0: <https://www.lora-alliance.org/portals/0/specs/LoRaWAN%20Specification%201R0.pdf> [Último acceso 05/06/2015]
- [2] SIGFOX: <https://www.sigfox.com/en> [Último acceso 05/06/2015]
- [3] MQTT standard specification: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf> [Último acceso 05/06/2015]
- [4] Pila ELK, Elasticsearch Logstash Kibana: <https://www.elastic.co/> [Último acceso 05/06/2015]
- [5] CoAP (RFC 7252) specification: <https://tools.ietf.org/html/rfc7252> [Último acceso 16/06/2015]
- [6] Servidor WMS de Terrestris: <http://ows.terrestris.de/dienste.html#openstreetmap-wms> [Último acceso 18/06/2015]
- [7] The Things Network TTN: <https://www.thethingsnetwork.org/> [Último acceso 10/07/2017]
- [8] AWS IoT: <https://aws.amazon.com/es/iot-platform/> [Último acceso 12/07/2017]
- [9] Servicios IoT de Azure: <https://docs.microsoft.com/es-es/azure/iot-suite/iot-suite-overview> [Último acceso 12/07/2017]
- [10] Google Cloud Platform: <https://cloud.google.com/solutions/iot/?hl=es> [Último acceso 12/07/2017]
- [11] LoRa Alliance: <https://www.lora-alliance.org/> [Último acceso 15/08/2017]
- [12] BOE-A-2013-4845: <https://www.boe.es/buscar/doc.php?id=BOE-A-2013-4845> [Último acceso 27/08/2017]
- [13] LORIoT: <https://www.loriot.io> [Último acceso 27/08/2017]
- [14] OWS Terrestris: <https://ows.terrestris.de/> [Último acceso 27/08/2017]
- [15] Mobile World Congress: <https://www.mobileworldcongress.com/> [Último acceso 15/09/2017]

ANEXO 1 Pliego de condiciones. Normativa aplicable

En este anexo se recoge la normativa aplicable relevante para este TFM. Dado que este trabajo ofrece una arquitectura con la que recoger datos de múltiples fuentes, con el fin de ofrecer un servicio para la recogida, almacenamiento y presentación de datos de diversos sensores IoT. Muchos de estos datos podrían tener un carácter confidencial o privado y por ello será necesario tener en cuenta la normativa aplicable a la recogida y almacenamiento de estos tipos de datos.

No obstante, no existe una legislación como tal para la recogida y almacenamiento de este tipo de datos, a diferencia de los datos de carácter personal los cuales si están recogidos en la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. Por eso se asume que las obligaciones y requisitos para recoger y almacenar estos datos y el uso que se hace de ellos serán recogidos en el contrato realizado al solicitar un servicio de este tipo.

Adicionalmente, en la maqueta se utiliza un esquema de modulación que hace uso de la banda de 868 MHz, por lo que habrá que tener en cuenta el BOE-A-2013-4845 [12] donde se establecen las limitaciones de utilización de esta banda **ISM (Industrial, Scientific and Medical)**:

- Potencia de transmisión de hasta 25mW.
- Utilización de técnicas de acceso y mitigación de interferencias.
- No sobrepasar el 1% de ciclo de trabajo en la banda de 865-868 MHz.

Todas estas condiciones son tenidas en cuenta por el propio protocolo LoRaWAN que tiene las mismas restricciones y añade mecanismos de acceso y mitigación de interferencias en el protocolo. Por ello en la maqueta de este TFM se ha optado por utilizar unas frecuencias en las transmisiones que cumplan con el objetivo de no sobrepasar el 1% del ciclo de trabajo.

ANEXO 2 Planos y esquemas

En este anexo se encuentran los diagramas diseñados para la arquitectura de recogida y presentación de medidas de sensores IoT, y el diagrama de la maqueta diseñada para su demostración.

Diagrama de la arquitectura

La arquitectura diseñada deberá ser capaz de recoger datos de fuentes muy diversas de forma sencilla y segura. Estos datos deberán ser almacenados por la arquitectura y se deberá permitir la presentación de ellos de forma visual y simple.

A continuación se puede apreciar la arquitectura diseñada para la recogida, almacenamiento y presentación de los datos de diversos sensores IoT.

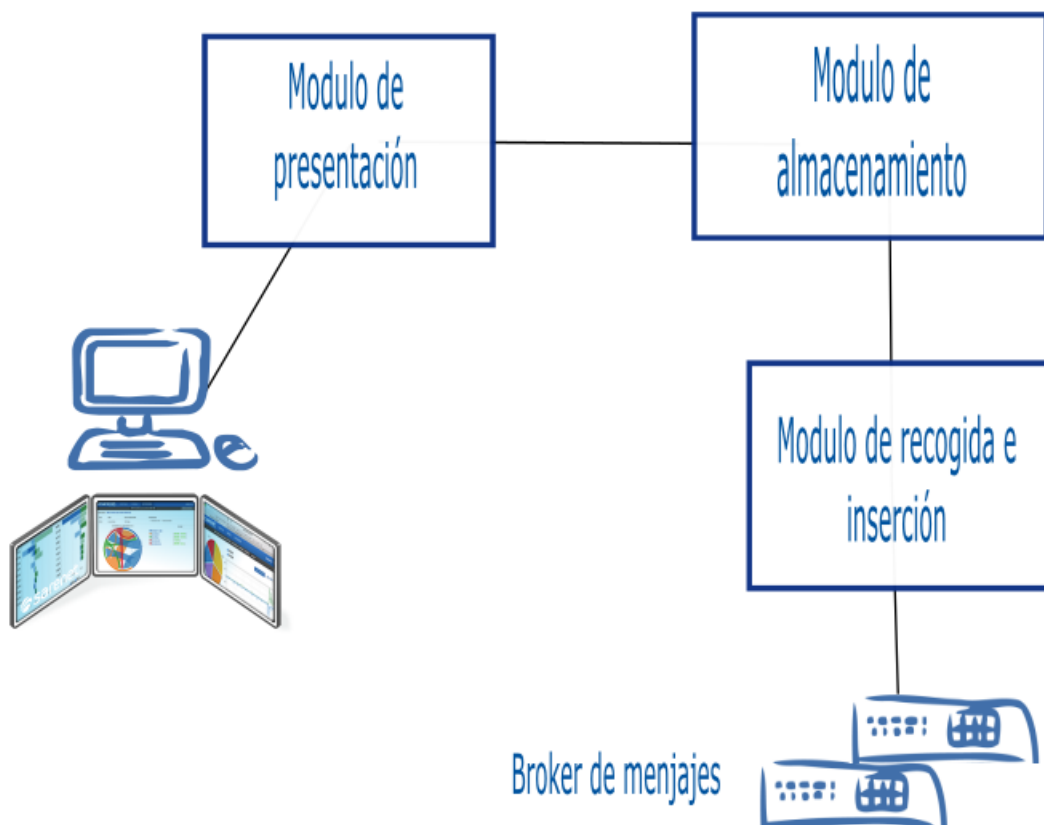


Figura 24 Diagrama de arquitectura de recogida, almacenamiento y presentación

Como se observa en la Figura 24 y como ya se ha explicado en la descripción de la solución, la arquitectura diseñada está compuesta por:

- Un módulo de almacenamiento que procesará búsquedas de datos solicitadas por el módulo de presentación, o cualquier búsqueda solicitada por otra fuente, y almacenará los datos recogidos.
- Un módulo de presentación de datos, con el cual se podrán generar visualizaciones graficas de los datos, y se podrán diseñar paneles de control con las visualizaciones generadas.
- Un equipo que ejecute un módulo de recogida e inserción de datos, el cual contiene un programa cliente, que recibe los datos publicados en el bróker de mensajes. Este módulo se encarga de almacenar los datos en el módulo de almacenamiento.

- Un bróker mensajes redundado implementado mediante un software que recibirá los datos publicados por sus clientes y los reenviará a los clientes que soliciten estos datos.

Diagrama de la maqueta

La arquitectura simplificada utilizada en la maqueta consistirá en implementar un único nodo Elasticsearch, junto al software Kibana y Logstash en una única máquina. Este equipo también ejecutará el programa cliente de MQTT que recibe los datos del bróker MQTT y que en el diseño se ejecuta en el equipo Logstash.

Para demostrar el funcionamiento de esta arquitectura se recogerán medidas de varios sensores y dispositivos. En particular una serie de sensores inalámbricos mediante LoRaWAN a través de TTN. También se recibirán datos de contadores eléctricos a través del protocolo Modbus mediante tecnología Ethernet, para ilustrar mejor las múltiples formas de adquirir los datos de dispositivos Modbus se leerán los datos directamente del dispositivo mediante Modbus y mediante ModbusTCP a través de un Gateway ModbusTCP-Modbus.

Por último, se enviarán los datos obtenidos de los dispositivos a un bróker MQTT mediante los propios programas de lectura en el caso de los dispositivos Modbus, y mediante un script para recoger los datos de TTN y reenviarlos a la arquitectura definida.

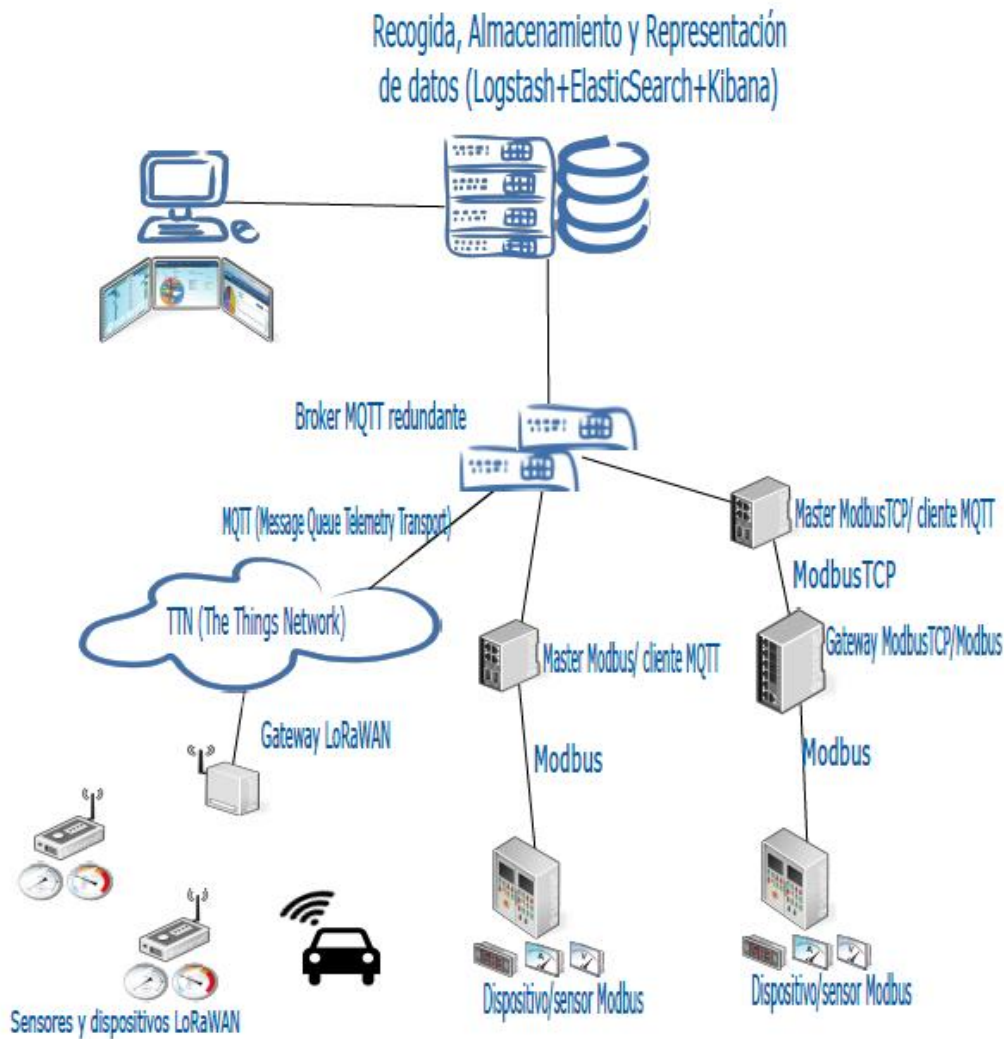


Figura 25 Diagrama de la maqueta

En la Figura 25 se puede observar un diagrama de la maqueta utilizada en este TFM. Esta maqueta se puede dividir en los siguientes módulos:

- La arquitectura de recogida, almacenamiento y representación de los datos.
- La recogida de datos mediante dispositivos LoRa a través de TTN.
- La lectura de datos del contador eléctrico mediante Modbus.
- La lectura de datos del contador eléctrico mediante ModbusTCP.
- Representación de los datos medidos con Kibana.

ANEXO 3 Código y ficheros de configuración

En este anexo se encontrará un desglose de todos los ficheros de configuración utilizados en el despliegue de la maqueta, exceptuando aquellos que permanezcan con valores por defecto, y el código informático desarrollado para la recogida de datos de la maqueta, previamente explicados en el apartado 9 Descripción de la Solución, los cuales se habrán adjuntado con este TFM.

Para empezar en la fi se pueden ver los diferentes ficheros y códigos utilizados, los cuales sus funciones serán descritas a continuación.

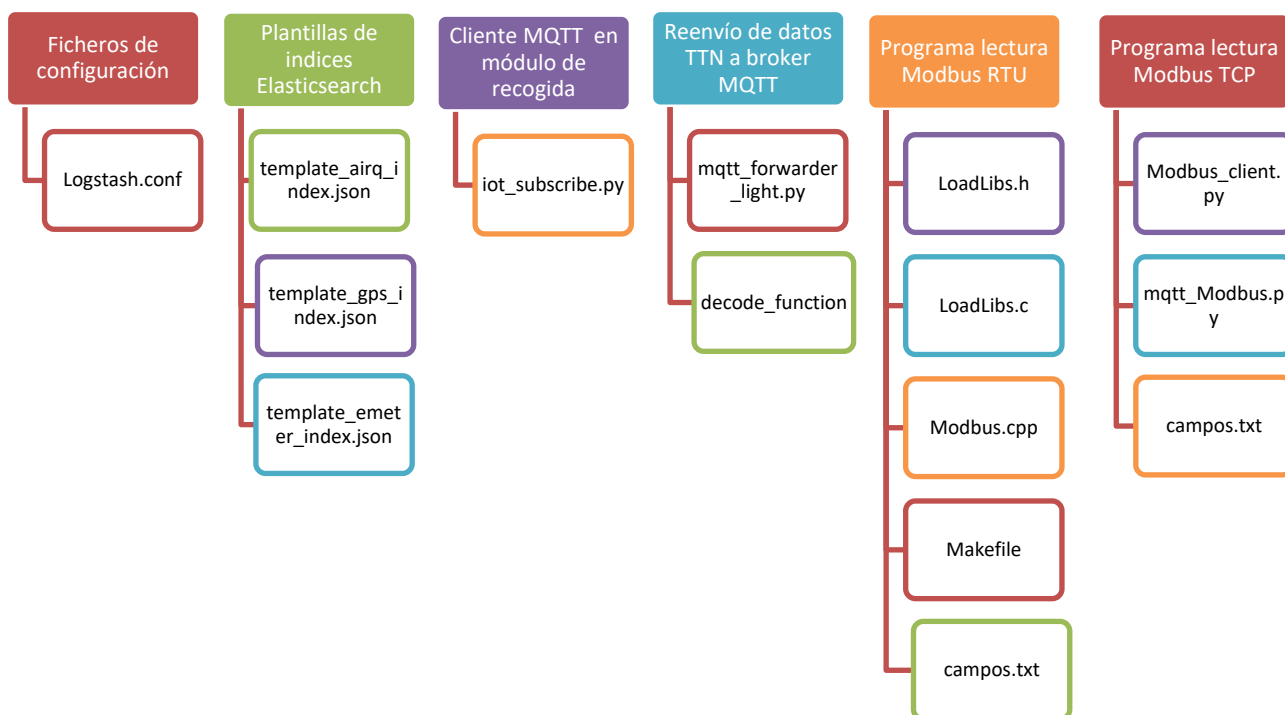


Figura 26 Diagrama de ficheros del TFM

Ficheros de configuración

El fichero Logstash.conf es una versión modificada del fichero de configuración básico en el cual, se han configurado los puertos en los que escuchara el servicio Logstash, así como indicar el índice con el que se insertarán los datos en la base de datos Elasticsearch.

Plantillas de índices Elasticsearch

En estos ficheros se define el mapeo de los campos de los índices utilizados, en ellos se indican los nombres de los campos y el tipo de dato al que pertenecen cada uno. Estos ficheros no son obligatorios, no obstante, permitirán evitar conflictos entre tipos de variables en la base de datos debido a la auto-detección de tipo de variable de Elasticsearch.

Cliente MQTT en módulo de recogida

El script de Python `iot_subscribe.py`, opera en el módulo de recogida de datos subscribiéndose a un tema MQTT y recogiendo los mensajes que recibe del bróker y reenviándolos a Logstash para la inserción en el módulo de almacenamiento de la maqueta basado en Elasticsearch.

Reenvío de datos TTN a bróker MQTT

El scrip de Python `mqtt_forwarder_light.py` recibe los mensajes recibidos a través de LoRaWAN a través de la red TTN y estos son reenviados a través de internet a un bróker MQTT.

Adicionalmente, el fichero `deco_function` contiene una función de decodificación definida en TTN para convertir los datos en hexadecimal recibidos de los sensores a valores numéricos y cadenas de caracteres correspondientes.

Programa lectura Modbus RTU

Este programa realizará la lectura de los registros de un contador eléctrico Circutor CVM-C10, mediante Modbus RTU, con un dispositivo embebido industrial del fabricante Owasys. Para ello utilizará una versión modificada de la librería `libmodbus` de C++ y la librería Eclipse Paho. Este programa está compuesto por los siguientes ficheros:

- `LoadLibs.h`: Fichero de cabecera que permite cargar las librerías propias del dispositivo Owasys, que permite utilizar la interfaz RS-485, desarrollado por Owasys.
- `LoadLibs.c`: Fichero de código que permite cargar las librerías propias del dispositivo Owasys, que permite utilizar la interfaz RS-485, desarrollado por Owasys.
- `Modbus.cpp`: Código de programa que lee los datos de un contador eléctrico CVM-C10 a través de la interfaz RS-485 y enviarlos a través del protocolo MQTT.
- `Makefile`: Fichero con instrucciones de compilación para el programa.
- `Campos.txt`: Fichero utilizado por el programa para almacenar los identificadores de los campos leídos por el programa.

Programa lectura Modbus TCP

Este programa permite leer los registros de un contador eléctrico Circutor CVM-C10 desde cualquier equipo con python a través de un Gateway ModbusTCP-Modbus, para ello utiliza las librerías `mqtt` de Eclipse Paho y `pymodbus`. Está compuesto de los siguientes ficheros en lenguaje C++:

- `Modbus_client.py`: Clase python para realizar la lectura de los registros modbus y su parseo, mediante `modbusTCP`.
- `Mqtt_Modbus.py`: fichero con código que llama a la clase anterior y envía los datos recibidos a través de MQTT.