

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Diseño y desarrollo de una plataforma IoT para la visualización de datos
de maquinarias en las industrias

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Computación

Presentado por:

Miguel Angel Licea Céspedes

Danny Josué Montenegro García

GUAYAQUIL - ECUADOR

Año: 2022-2023

DEDICATORIA

A mis padres, por su amor incondicional y por haber sido mi constante apoyo en los momentos difíciles. A mi tutor, por su sabiduría y orientación en este proceso de investigación. Y a mi pareja, por su comprensión y paciencia durante estos años de trabajo. Les agradezco de corazón por todo su apoyo y dedicación.

Miguel Licea Céspedes

DEDICATORIA

A mis padres, por apoyarme en cada etapa de mi vida que me ha llevado a este momento y amarme más que nadie en el mundo. A mi tutor, por toda la orientación brindada durante el desarrollo de este proyecto. A mis amigos, por sus constantes palabras de aliento.

Danny Montenegro García

AGRADECIMIENTOS

Quiero expresar mi profunda gratitud a todas las personas que han contribuido de alguna manera a la realización de esta tesis. En primer lugar, a mi tutor de tesis, Miguel Realpe, por su valiosa orientación, apoyo y dedicación durante todo este proceso. A Jonathan Paillacho por ser el cliente de esta tesis y ayudarme a desarrollar un proyecto tan interesante y desafiante. A mi familia, por su amor y apoyo incondicional, y por haber sido mi roca sólida durante todos estos años. A mis amigos y colegas, por su compañía y ánimo. Finalmente, agradezco a todos los participantes de la investigación por prestar su tiempo y colaboración. Sin su valiosa contribución, esta tesis no habría sido posible.

Miguel Licea Céspedes

AGRADECIMIENTOS

Agradezco a Miguel Realpe, por todo su tiempo dedicado durante el desarrollo de esta tesis. A Jonathan Paillacho por brindarme su tiempo y conocimiento para desarrollar esta tesis. A mi mamá Yinyer García y mi papá Wilson Montenegro, por siempre estar a mi lado en cada paso que doy y motivarme a dar todo de mí y ser mejor cada día. A mis amigos, por siempre tener las palabras adecuadas para motivarme a seguir adelante.

Danny Montenegro García

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, me(nos) corresponde conforme al reglamento de propiedad intelectual de la institución; Miguel Licea y Danny Montenegro doy(damos) mi(nuestro) consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



Miguel Licea

Danny Montenegro

EVALUADORES

Miguel RealpeMiguel Realpe

PROFESOR DE LA MATERIA / TUTOR

RESUMEN

En la actualidad existe una gran cantidad de plataformas y/o servicios que permiten recolectar datos obtenidos por dispositivos IoT, pero tienen limitaciones respecto a cómo visualizar dichos datos. El objetivo de este proyecto es desarrollar un prototipo de plataforma que permita la visualización de datos de manera sencilla de entender para los usuarios además de disminuir el trabajo de recolección de datos de los dispositivos. Este proyecto fue desarrollado usando React como framework para la página web y la metodología Design Thinking a través de la cual se buscó empatizar con el usuario para desarrollar características en base a sus problemas y necesidades de forma que el producto final cumpla sus expectativas y resuelva los problemas mencionados. Gracias al uso de esta metodología la plataforma desarrollada cumplió con todos los requerimientos del usuario, además hubo una satisfacción de aproximadamente 90% respecto a la facilidad de uso de esta, el 10% restante corresponde a mejoras a realizar en la plataforma tanto en nuevas características como en cambiar el funcionamiento de alguna característica actual. Esto demuestra la utilidad del producto y la importancia que tiene conocer las necesidades del usuario para el desarrollo, pues esto permite obtener un producto que soluciona problemas del usuario, además de la facilidad con la que el usuario puede configurar los dispositivos y visualizar los datos obtenidos por estos.

Palabras Clave: IoT, Plataforma web, Dashboard, Protocolos de comunicación.

ABSTRACT

Currently, there are many platforms and/or services that allow the collection of data obtained by IoT devices, but they have limitations regarding how to visualize said data. The objective of this project is to develop a platform prototype that allows the visualization of data in a way that is easy for the user to understand, in addition to reducing the work of data collection from the devices. This project was developed using React as a framework for the web platform and the Design Thinking methodology through which we sought to empathize with the user and develop features based on their problems and needs so that the final product meets their expectations and resolves the mentioned problems. Thanks to the use of this methodology, the developed platform met all the user's requirements, in addition there was a satisfaction of approximately 90% regarding its ease of use, the remaining 10% corresponds to improvements to be made to the platform in new features and change how an existing feature works. This demonstrates the usefulness of the product and the importance of knowing the user's needs for development, since this allows obtaining a product that solves user problems, as well as the ease with which the user can configure the devices and view the data obtained by these.

Keywords: IoT, Web platform, Dashboard, Communication Protocols

ÍNDICE GENERAL

EVALUADORES	7
RESUMEN.....	I
<i>ABSTRACT</i>	II
ÍNDICE GENERAL.....	III
ABREVIATURAS.....	V
SIMBOLOGÍA	VI
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	IX
CAPÍTULO 1.....	10
1. Introducción.....	10
1.1 Descripción del problema.....	10
1.2 Justificación del problema	11
1.3 Objetivos	11
1.3.1 Objetivo General	11
1.3.2 Objetivos Específicos	12
1.4 Marco teórico	12
1.4.1 Protocolos de comunicación.....	13
1.4.2 Message Queuing Telemetry Transport (MQTT).....	13
1.4.3 Constrained Application Protocol (CoAP)	14
1.4.4 Advanced Message Queuing Protocol (AMQP)	14
1.4.5 Broker: RabbitMQ	15
1.4.6 Soluciones multi-protocolo	15
1.4.7 Software as a Service.....	17

1.4.8	Base de datos	17
CAPÍTULO 2.....		19
2.	Metodología.....	19
2.1	Análisis de Requerimientos.....	19
2.1.1	Definición de usuarios	19
2.2	Requerimientos y funcionalidades	20
2.2.1	Casos de Uso.....	20
2.2.2	Requerimientos funcionales	21
2.3	Diseño de la solución	22
2.3.1	Arquitectura.....	23
CAPÍTULO 3.....		34
3.	Resultados Y ANÁLISIS	34
3.1	Producto final.....	34
3.1.1	Tablero	34
3.1.2	Manejo de los dispositivos.....	36
3.1.3	Manejo del Equipo de Trabajo.....	37
3.1.4	Manejo de Plantillas.....	38
3.1.5	Protocolos de comunicación.....	39
3.2	Evaluación	41
3.3	Análisis de los resultados.....	45
3.4	Análisis de costos	46
CAPÍTULO 4.....		49
4.	Conclusiones Y Recomendaciones.....	49
4.1	Conclusiones	49
4.2	Recomendaciones.....	49
BIBLIOGRAFÍA		51

ABREVIATURAS

IoT	Internet of Things
SaaS	Software as a Service
CoAP	Constrained Application Protocol
AMQP	Advanced Message Queuing Protocol
MQTT	Message Queuing Telemetry Transport
HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
URI	Uniform Resource Identifier
IoTMMB	Internet of Things Multi-Protocol Message Broker
AWS	Amazon Web Services
GCP	Google Cloud Platform
KNX	Konnex
API	Application Programming Interface
RDS	Relational Database Service
EC2	Elastic Compute Cloud

SIMBOLOGÍA

GiB Gibibyte

ÍNDICE DE FIGURAS

Figura 2.1 Diagrama de Casos de Uso	21
Figura 2.2 Diagrama de Bloque	23
Figura 2.3 Diagrama Entidad Relación	25
Figura 2.4 Colección de Datos	25
Figura 2.5 Arquitectura del Sistema	26
Figura 2.6 Diagrama de Despliegue	26
Figura 2.7 Pantalla de Inicio de Sesión	29
Figura 2.8 Pantalla de Registro	29
Figura 2.9 Pantalla Principal	30
Figura 2.10 Agregar nuevo Gráfico	30
Figura 2.11 Pantalla de Equipo	31
Figura 2.12 Invitar participante al Equipo	31
Figura 2.13 Pantalla de Dispositivos	32
Figura 2.14 Agregar nuevo Dispositivo	32
Figura 2.15 Pantalla de Plantillas	33
Figura 2.16 Agregar nueva Plantilla	33
Figura 3.1 Tablero de usuario sin Plantillas activas	35
Figura 3.2 Tablero de usuario sin Gráficos	35
Figura 3.3 Formulario para agregar nuevo Gráfico	35
Figura 3.4 Tablero con Gráficos agregados	36
Figura 3.5 Visualización de Plantillas	36
Figura 3.6 Creación de Nueva Plantilla	37
Figura 3.7 Pantalla de Equipo	37
Figura 3.8 Invitación de un nuevo Participante	38
Figura 3.9 Registro de nuevo Participante del Equipo	38
Figura 3.10 Pantalla de Administración de Plantillas	39
Figura 3.11 Mediciones a través del Protocolo AMQP	40
Figura 3.12 Mediciones a través del Protocolo MQTT	40
Figura 3.13 Resultados de la Evaluación Pregunta 1	42
Figura 3.14 Resultados de la Evaluación Pregunta 2	43

Figura 3.15 Resultados de la Evaluación Pregunta 3.....	43
Figura 3.16 Resultados de la Evaluación Pregunta 4.....	44
Figura 3.17 Resultados de la Evaluación Pregunta 5.....	44

ÍNDICE DE TABLAS

Tabla 3.1 Formulario para la Evaluación de la Plataforma.....	41
Tabla 3.2 Costo de Diseño, Desarrollo y Despliegue de la Plataforma	47
Tabla 3.3 Costo Operacional de la Plataforma	47

CAPÍTULO 1

1. INTRODUCCIÓN

1.1 Descripción del problema

En la actualidad el término IoT (Internet of Things) está transformando la forma en la que interactuamos con dispositivos o sistemas inteligentes tanto en el sector industrial como en nuestros hogares. Esta tecnología permite conectar el mundo real y el internet a través de objetos físicos que embeben sensores inteligentes. El auge de estos dispositivos interconectados va en constante aumento, por lo que para el año 2025 se espera más de 26 billones de dispositivos conectados incluyendo casas, teléfonos, autos y fábricas [1]. Estos dispositivos inteligentes que conforman el IoT pueden utilizarse de forma generalizada para recolectar los datos sobre el terreno objetivo.

A su vez, dentro del sector industrial ha surgido la nueva revolución industrial conocida como la Industria 4.0. La misma consiste en la digitalización de procesos industriales por medio de la automatización y el empleo de máquinas y fábricas inteligentes, siendo los datos del funcionamiento el factor más importante, capaz de mejorar la forma en la que se producen bienes de forma eficiente y productiva durante toda la cadena de valor. Además, la gran cantidad de información que nos proporcionan las industrias es la principal protagonista en el avance a una industria conectada. Con la ayuda de este recurso y la Analítica de Datos es posible tomar decisiones inteligentes de cualquier fábrica o industria [2].

Existe gran sinergia entre IoT y la Industria 4.0 mediante la recopilación de datos de las maquinarias y el monitoreo de los parámetros de control de los equipos para optimizar su rendimiento. Lo que permitiría detectar cuando los equipos funcionan fuera de los parámetros regulares y contar con sugerencias para la toma de decisiones y la implementación de mejoras a futuro.

Actualmente existen SaaS (Software as a Service) que proporcionan diferentes servicios capaces de realizar diversas tareas necesarias durante el proceso de recolección, almacenamiento y visualización de los datos necesarios para el

monitoreo, pero no cuentan con una sencillez para el monitoreo. También varias plataformas han creado diferentes soluciones para permitir una integración más sencilla de grandes fábricas hacia cada plataforma web mediante arquitecturas que conectan los diferentes servicios de uno o varios SaaS.

1.2 Justificación del problema

Las plataformas IoT actuales colaboran a crear entornos especializados en usos específicos para distintos tipos de grandes empresas, por lo que en fábricas pequeñas y medianas es complicado el uso y aprovechamiento de esta tecnología. También existen pocas plataformas que permiten a sus usuarios la capacidad de crear sus propios paneles de visualización o existen límites con respecto a los gráficos necesarios para el proceso de analítica de datos. A pesar de que estas plataformas ya cuentan con algunos de los protocolos de comunicación más usados en el IoT, existen otros como CoAP y AMQP que no son parte de su catálogo para la comunicación y cuentan con una amplia gama de usos y funcionalidades que aportan grandes beneficios a la Industria 4.0.

Por ello, las ventajas del IoT acompañado de la nueva revolución industrial implementado en el sector industrial ayudan a mejorar la eficiencia durante todo el proceso de la cadena de valor. De este modo, este proyecto contribuirá a diseñar e implementar una plataforma IoT de visualización de datos de los equipos en las industrias del Ecuador que permita el aprovechamiento de las nuevas tecnologías relacionadas al IoT para el crecimiento del sector manufacturero ecuatoriano.

1.3 Objetivos

1.3.1 Objetivo General

Diseñar e implementar un prototipo funcional de una plataforma IoT de visualización de datos de maquinarias en las industrias del Ecuador.

1.3.2 Objetivos Específicos

- Desarrollar un prototipo de una plataforma IoT que sea capaz de mostrar mediante un dashboard los datos del funcionamiento en tiempo real de las maquinarias en las industrias.
- Desarrollar un servicio en la nube que pueda conectarse a distintos equipos de forma agnóstica con respecto al funcionamiento y protocolo de comunicación.

1.4 Marco teórico

Evaluar el rendimiento de maquinaria es una tarea compleja ya que es necesario recolectar, procesar y analizar los datos obtenidos de las máquinas para poder tomar una decisión, especialmente cuando existen condiciones cambiantes como la demanda de productos a producir, fallos en la maquinaria o reconfiguraciones del sistema. Además, siempre se necesita un valor de rendimiento de referencia para saber si la máquina cumple las expectativas [3].

Por otro lado, evaluar la condición actual de la maquinaria ayuda a predecir si esta puede presentar fallos en un futuro cercano, esto se puede conseguir en tres pasos: Extraer las características a analizar, realizar un diagnóstico en base a los datos obtenidos y realizar una predicción [4]. Extraer las características y realizar un diagnóstico permiten encontrar anomalías, así como localizar la ubicación donde se puede presentar una falla.

Los métodos para monitorear la condición de maquinaria se pueden dividir en tres categorías [5]:

- Basado en el modelo: Se utilizan las características físicas de la máquina para establecer un modelo matemático. La desventaja de este método es que difícil determinar un modelo matemático para maquinaria compleja.
- Estadístico: Se utilizan datos históricos para determinar si una máquina está presentando fallos.
- Inteligencia artificial: Utilizado para maquinaria compleja en la que se requiere conocer cómo se puede comportar en el futuro.

El monitoreo del estado de maquinaria generalmente se realiza conectando sensores a la maquinaria deseada, se envían los datos a través de la red a una computadora central y se analizan los datos. El principal problema es que los sensores se encuentran conectados mediante cables (o similares) lo que limita su uso a máquinas que van a estar en un solo lugar [6]. Debido a esta limitación se han propuesto soluciones como sensores inalámbricos los cuales, además de resolver el problema mencionado, también ayudan a tener control sobre maquinaria que se encuentre en áreas inaccesibles o áreas que sean difíciles de ser monitoreadas por personas [7].

1.4.1 Protocolos de comunicación

Para recolectar información a través de sensores inalámbricos se consideran algunos aspectos como [8]:

- Lo que se va a medir, por ejemplo, la temperatura de un motor o la velocidad de una turbina.
- Los sensores que se van a encargar de tomar las mediciones.
- Dispositivo(s) que se va(n) a encargar recolectar los datos obtenidos por los sensores.
- El medio del cual se obtendrán las mediciones.
- Lo que se va a realizar con los datos, por ejemplo, enviarlos a la nube o a una computadora central para procesarlos.

En el último punto también se define la forma en la que los datos serán enviados, para ello existen varios protocolos para el paso de mensajes [9], entre estos se encuentran:

1.4.2 Message Queuing Telemetry Transport (MQTT)

Este protocolo provee conectividad entre aplicaciones o usuarios y la red. Sigue una arquitectura publisher/subscriber, dicha arquitectura tiene tres componentes principales:

- Publisher: Envía los datos al broker, por ejemplo, los sensores.

- **Subscriber:** Son las aplicaciones o usuarios que desean obtener información sobre uno o varios tópicos por lo que se conectan al broker y se suscriben a dichos tópicos.
- **Broker:** Se encarga de recibir y clasificar en tópicos los mensajes recibidos por los publisher y enviar dichos mensajes a los suscriber interesados en dichos tópicos.

Es ideal para grandes redes de pequeños dispositivos que necesitan ser monitoreados o contralados usando internet [10].

1.4.3 Constrained Application Protocol (CoAP)

Proporciona una interfaz RESTful de bajo costo computacional respecto al tiempo y memoria. Si bien REST es muy utilizado en aplicaciones web, debido a las limitaciones de hardware que tienen los sensores no es posible utilizarlo en IoT. CoAP se diseñó para resolver este problema, permitiendo que plataformas IoT usen servicios RESTful sin sobrecargar sus recursos. Su arquitectura se divide en dos subcapas:

- **Mensajería:** Encargada de la fiabilidad y duplicación de mensajes.
- **Solicitud/Respuesta:** Encargada de la comunicación.

CoAP utiliza Universal Resource Identifier (URI) en lugar de tópicos, es decir, el publisher envía mensajes al URI y el subscriber se suscribe a un recurso específico del URI [10].

1.4.4 Advanced Message Queuing Protocol (AMQP)

Este protocolo está enfocado en brindar confiabilidad, seguridad e interoperabilidad. Permite comunicación pedido/respuesta y publish/subscribe. Para que exista comunicación el publisher o el subscriber debe crear un “exchange” al cual le asigna un nombre y luego transmitir dicho nombre, este nombre sirve para que publishers y subscribers se encuentren los unos a los otros. Luego un publisher crea una “queue” y la vincula al “exchange”. Todo mensaje que se reciba en el exchange tiene que ser emparejado con una queue a través de un proceso llamado “Binding”.

Debido a la variedad de protocolos que existen las industrias se pueden ver restringidas a usar uno de estos protocolos, incluso si este no se ajusta a todas sus necesidades. Esto puede provocar problemas como [11]:

- Dependencia de proveedor: Debido a la gran variedad de aplicaciones y servicios heterogéneos, formato de datos y protocolos de comunicación los usuarios se ven obligados a adquirir los dispositivos o software de un solo proveedor, lo que implica mayor riesgo de costos operacionales o funcionalidad del producto.
- No existencia de solución multiplataforma: Este problema es desde el punto de vista del desarrollador, ya que los sistemas IoT son diseñados para trabajar con un API específico. Esto dificulta la creación de una solución multiplataforma.

1.4.5 Broker: RabbitMQ

RabbitMQ es uno de los broker más populares, permite enviar y recibir mensajes, además de ser compatible con múltiples protocolos de comunicación, entre ellos MQTT y AMQP. Cuenta con una interfaz que permite monitorear y controlar el broker [12].

1.4.6 Soluciones multi-protocolo

Un aspecto clave para llegar a la interoperabilidad de dispositivos IoT es el uso de un middleware [13] el cual se encarga de proveer una conexión entre dispositivos heterogéneos. Se han propuesto varias soluciones que permiten conectar dispositivos usando diferentes protocolos de comunicación, una de estas es Mainflux [14], la cual permite a dispositivos enviar mensajes a través de HTTP, Web sockets, MQTT y CoAP. Además, se menciona la importancia del almacenamiento de los datos que se obtienen de los dispositivos, para ello hacen uso de Cassandra al ser una base de datos distribuida, escalable y tolerante a fallos. Entre algunos de los aspectos considerados para la creación de esta plataforma se encuentran:

- Rendimiento: Establece un límite superior de 40000 mensajes por segundo para asegurar que se puedan conectar una gran cantidad de dispositivos a la plataforma.
- Conectividad: Debido a que es una solución multi-protocolo debe ser capaz de aceptar mensajes a través de los protocolos mencionados.
- Seguridad: Para proteger los datos de la industria donde se utilice esta plataforma es importante asegurar que únicamente las peticiones realizadas por usuarios autorizados son aceptadas y procesadas. Se sigue la misma lógica respecto a los dispositivos conectados, solo se deben aceptar mensajes de aquellos dispositivos registrados en la plataforma.

Otra solución propuesta es Internet of Things Multi-Protocol Message Broker (IoTMMB) [15], la cual provee un intermediario que permite tanto obtener información por parte de dispositivos IoT que utilizan diferentes protocolos, específicamente HTTP, MQTT y CoAP; como enviar dicha información a una computadora central o a la nube. Se pueden identificar las siguientes etapas:

- Recolección de datos: Los sensores obtienen datos del medio donde se encuentren y los envían a IoTMMB
- Estandarización de los mensajes: El broker se encarga de recibir los mensajes de los sensores y transformarlos en un mismo formato antes de enviarlo a su destino. Para ello primero extrae los datos enviados por el sensor, posteriormente crea una petición HTTP cuyo cuerpo serán los datos y envía dicha petición.

La visualización de los datos es fundamental para obtener información significativa para los usuarios. Esto permite realizar análisis en dichos datos, lo que ayuda a tomar mejores decisiones. Se deben considerar algunos factores fundamentales para la visualización de datos como [16]:

- Saber qué es lo que se quiere comunicar con los datos, es decir, se deben tener objetivos respecto al uso de los datos. Esta información se obtiene a partir de los usuarios y sus necesidades.
- Elegir la manera adecuada de presentar los datos, no deben ser presentados de forma muy compleja ya que puede causar confusión.

1.4.7 Software as a Service

Actualmente existe una gran variedad de ofertas que proporcionan plataformas IoT para la innovación de las fábricas hacia la Industria 4.0. La mayoría de las industrias optan por crear su propia plataforma que conecte sus equipos directamente al internet realizando grandes inversiones para obtener las ventajas del IoT. También existen SaaS (Software as a Service) que proporcionan diferentes servicios capaces de realizar tareas para el almacenamiento y visualización de datos necesarios para el monitoreo desde plataformas web y móviles. Ejemplos de estas plataformas son Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP).

Sin embargo, para desarrollar una Plataforma IoT con estos SaaS se debe contar con una arquitectura que ayude a complementar los diferentes servicios que brindan. Existen diferentes softwares que ayudan a resolver dicha problemática. De las cuales se pueden nombrar: ThingSpeak, una plataforma IoT que permite agregar, visualizar y analizar datos obtenidos en tiempo real. Permite ejecutar código de MATLAB para realizar procesamiento y análisis de datos. Por ello es más utilizado para sistemas IoT que requieren analítica; Thingstream, una plataforma de entrega basada en la nube con interfaz de administración para servicios IoT. Comprende conectividad, seguridad y un broker MQTT de nivel empresarial; OpenRemote, una plataforma IoT que permite a los usuarios integrar dispositivos independientemente del protocolo, específicamente HTTP, MQTT y Konnex (KNX). Además, permite a los usuarios rediseñar la interfaz.

1.4.8 Base de datos

Existen dos tipos de base de datos: relacionales y no relacionales. Una base de datos relacional es útil para realizar consultas de datos con relaciones entre ellos, por ejemplo, relaciones uno a uno o uno a muchos, sin embargo, presentan limitaciones respecto a grandes cantidades de datos [17]. Un ejemplo de base de datos relacional es PostgreSQL.

Una base de datos no relacional es un sistema de almacenamiento que permite representar datos complejos en una estructura flexible, por ejemplo, clave-valor o

como documento [17]. Un ejemplo de base de datos no relacional es MongoDB. Además, InfluxDB para aplicaciones IoT [18].

CAPÍTULO 2

2. METODOLOGÍA

En el siguiente capítulo se detalla la propuesta de la solución en la cual se inició con el análisis de los requerimientos propuestos por el cliente. A través de estos se diseñó una arquitectura que incluye diferentes módulos necesarios a desarrollarse para cumplir con las expectativas del cliente. Por último, se describen de forma detallada los módulos que incluye la solución a través de la arquitectura de esta.

2.1 Análisis de Requerimientos

A través de varias reuniones con el cliente se pudo tomar los requerimientos necesarios para cumplir con sus necesidades y resolver el problema planteado en el capítulo 1. Se encontró que era necesaria la creación de una plataforma web en la que el cliente pueda visualizar los datos del funcionamiento en tiempo real de las maquinarias, así como el diseño de un dashboard amigable y fácil de usar. También se necesita configurar los dispositivos desde la plataforma web y que la plataforma permita diferentes protocolos de comunicación, entre los que se mencionaron MQTT, HTTP y AMQP para establecer la comunicación. A su vez, se comentó que sería de gran utilidad que dentro del sistema se pueda manejar diferentes roles por usuario para manejar de forma más eficaz su equipo de trabajo. Por último, es necesario mantener la información tomada de los equipos guardada para generar reportes que ayudarán a su respectivo análisis.

2.1.1 Definición de usuarios

Después del análisis de la solución a implementar, se encontraron 3 tipos de usuarios principales que usarían la plataforma: Administrador, el cual estará encargado de registrar la información de la fábrica, así como la gestión del equipo de trabajo. También podrá generar reportes del funcionamiento y monitorear las maquinarias; Supervisor de Planta, el cual podrá monitorear el funcionamiento de los equipos, así como modificar el dashboard según sus necesidades, además, podrá generar reportes del funcionamiento de las maquinarias; Administrador de Equipos, será el encargado de la comunicación y gestión de los diferentes equipos para que se puedan comunicar exitosamente con la plataforma.

2.2 Requerimientos y funcionalidades

Luego del análisis de los tipos de usuarios y las necesidades del cliente, se procede con la definición de los requerimientos necesarios relacionados a cada usuario. A través de esto, se pueden generar el diagrama de casos de uso y resumir las tareas asociadas a cada tipo de usuario según las necesidades del cliente.

2.2.1 Casos de Uso

A través de la definición de los usuarios se procede con la realización del diagrama de casos de uso (ver Fig. 2.1) en el cual se pueden definir las diferentes funcionalidades asociadas a cada tipo de usuario.

Las funcionalidades asignadas a cada tipo de usuario son las siguientes:

- El usuario Administrador de Equipos tiene la posibilidad de ver todos los dispositivos que se encuentren configurados en la fábrica. Además, puede agregar nuevas maquinarias a la plataforma para monitorear su información. A su vez, tiene la capacidad de elegir el protocolo de comunicación a usar y gestionar los diferentes dispositivos asociados a una fábrica que incluye la eliminación y edición de las configuraciones de los equipos.
- El usuario Administrador de Fábrica será quien cree la fábrica para que los dispositivos sean agregados a esa cuenta. Para el manejo de su equipo tendrá la posibilidad de invitar a todo su equipo de trabajo con roles definidos dentro del sistema. Este tipo de usuario también puede visualizar los dashboards de monitoreo de equipos y generar reportes.
- El usuario Supervisor de Planta puede visualizar la información del monitoreo de los equipos, configurar el dashboard para modificar los gráficos asociados al funcionamiento de cada maquinaria o agregar nuevos esquemas al dashboard. Por último, este tipo de usuario tiene la capacidad de generar reportes acerca del funcionamiento de la fábrica en los cuáles podrá elegir qué máquinas y el rango de fechas a mostrar en el reporte.

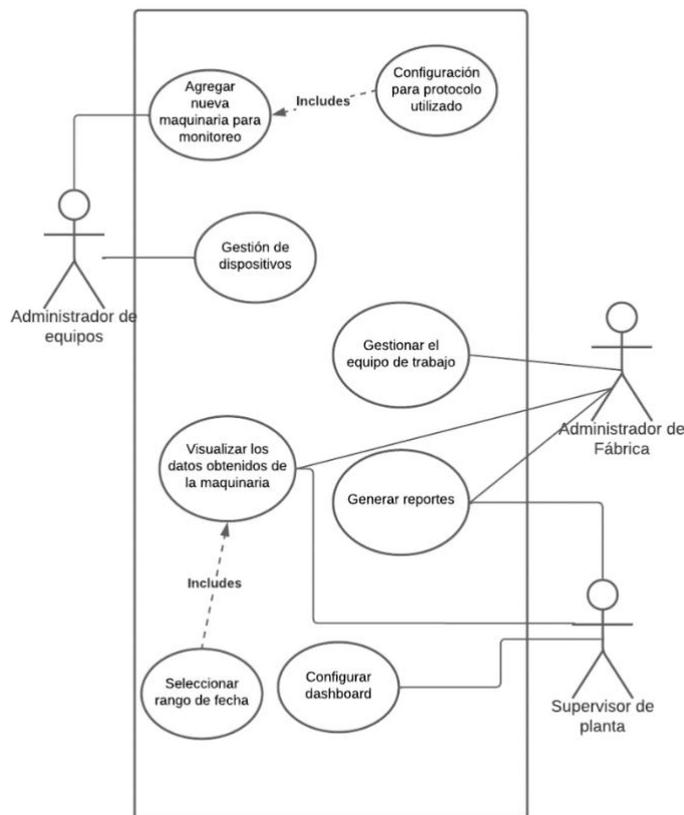


Figura 2.1 Diagrama de Casos de Uso

2.2.2 Requerimientos funcionales

El sistema contará con un conjunto de requerimientos funcionales los cuales conformarán las pautas a seguir para el diseño de la solución. Estos requerimientos son los siguientes:

Requerimientos funcionales:

- i. El Sistema debe permitir el registro de negocios y usuarios.
- ii. El sistema debe permitir la conexión de maquinarias a través de los protocolos de comunicación MQTT, HTTP y AMQP.
- iii. El sistema debe permitir la integración de aplicaciones de terceros mediante un API REST.
- iv. El sistema debe almacenar toda la información acerca del funcionamiento de las maquinarias.
- v. El sistema debe permitir la diferenciación de las acciones a realizar dependiendo del rol del usuario registrado.
- vi. El usuario con rol Administrador de Equipos debe ser capaz de integrar maquinarias con el Sistema.
- vii. El usuario con rol Supervisor de Planta debe ser capaz de editar, eliminar y agregar gráficos a las plantillas para la visualización de datos.

- viii. El usuario con rol Supervisor de Planta debe ser capaz de gestionar en el dashboard un máximo de 5 plantillas.
- ix. El usuario con rol Supervisor de Planta deber ser capaz de visualizar los datos de las maquinarias en tiempo real.
- x. El usuario con rol Administrador debe ser capaz de realizar las mismas acciones asignadas al rol Supervisor de Planta.
- xi. El usuario con rol Administrador debe ser capaz de permitir la invitación de nuevos usuarios con roles definidos en la aplicación.

2.3 Diseño de la solución

Para el diseño de la solución se construyó un diagrama de bloque (ver Fig. 2.2) en el cual se definen los componentes principales que constituirán el sistema dentro de los cuales se pueden mencionar:

- Adaptadores: Será el componente que permita a conectar los dispositivos en las fábricas con algunos protocolos de comunicación. Por ejemplo, para la comunicación a través de MQTT y AMQP se usará RabbitMQ.
- Middleware: Será la capa que transformará la información que los adaptadores reciben en un formato ya definido por el sistema para una correcta comunicación.
- Servicios: Lo complementan las diferentes aplicaciones definidas como servicios core que se comunicarán con la base de datos, con la plataforma web y las aplicaciones de terceros que se conecten al sistema.
- Base de datos: Será el componente que almacenará toda la información perteneciente a los usuarios y el monitoreo de las maquinarias.

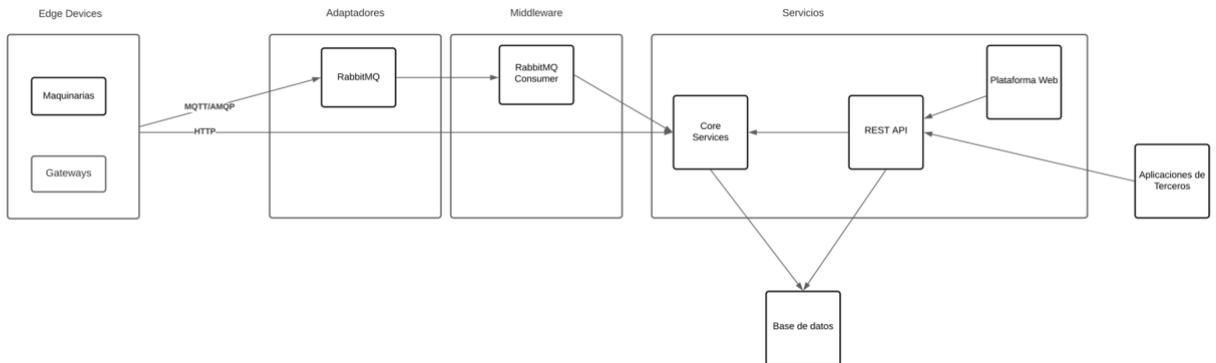


Figura 2.2 Diagrama de Bloque

2.3.1 Arquitectura

A partir de este diseño inicial del sistema, se podrá crear la arquitectura (ver Fig. 2.5). Esta contendrá los mismos componentes del diagrama de bloque, pero con una mejor definición acerca de qué tipo de servidores usar y se ampliarán las partes del componente de base de Datos.

A continuación, se describirá cada componente inicialmente mencionado en la ilustración (ver Fig. 2.2) pero en un nivel más detallado acerca de las tecnologías a ser usadas.

Adaptadores:

Como ya se mencionó, ayudará a comunicar los dispositivos con la plataforma. Dentro de los requerimientos funcionales especificados, la plataforma será compatible con los protocolos de comunicación MQTT y AMQP, en base a esto usaremos el broker de mensajería llamado RabbitMQ el cuál será instalado en un servidor para que este sea capaz de recibir los mensajes y almacenarlo en una cola de mensajes para su posterior procesamiento.

Middleware:

Este componente será el encargado de transformar la información almacenada en los adaptadores, esta información vendrá de diferentes protocolos, por lo que dentro de los servicios en esta capa se detectará el formato y se realizará el respectivo análisis. Tal como muestra la arquitectura, se contará con un servicio que actuará como un consumidor

de RabbitMQ. Este servicio tomará los mensajes de la capa de adaptadores según el adaptador configurado y transformará la información para ser enviada en un requerimiento POST hacia la capa de Servicios.

Servicios:

Este es el componente más importante de la arquitectura debido a que contiene las aplicaciones que conforman el núcleo del sistema. Dentro de este se podrá apreciar la aplicación Core Services la cual tendrá la responsabilidad de tomar los requerimientos enviados por los Middleware o directamente desde los dispositivos en el caso de que el protocolo de comunicación sea HTTP. A su vez, Core Services será quién modifique la información almacenada en la base de datos, tanto la información asociada a la gestión de usuarios como la configuración y monitorización de los equipos. También se contará con un REST API que tendrá la capacidad leer la información almacenada en el componente de Base de Datos para que tome los datos de usuarios y monitorización de equipos. Esta aplicación no podrá modificar la información de la base de datos, para esto tendrá que comunicarse con Core Services para la modificación de datos de usuarios y demás información que tenga que actualizarse. El principal objetivo del REST API es la comunicación con la plataforma web y las aplicaciones de terceros, para esto los usuarios tendrán a su disposición la documentación para la integración con el aplicativo. Por último, se contará con plataforma web la cuál será el medio por el cual los diferentes tipos de usuarios realicen las acciones ya definidas en el Análisis de Requerimientos. Este servicio contará con la capacidad de comunicación con el REST API para así obtener toda la información necesaria en la web.

Almacenamiento:

Para el almacenamiento de los datos se contará con una base de datos relacional y una no relacional. La base de datos relacional mantendrá la información de los usuarios, fábricas, equipos a monitorear y plantillas. Esto debido a que permite relacionar las diferentes entidades del sistema

de forma más eficiente. Para esto se contará una instancia de PostgreSQL instalada en un servicio de AWS llamado RDS. En la Fig. 2.3 se muestra el diagrama entidad relación para la base de datos relacional que contará con las diferentes entidades necesarias para manejar la relación de los usuarios, plantillas y dispositivos.

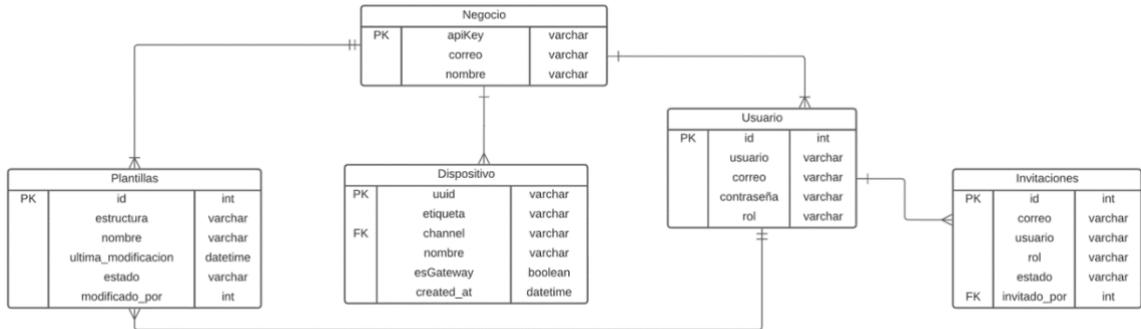


Figura 2.3 Diagrama Entidad Relación

La base de datos no relacional mantendrá la información del funcionamiento de los equipos debido a que este tipo de bases de datos proporcionan una mayor rapidez en la lectura y escritura de datos comparado con las RDB. En este apartado se utilizará InfluxDB, una base de datos de series temporales de código abierto. Este servicio será instalado en una instancia de AWS EC2. Esta base de datos contará con una colección llamada medición que tendrá los datos de la medición como el valor, unidad de medida, fecha de medición y a cuál dispositivo pertenece dicha medición (ver Fig. 2.4).

Medición	
uuid	objectId
nombreValor	string
valor	string/float
unidadMedida	string
fecha	datetime
canal	string
numero	boolean

Figura 2.4 Colección de Datos

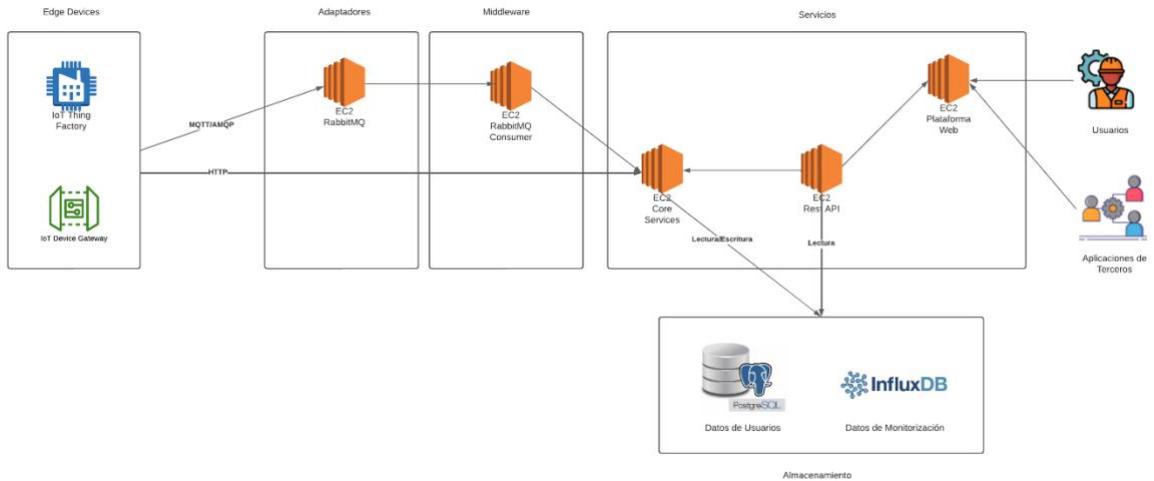


Figura 2.5 Arquitectura del Sistema

Adicional se ha agregado el diagrama de despliegue, que servirá para saber que elementos de software se implementará en qué nodos de hardware. También se pueden representar las relaciones entre nodos, en este caso se muestra el protocolo de comunicación y el puerto mediante los cuales se establecerá la comunicación (ver Fig. 2.6).

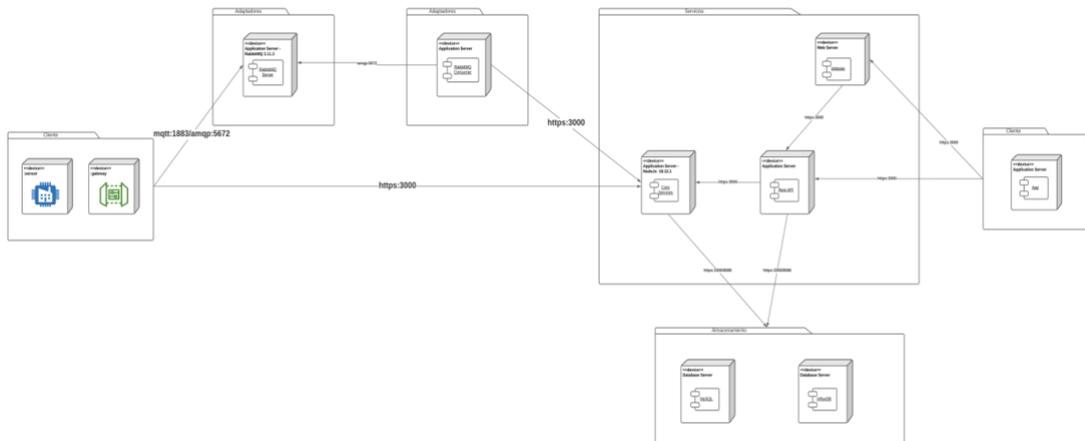


Figura 2.6 Diagrama de Despliegue

Debido a que la naturaleza del proyecto es resolver un problema en base a las necesidades específicas de los usuarios, es importante incluir a dichos usuarios en todo el proceso de diseño y desarrollo de la solución, para ello se utilizó una metodología basada en Design Thinking, iniciando con la etapa de Empatizar, en

la que se buscó entender a los usuarios, es decir, entender cómo piensan y cuáles son sus necesidades; la segunda etapa fue Definir, en la que se sintetizó la información y se identificaron los puntos de dolor de los usuarios; la tercera etapa fue Idear, en la cual a partir de los puntos de dolor identificados en la etapa anterior se propusieron ideas que solucionan los problemas identificados; la cuarta etapa fue prototipar, en la cual se diseñaron soluciones que mejoren la experiencia de los usuarios al utilizar la plataforma; la última etapa fue evaluar, en la cual se obtuvo retroalimentación por parte de los usuarios respecto a la solución propuesta y en base a dicha retroalimentación se realizaron ajustes a la solución, por ello, las dos últimas etapas se realizaron de forma cíclica.

El proceso de Design Thinking fue utilizado únicamente para el dashboard, pues el enfoque de este proyecto es únicamente crear una plataforma IoT que brinde a los usuarios soluciones a los problemas que han presentado anteriormente sin tener que preocuparse por el aspecto técnico de la plataforma.

Empatizar y Definir

Para esta etapa se realizaron entrevistas a usuarios que cumplen alguno de los perfiles mencionados, el objetivo de la entrevista era obtener información sobre la experiencia que han tenido los usuarios al utilizar plataformas IoT en la industria y utilizar dicha información para conocer cuáles son las principales dificultades que presentan los usuarios al trabajar con plataformas IoT.

De acuerdo con lo manifestado por los usuarios, los problemas residen principalmente en la necesidad de obtener los datos del funcionamiento de la maquinaria pues consideran que esto ayuda a realizar mantenimiento preventivo en lugar de mantenimiento correctivo. Uno de los principales problemas es adaptarse a la solución ya existente en la industria, pues integrar nuevos sensores implica encontrar puertos libres y válidos, además que no permitían la conexión de sensores que utilizaran otro protocolo de comunicación.

También es importante que los usuarios puedan acceder a información del funcionamiento en un formato que sea fácil de entender pues

normalmente se presenta en forma de bitácoras donde los datos se muestran de la misma forma que se obtienen de los sensores.

En el caso de utilizar gráficos para visualizar los datos un aspecto importante es permitir a los usuarios personalizar la plataforma de forma que se sientan cómodos con los gráficos utilizados y el diseño de la plataforma, por ejemplo, los colores a utilizar y la posición de los gráficos. Además, los gráficos deberían permitir la visualización de datos de periodos de tiempo específico para no tener que revisar las bitácoras textuales.

Idear

En base a las necesidades identificadas en las entrevistas de los usuarios se propusieron las siguientes características para la creación del dashboard:

- Para la personalización del dashboard se tendrá acceso a una pestaña de plantillas en la que los usuarios podrán guardar la distribución de gráficos del dashboard de forma que puedan cambiar de una a otra con solo un clic. Además, los usuarios podrán arrastrar los gráficos a la posición deseada.
- Cada gráfico tendrá la opción de seleccionar el rango de fecha
- Actualizaciones en tiempo real de los datos recolectados de los sensores

Prototipar

El proceso de prototipado se realizó en conjunto con el cliente. El prototipo se diseñó con el objetivo de que los usuarios puedan observar e interactuar con la interfaz de la solución antes de empezar a desarrollarla. En el prototipo se plasmaron todos los requerimientos identificados en las etapas de empatizar, definir e idear.

En base a la retroalimentación obtenida en la etapa de evaluación se realizaron los cambios respectivos al prototipo con el fin de que los usuarios se sientan cómodos con la solución. A continuación, se presentan las pantallas diseñadas para el prototipo

- Login/Registro: Una pantalla sencilla en la que los usuarios pueden iniciar sesión para ingresar a la plataforma, únicamente usuarios registrados tienen acceso y solo a aquellas funcionalidades específicas de su rol en la empresa. La pantalla de registro es únicamente para empresas

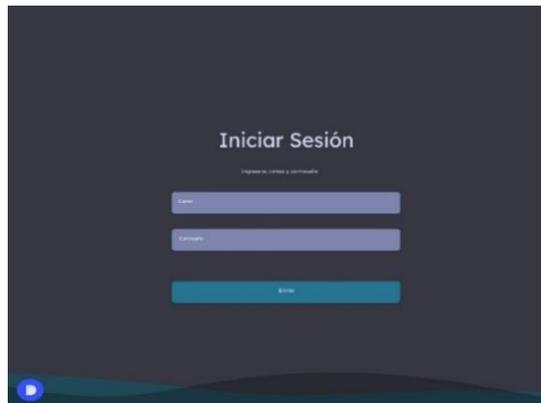


Figura 2.7 Pantalla de Inicio de Sesión

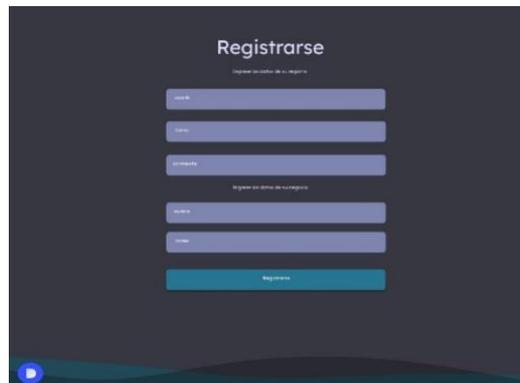


Figura 2.8 Pantalla de Registro

- Pantalla principal: En la pantalla principal de la aplicación los usuarios encuentran los gráficos creados a partir de los datos obtenidos de la maquinaria, los usuarios tienen la opción de crear gráficos a los cuáles pueden asignarles un nombre, el dispositivo del cual se obtienen los datos y el tipo de gráfico. Los usuarios tienen la opción de mover los gráficos a la posición que deseen, además de eliminar gráficos que no desean tener en la pantalla. Todos los gráficos se actualizan en tiempo real.

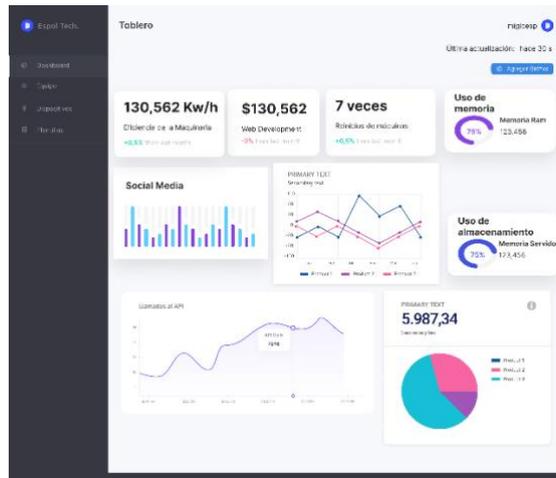


Figura 2.9 Pantalla Principal

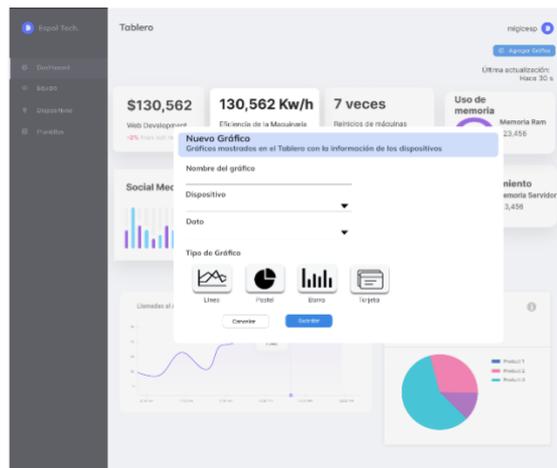


Figura 2.10 Agregar nuevo Gráfico

- Equipo: Esta pantalla es únicamente accesible para el administrador a través de la cual puede invitar nuevos usuarios a la aplicación con un rol específico, además de conocer quienes conforman su equipo actualmente.

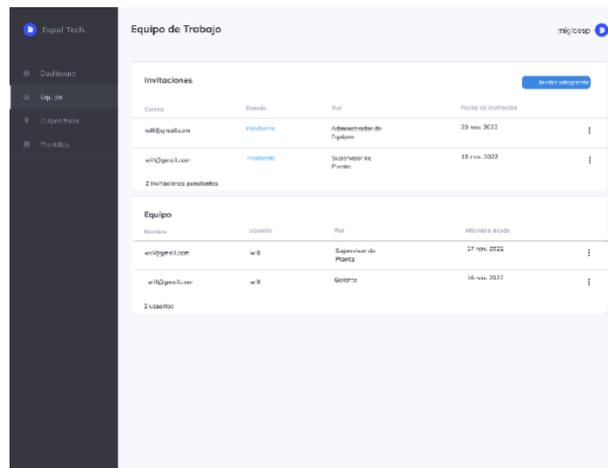


Figura 2.11 Pantalla de Equipo

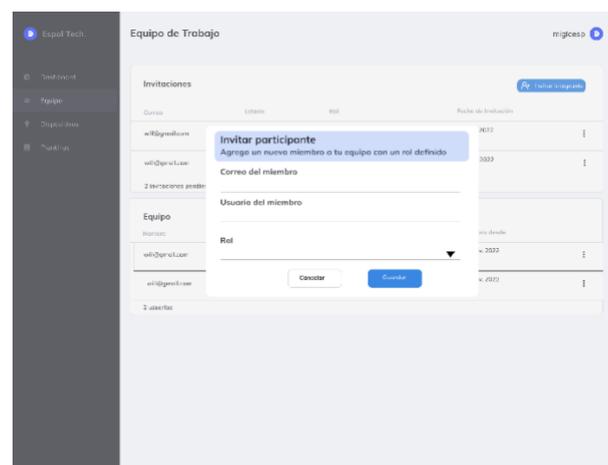


Figura 2.12 Invitar participante al Equipo

- **Dispositivos:** Esta pantalla permite a los usuarios añadir nuevos dispositivos a través de los cuales se obtendrán los datos que podrán ser mostrados en los gráficos de la pantalla principal. Agregar un nuevo dispositivo implica proporcionarle un nombre para identificar a futuro y el protocolo de comunicación a utilizar.

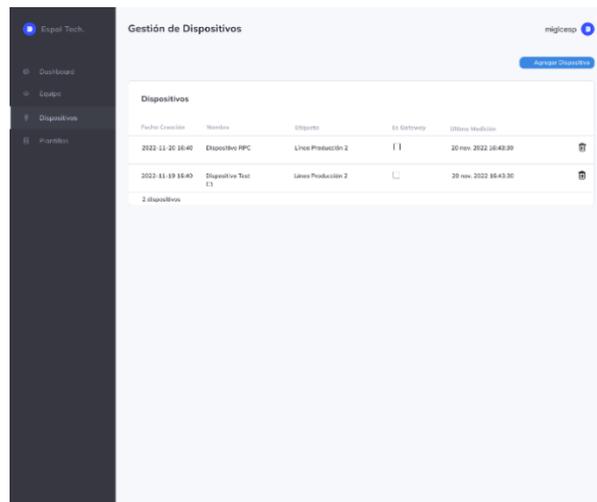


Figura 2.13 Pantalla de Dispositivos

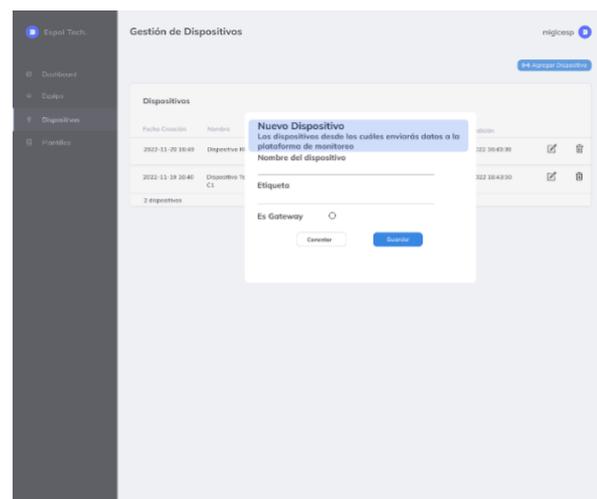


Figura 2.14 Agregar nuevo Dispositivo

- Plantillas: Con el fin de brindar al usuario la capacidad de personalizar el dashboard los usuarios tienen la opción de crear plantillas en las que pueden agregar y eliminar gráficos, estas plantillas permanecen almacenadas de forma que el usuario puede alternar entre plantillas de forma rápida.

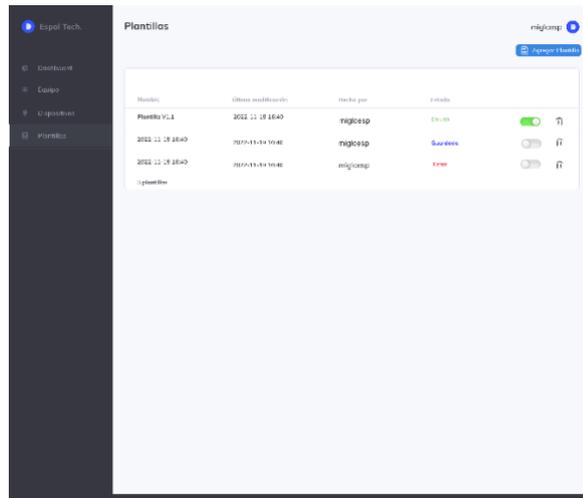


Figura 2.15 Pantalla de Plantillas

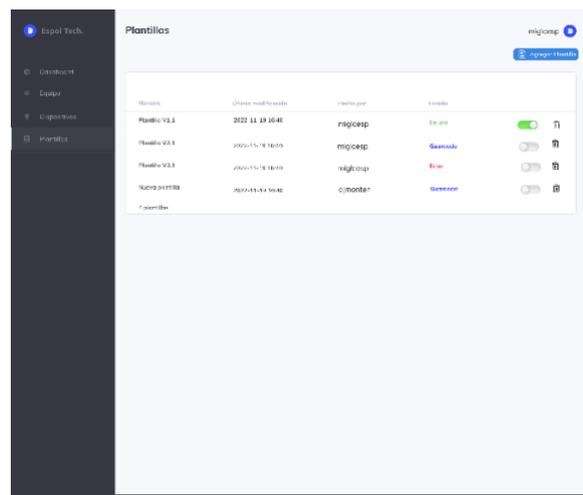


Figura 2.16 Agregar nueva Plantilla

Evaluar

Se realizó una reunión con el cliente y se obtuvo la siguiente retroalimentación:

- Mostrar el usuario que ha iniciado sesión en el dashboard.
- Definir la configuración del protocolo de comunicación que se usará para la conexión con un dispositivo.
- Conocer quien ha realizado el último cambio en una plantilla.
- Utilizar nombres de usuario para los miembros del equipo.

En base a estos comentarios se realizaron los cambios respectivos en el prototipo hasta llegar al mostrado en la sección anterior. El objetivo de evaluar el prototipo con los usuarios es obtener retroalimentación temprana, de forma que se puedan realizar las correcciones necesarias antes de desarrollar la solución final, sin embargo, es importante que la solución desarrollada también pase por una evaluación por parte de los usuarios, de esta forma la solución será la más adecuada para los usuarios.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

En el presente capítulo se describe el producto final y los resultados obtenidos en las pruebas de usuarios y evaluación del sistema. También, se ofrece un análisis de costos para que el producto final se desempeñe de forma óptima en el ambiente de producción. Por último, se analiza la factibilidad de despliegue del proyecto bajo distintas condiciones de estrés para el sistema.

3.1 Producto final

La metodología de Design Thinking ayudó a realizar distintas evaluaciones del prototipo para que la solución diseñada se ajuste a las necesidades del cliente. Esto conllevó a que el producto final requiera pocos cambios en relación al prototipo propuesto. A continuación, se detallan las principales características que mejor describen al producto final y las necesidades del cliente:

3.1.1 Tablero

La pantalla de tablero con la que se encuentran los usuarios apenas se crea una cuenta les muestra un mensaje para que creen o activen una plantilla, en este caso el botón “Agregar gráfico” se encuentra oculto.

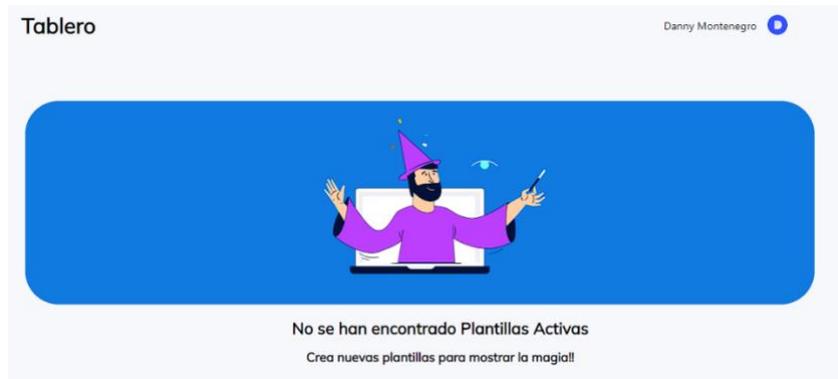


Figura 3.1 Tablero de usuario sin Plantillas activas

Una vez exista una plantilla la ventana de tablero mostrará el botón de “Agregar gráfico”

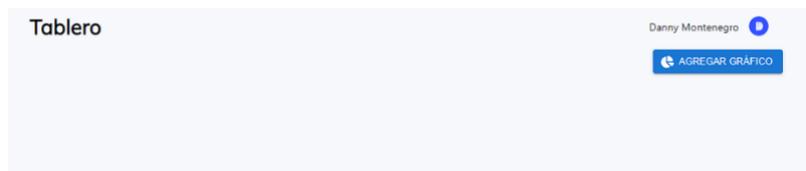


Figura 3.2 Tablero de usuario sin Gráficos

El botón abrirá un formulario en el que el usuario colocará los datos del gráfico como nombre, dispositivo o el tipo de gráfico.

Nuevo Gráfico
Gráficos mostrados en el Tablero con la información de los dispositivos

Nombre del gráfico

Dispositivo

Dato

1 Día 7 Días Último mes

Tipo de gráfico

Linea Pastel Barra Tarjeta

Figura 3.3 Formulario para agregar nuevo Gráfico

Finalmente, se agregará el tipo de gráfico seleccionado y se obtendrán los datos asociados al dispositivo seleccionado. El usuario puede agregar, eliminar o mover los gráficos, la única restricción es que los gráficos deben tener nombres diferentes. Un ejemplo de agregar varios gráficos se muestra en la Fig 3.4.



Figura 3.4 Tablero con Gráficos agregados

3.1.2 Manejo de los dispositivos

Desde esta página el usuario con rol de Administrador o Administrador de Equipos será capaz de crear, eliminar y visualizar los dispositivos asociados a la fábrica de la que forma parte.

The 'Dispositivos' page shows a table of devices with the following data:

Fecha Creación	Nombre	Etiqueta	Es Gateway	Última medición
2023-01-17T05:23:07.459Z	Máquina de inyección	Linea Producción 1	<input type="checkbox"/>	Aún sin mediciones
2023-01-17T05:23:47.457Z	Máquina CNC	Linea Producción 1	<input type="checkbox"/>	Aún sin mediciones
2023-01-17T05:24:02.602Z	Extrusora	Linea Producción 1	<input type="checkbox"/>	Aún sin mediciones
2023-01-17T05:24:24.492Z	Gateway CISCO	Linea Producción 2	<input checked="" type="checkbox"/>	Aún sin mediciones
2023-01-17T05:24:42.240Z	Gateway Huawei	Linea Producción 3	<input checked="" type="checkbox"/>	Aún sin mediciones

5 dispositivos

Figura 3.5 Visualización de dispositivos

Nuevo Dispositivo

Los dispositivos desde los cuáles enviarás datos a la plataforma de monitoreo.

Nombre del Dispositivo

Etiqueta del Dispositivo

Es Gateway

CANCELAR
CREAR

Figura 3.6 Creación de Nueva Plantilla

3.1.3 Manejo del Equipo de Trabajo

Desde esta página el usuario con rol de Administrador será capaz de gestionar su equipo de trabajo a través de invitaciones a nuevos miembros. Estas invitaciones tienen que ser aceptadas registrándose en la aplicación para que luego el usuario pueda ingresar a la plataforma y realizar las actividades que su rol lo permita.

ESPOL CIA, LTDA

- Dashboard
- Equipo
- Dispositivos
- Plantillas

Equipo

migloesp2

Invitaciones INVITAR INTEGRANTE

Correo	Estado	Rol	Fecha Invitación
angelmigue.cespedes@gmail.com	Pendiente	2	2023-01-16T17:39:22.414Z

1 invitación pendiente

Equipo

Email	Usuario	Rol	Miembro desde
migloesp@gmail.com	migloesp2	1	2022-12-19T03:12:28.587Z
launavil@espol.edu.ec	1234	1	2023-01-09T06:07:02.700Z

2 usuarios

Figura 3.7 Pantalla de Equipo

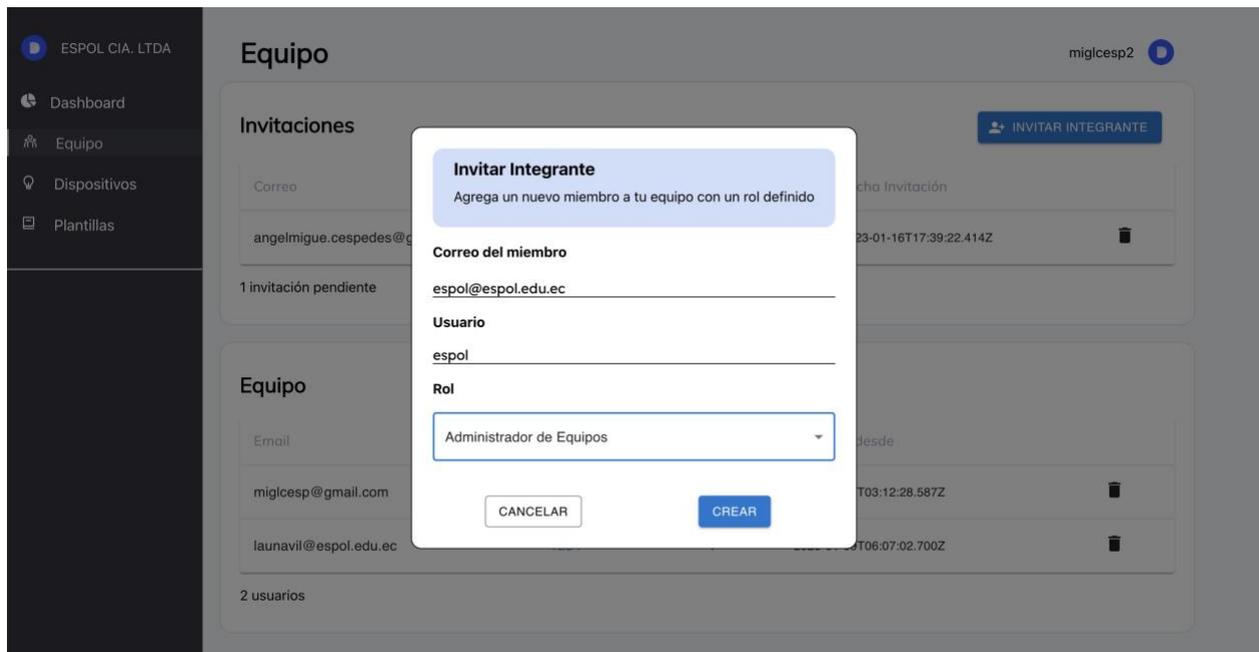


Figura 3.8 Invitación de un nuevo Participante

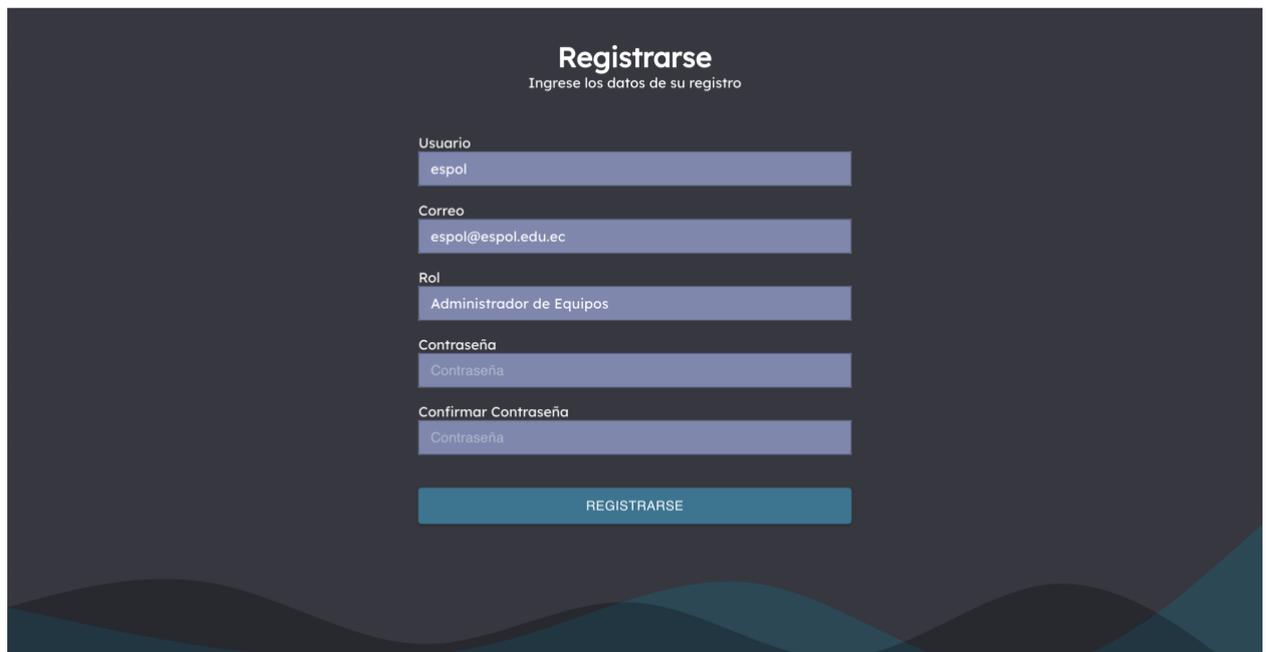


Figura 3.9 Registro de nuevo Participante del Equipo

3.1.4 Manejo de Plantillas

El manejo de plantillas permite al usuario con rol de Administrador o Supervisor gestionar las plantillas que se tienen creadas. Así como activarlas para ser mostrada la estructura en el panel principal de la aplicación.

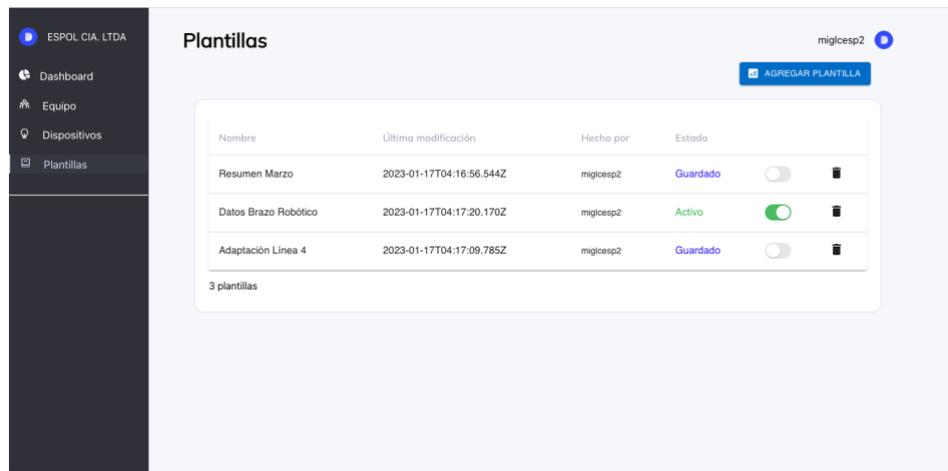


Figura 3.10 Pantalla de Administración de Plantillas

3.1.5 Protocolos de comunicación

Tal como se mencionó en los requerimientos del sistema el aplicativo debe ser capaz de establecer la comunicación del dispositivo con la plataforma a través de los protocolos de comunicación MQTT, HTTP y AMQP. Tal como se comentó en la sección de Metodología esto se logró a partir de RabbitMQ y un consumidor de las colas de mensajes que toma las mediciones, convierte la información en el formato del API Rest requerido y lo envía al servicio del EC2 RestAPI. En las figuras 3.11 y 3.12 se muestran los datos ingresados a partir de una simulación del ingreso de los datos.

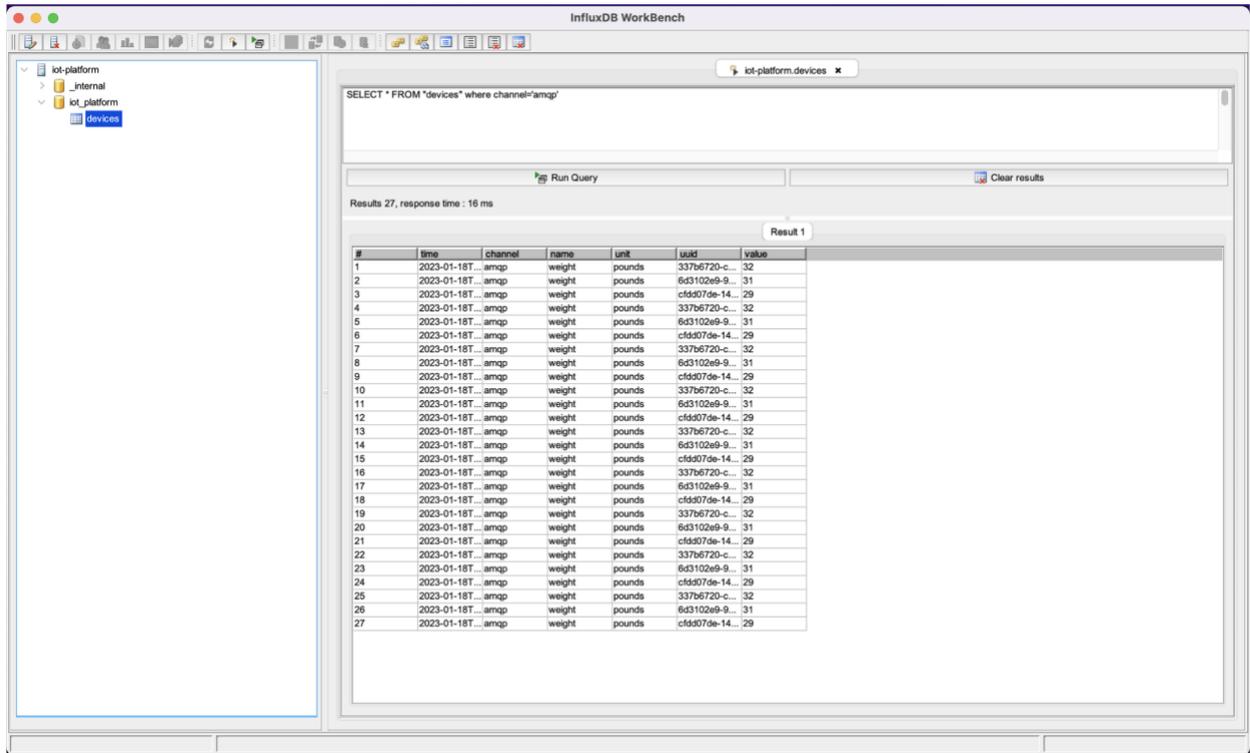


Figura 3.11 Mediciones a través del Protocolo AMQP

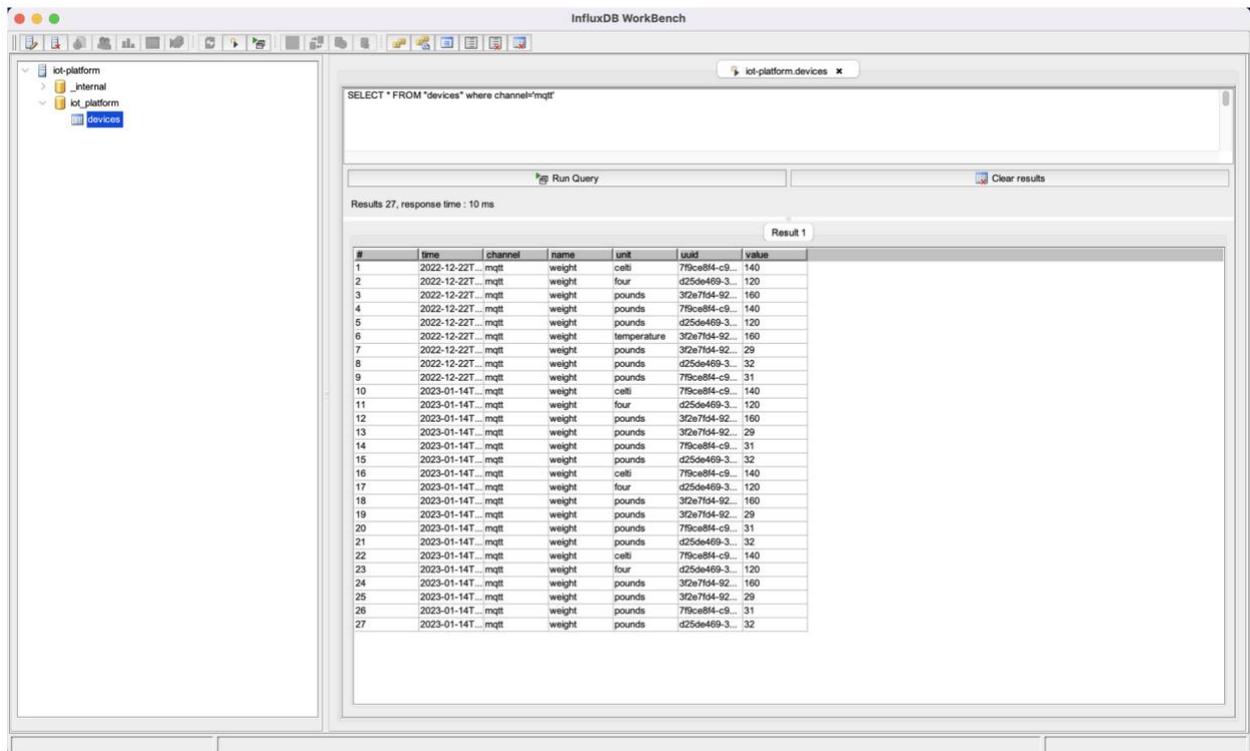


Figura 3.12 Mediciones a través del Protocolo MQTT

3.2 Evaluación

Para evaluar la satisfacción del cliente y posibles usuarios sobre el producto final se preparó un formulario de satisfacción que fue tomado luego de que los mismos utilizaran la plataforma. Este formulario recoge el nivel de satisfacción de los usuarios con respecto a las funcionalidades principales del sistema. En la tabla 3.1 se muestran las preguntas del formulario aplicado y el motivo de la realización de cada pregunta.

Tabla 3.1 Formulario para la Evaluación de la Plataforma

Pregunta	Motivo
¿Cuán fácil ha sido para usted personalizar su dashboard en la plataforma? Califique del 1 al 5 la facilidad donde 5 es muy fácil y 1 muy difícil.	Esta pregunta ayuda a entender si la plataforma IoT está cumpliendo con las expectativas del usuario en cuanto a la facilidad de personalización de su dashboard, lo que puede indicar la necesidad de mejoras en la interfaz de usuario o en la documentación de ayuda.
¿Cuán fácil ha sido para usted monitorear y analizar los datos de los dispositivos a través del dashboard? Califique del 1 al 5 la facilidad donde 5 es muy fácil y 1 muy difícil.	Esta pregunta ayuda a entender si la plataforma IoT está cumpliendo con las expectativas del usuario en cuanto a la facilidad de monitoreo y análisis de los datos de los dispositivos.
¿Cuán fácil ha sido para usted configurar nuevos dispositivos IoT en la plataforma? Califique del 1 al 5 la facilidad donde 5 es muy fácil y 1 muy difícil.	Esta pregunta ayuda a entender si el flujo de trabajo para la configuración de un nuevo dispositivo es el más sencillo posible para configurar el mismo y que se puedan guardar los datos en la plataforma.
Como usuario final, califique del 1 al 5 la gestión de distintas plantillas en la plataforma.	Esta pregunta ayuda a entender si la gestión de distintas plantillas del dashboard ha sido eficaz para mostrar diferentes estructuras de dashboards guardados desde la plataforma.
¿Cuán fácil ha sido para usted gestionar su equipo de trabajo, desde la invitación de un nuevo miembro hasta la gestión de este? Califique del 1 al 5 la facilidad donde 5 es muy fácil y 1 muy difícil.	Esta pregunta ayuda a entender si la conformación del equipo de trabajo a través de invitaciones por correo electrónico ha sido eficaz para agregar nuevos usuarios con roles diferentes capaces de trabajar bajo los mismos datos de los equipos asociados a la empresa.
¿Qué aspectos de la plataforma mejoraría?	Esta pregunta se ha hecho con la intención de que los usuarios finales nos ayuden con las

	mejoras que se le pueden aplicar a la plataforma.
--	---

Luego de la conformación del formulario, el mismo fue tomado a 10 usuarios diferentes que colaboraron en la evaluación final del sistema. Dichos usuarios también participaron en la toma de requerimientos al inicio del proyecto y han estado al tanto del mismo para la posible implementación en sus centros laborales. Tras aplicar el formulario a los usuarios, se obtuvieron los resultados mostrados a continuación:

Pregunta 1:

¿Cuán fácil ha sido para usted personalizar su dashboard en la plataforma? Califique del 1 al 5 la facilidad donde 5 es muy fácil y 1 muy difícil.

10 respuestas

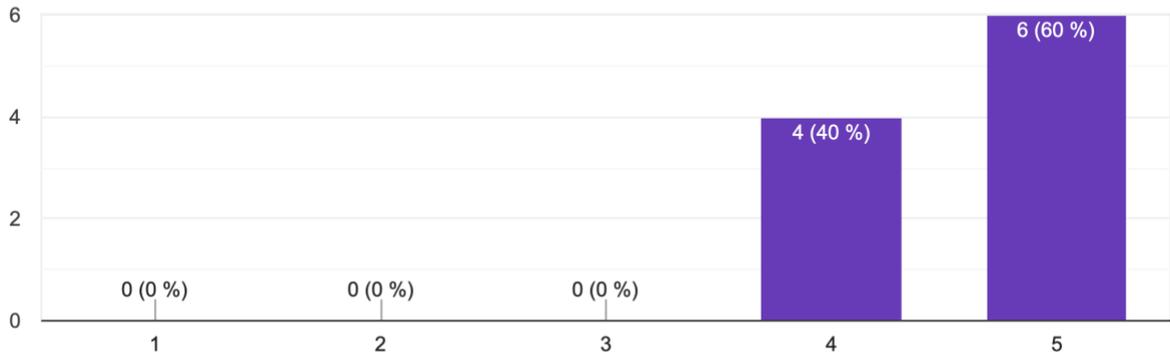


Figura 3.13 Resultados de la Evaluación Pregunta 1

Pregunta 2:

¿Cuán fácil ha sido para usted monitorear y analizar los datos de los dispositivos a través del dashboard? Califique del 1 al 5 la facilidad donde 5 es muy fácil y 1 muy difícil.

10 respuestas

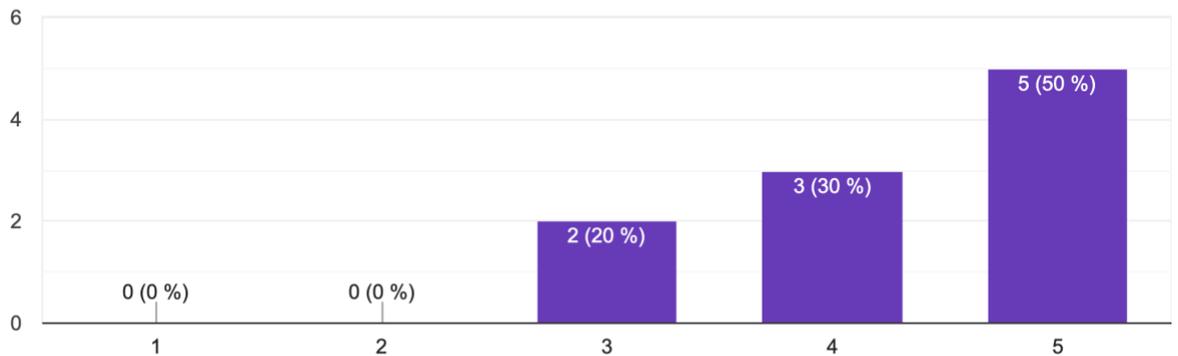


Figura 3.14 Resultados de la Evaluación Pregunta 2

Pregunta 3:

¿Cuán fácil ha sido para usted configurar nuevos dispositivos IoT en la plataforma? Califique del 1 al 5 la facilidad donde 5 es muy fácil y 1 muy difícil.

10 respuestas

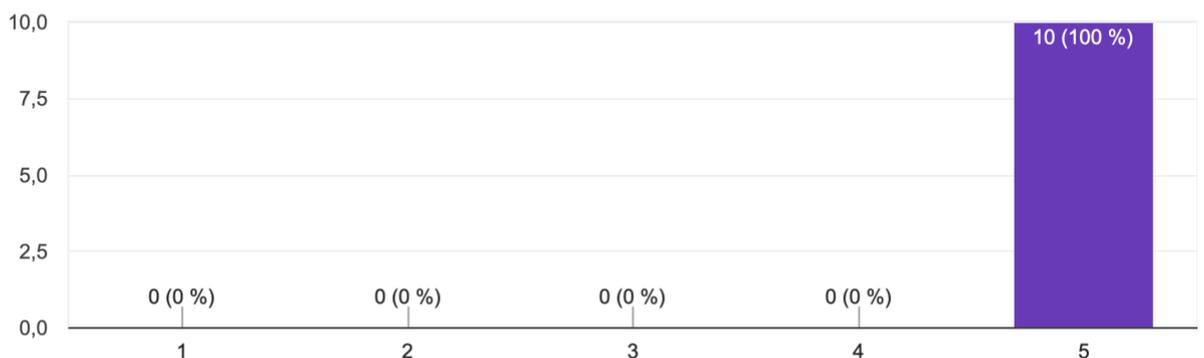


Figura 3.15 Resultados de la Evaluación Pregunta 3

Pregunta 4:

Como usuario final, califique del 1 al 5 la gestión de distintas plantillas en la plataforma.

10 respuestas

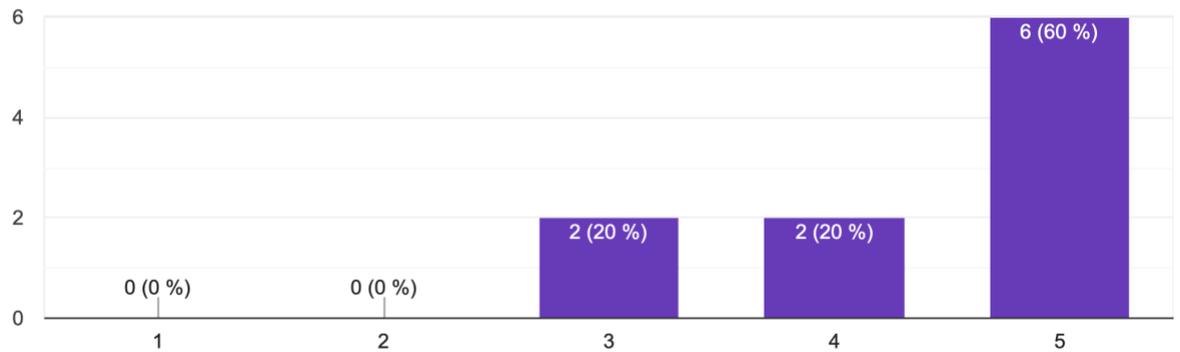


Figura 3.16 Resultados de la Evaluación Pregunta 4

Pregunta 5:

¿Cuán fácil ha sido para usted gestionar su equipo de trabajo, desde la invitación de un nuevo miembro hasta la gestión del mismo? Califique del 1...la facilidad donde 5 es muy fácil y 1 muy difícil.

10 respuestas

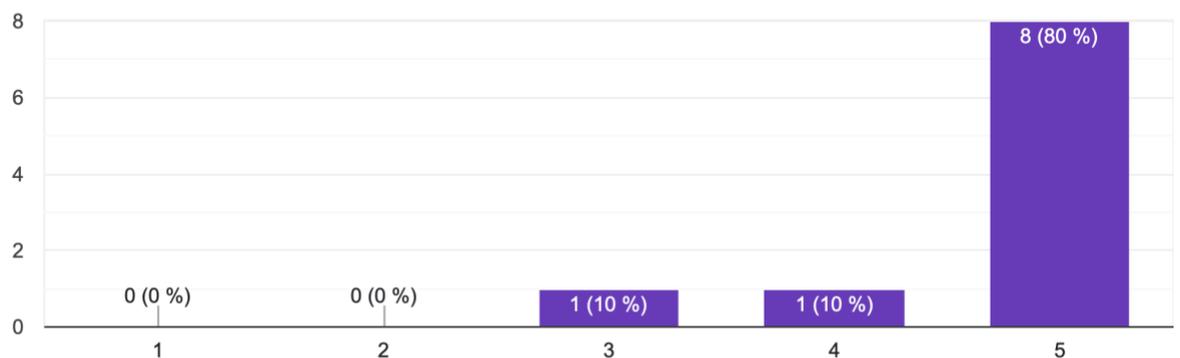


Figura 3.17 Resultados de la Evaluación Pregunta 5

Pregunta 6:

Esta pregunta es abierta debido a que se necesitaba que el usuario proponga mejoras a la plataforma. Luego de analizar estas mejoras se resumieron los siguientes puntos como los más importantes:

- Agregar más opciones de gráficos para mostrar en el dashboard
- Agregar más variables para distintos gráficos como el de línea y el de barras.
- Quisiera monitorear el estado de la invitación por correo, saber si el usuario ya recibió el correo o lo abrió.
- Crear agrupaciones de dispositivos por líneas de producción o fábricas.
- Poder realizar modificaciones a los gráficos sin tener que eliminarlos y crearlos nuevamente.

3.3 Análisis de los resultados

En base a los resultados obtenidos de la encuesta se muestra el análisis respectivo por cada pregunta. Para las preguntas 1,2,3,4 y 5 el análisis se realiza en base los puntajes seleccionados por los usuarios, mientras que para la pregunta 6 el análisis se basa en extraer la información más relevante de los comentarios realizados por los usuarios.

- Pregunta 1: Esta pregunta hace referencia a la facilidad en la personalización del dashboard. Se obtuvo que el 60% de los encuestados respondieron con la opción 5 y el 40% restante con la opción 4. Lo que implica que la modificación del dashboard para los usuarios es bastante sencilla.
- Pregunta 2: En esta pregunta se obtuvo que el 20% de los usuarios califican la monitorización y análisis de los datos con una facilidad media, por lo que se debe mejorar la calidad del dashboard.
- Pregunta 3: Según los usuarios encuestados el 100% calificó que el proceso de configuración de nuevos dispositivos es muy fácil, siendo un resultado exitoso debido a que era uno de los objetivos de la plataforma.
- Pregunta 4: La gestión de distintas plantillas obtuvo que un 20% de los usuarios indican que tienen algunas dificultades o insatisfacciones al

realizar esta tarea, mientras que un 60% calificó la gestión de las plantillas con un 5, lo que indica un alto nivel de facilidad de uso y satisfacción. En resumen, estos resultados sugieren que la gestión de las plantillas es una característica valorada por los usuarios, pero hay un pequeño grupo de usuarios que tienen sugerencias para mejorarla.

- Pregunta 5: Esta pregunta hace referencia a la facilidad en la gestión del equipo de trabajo, según los resultados el 80% de los encuestados califica con 5 a esta actividad. Por lo que una alta mayoría de los posibles usuarios califican esta actividad como fácil de usar dentro de la plataforma.
- Pregunta 6: En general, los resultados de esta pregunta indican que los usuarios tienen sugerencias para mejorar la plataforma en términos de visualización de datos y monitoreo. Varias respuestas sugieren que los usuarios desean más opciones de gráficos y más variables para mostrar en el dashboard. También se menciona la necesidad de monitorear el estado de las invitaciones por correo y la posibilidad de crear agrupaciones de dispositivos por líneas de producción o fábricas. Otros usuarios mencionan la necesidad de un tracker de las invitaciones y la posibilidad de crear usuarios con roles específicos y permisos. Finalmente, algunos usuarios mencionan que los datos de los gráficos son estáticos y sugieren una mayor flexibilidad en la personalización de los gráficos. En general, estos resultados indican que los usuarios desean una mayor personalización y flexibilidad en la visualización de datos, así como una mejora en el monitoreo de las invitaciones y agrupaciones de dispositivos.

3.4 Análisis de costos

En la presente sección se muestra una estimación de los recursos necesarios para la implementación del proyecto en un ambiente de producción. La solución propuesta cuenta con tecnologías de código abierto, sin embargo, se usó servicios ofrecidos por Amazon Web Services para tener máquinas virtuales capaces de ofrecen un buen rendimiento de nuestro aplicativo. Es válido recalcar, que los costos del sistema se basan principalmente en las horas de trabajo de cada uno de los servicios activos.

La realización del proyecto se ha estimado con un valor económico de \$3000 como pago inicial y un pago de \$50 de forma mensual para mantener la aplicación funcionando en el ambiente de producción y monitorear que la aplicación se encuentre funcionando tal como se espera. La siguiente tabla muestra el análisis de costos del diseño, desarrollo y despliegue del sistema.

Tabla 3.2 Costo de Diseño, Desarrollo y Despliegue de la Plataforma

Detalle	Cantidad de desarrolladores	Precio Unitario	Precio Final
Diseño de la aplicación	2	\$400	\$800
Desarrollo de la aplicación	2	\$1.075	\$2.150
Despliegue de la aplicación		\$50	\$50
Total	-	-	\$3.000

A continuación, se muestra una lista detallada con cada uno de los servicios necesarios y se han mostrado en la arquitectura del sistema para el buen funcionamiento del aplicativo (Fig. 2.5):

Tabla 3.3 Costo Operacional de la Plataforma

Nombre Servicio	Sistema Operativo	Número de instancias	vCPUs	Memoria	Precio Final (Mensual)
EC2 RabbitMQ	Linux	1	2	2GiB	\$8.83
EC2 RabbitMQ Consumer	Linux	1	2	2GiB	\$8.83
EC2 RestAPI	Linux	2	2	2GiB	\$17.66
EC2 Plataforma Web	Linux	1	2	2GiB	\$8.83
EC2	Linux	1	1	4GiB	\$20.66

Almacenamiento					
Total		-	-		\$63.81

Cabe recalcar que estos precios se sugieren para una cantidad limitada de usuarios y dispositivos asociados. En caso de que los servicios necesiten una mayor cantidad de procesamiento se puede configurar el auto escalamiento de estos, lo que permite que sea el mismo servicio la entidad que se dé cuenta de que necesite o no mayor poder de procesamiento.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

A partir de los resultados de la creación del sistema, formularios aplicados a usuarios finales y pruebas del funcionamiento del sistema se han obtenido las siguientes conclusiones y recomendaciones.

4.1 Conclusiones

- Mediante este proyecto se ha demostrado la factibilidad de crear una plataforma IoT multi protocolo y la utilidad que esta tiene al reducir la carga de configurar los dispositivos a los usuarios mediante la creación de intermediarios, en este caso el rest api y los consumidores de la cola de mensajes, de forma que los usuarios únicamente se encargan de realizar la configuración externa de los dispositivos mientras que la lectura y escritura se manejan de manera automática.
- La capacidad de mostrar gráficos usando los datos obtenidos de los dispositivos ayuda a los usuarios a tener un mejor entendimiento del funcionamiento de la maquinaria y tomar mejores decisiones en tiempo real.
- Con el rest api se demuestra cómo es posible contar con un único punto de entrada de datos de dispositivos que utilicen diferentes protocolos de comunicación, es este caso HTTP, MQTT y AMQP. Si bien fue necesario realizar configuraciones adicionales por cada protocolo esto no se compara con el trabajo que tomaría realizarlo de una forma tradicional donde se tiene un punto de entrada para cada protocolo. Asimismo, es posible agregar más protocolos de comunicación al rest api, realizando la configuración adecuada, de forma que se puedan conectar una mayor cantidad de dispositivos que utilicen diferentes protocolos de comunicación.

4.2 Recomendaciones

La plataforma IoT actual se ha mostrado eficaz en la conectividad de dispositivos y la recopilación de datos en tiempo real. Sin embargo, es importante considerar las implicaciones de seguridad y privacidad en el uso de esta tecnología. Estas recomendaciones serán útiles tanto para desarrolladores, fabricantes y

propietarios de dispositivos IoT para asegurar un uso seguro y eficaz de la plataforma:

- Cambio dinámico de rango de fechas para los gráficos, la plataforma únicamente cuenta con tres opciones en el rango de fechas para los gráficos: El último día, la última semana o el último mes. Una mejora a esta opción es permitir cambiar de forma dinámica entre los tres rangos después de crear el gráfico.
- Implementar un modo de pruebas de manera que los usuarios nuevos puedan interactuar con la plataforma sin temer a eliminar información importante.
- Ampliar el número de protocolos compatibles con la plataforma

BIBLIOGRAFÍA

- [1] L. D. Candia, A. S. Rodríguez, N. Castro, P. A. Bazán, V. M. Ambrosi, and F. J. Díaz, "Mejoras en maquinaria industrial con IoT: hacia la industria 4.0," *XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018)*., 2018, Accessed: Oct. 25, 2022. [Online]. Available: <http://sedici.unlp.edu.ar/handle/10915/73348>
- [2] L. Duan and L. da Xu, "Data Analytics in Industry 4.0: A Survey," *Information Systems Frontiers*, pp. 1–17, Aug. 2021, doi: 10.1007/S10796-021-10190-0/FIGURES/7.
- [3] M. Saez, F. P. Maturana, K. Barton, and D. M. Tilbury, "Real-Time Manufacturing Machine and System Performance Monitoring Using Internet of Things," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1735–1748, Oct. 2018, doi: 10.1109/TASE.2017.2784826.
- [4] W. Zhang, M. P. Jia, L. Zhu, and X. A. Yan, "Comprehensive Overview on Computational Intelligence Techniques for Machinery Condition Monitoring and Fault Diagnosis," *Chinese Journal of Mechanical Engineering (English Edition)*, vol. 30, no. 4, pp. 782–795, Jul. 2017, doi: 10.1007/S10033-017-0150-0/FIGURES/4.
- [5] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mech Syst Signal Process*, vol. 20, no. 7, pp. 1483–1510, Oct. 2006, doi: 10.1016/J.YMSSP.2005.09.012.
- [6] K. Shahzad and M. Onils, "Condition Monitoring in Industry 4.0-Design Challenges and Possibilities: A Case Study," *2018 Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2018 - Proceedings*, pp. 101–106, Aug. 2018, doi: 10.1109/METROI4.2018.8428306.

- [7] Z. Iqbal, K. Kim, and H. N. Lee, "A cooperative wireless sensor network for indoor industrial monitoring," *IEEE Trans Industr Inform*, vol. 13, no. 2, pp. 482–491, Apr. 2017, doi: 10.1109/TII.2016.2613504.
- [8] H. Barksdale, Q. Smith, and M. Khan, "Condition Monitoring of Electrical Machines with Internet of Things," *Conference Proceedings - IEEE SOUTHEASTCON*, vol. 2018-April, Oct. 2018, doi: 10.1109/SECON.2018.8478989.
- [9] T. Salman and R. Jain, "A Survey of Protocols and Standards for Internet of Things," *Advanced Computing and Communications*, Feb. 2019, doi: 10.48550/arxiv.1903.11549.
- [10] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings*, Oct. 2017, doi: 10.1109/SYSENG.2017.8088251.
- [11] A. Amjad, F. Azam, M. W. Anwar, and W. H. Butt, "A Systematic Review on the Data Interoperability of Application Layer Protocols in Industrial IoT," *IEEE Access*, vol. 9, pp. 96528–96545, 2021, doi: 10.1109/ACCESS.2021.3094763.
- [12] "Messaging that just works — RabbitMQ." <https://www.rabbitmq.com/> (accessed Nov. 25, 2022).
- [13] C. Akasiadis, V. Pitsilis, and C. D. Spyropoulos, "A Multi-Protocol IoT Platform Based on Open-Source Frameworks," *Sensors 2019, Vol. 19, Page 4217*, vol. 19, no. 19, p. 4217, Sep. 2019, doi: 10.3390/S19194217.
- [14] D. Mijić and E. Varga, "Unified IoT Platform Architecture Platforms as Major IoT Building Blocks," *Proceedings of the 2nd International Conference on Computing and Network Communications, CoCoNet 2018*, pp. 6–13, Sep. 2018, doi: 10.1109/COCONET.2018.8476881.
- [15] V. A. Barros *et al.*, "An IoT multi-protocol strategy for the interoperability of distinct communication protocols applied to web of things," *Proceedings of the 25th Brazilian Symposium on Multimedia and the Web, WebMedia 2019*, pp. 81–88, Oct. 2019, doi: 10.1145/3323503.3349546.
- [16] Z. Il-Agure and J. Dempered, "Review of Data Visualization Techniques in IoT Data," *8th International Conference on Information Technology Trends: Industry*

4.0: *Technology Trends and Solutions, ITT 2022*, pp. 167–171, 2022, doi: 10.1109/ITT56123.2022.9863948.

- [17] W. H. ADDAL, “A COMPARATIVE ANALYSIS OF RELATIONAL AND NON-RELATIONAL DATABASES FOR WEB APPLICATION,” NEAR EAST UNIVERSITY, 2019.
- [18] M. A. Kausar, “Suitability Of Influxdb Database For IoT Applications,” *Article in International Journal of Innovative Technology and Exploring Engineering*, no. 10, pp. 2278–3075, 2019, doi: 10.35940/ijitee.J9225.0881019.