

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Plataforma de Estandarización de Internet de las Cosas HayloT v2

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Telemática

Presentado por:

Jose Jhoan Zapata Becerra

Francheska Dayanna Iñiguez Herrera

GUAYAQUIL - ECUADOR

Año: 2023

DEDICATORIA

JOSE JHOAN ZAPATA BECERRA

Dedico este trabajo de a las personas que han sido mi faro y mi inspiración a lo largo de esta travesía académica. A mi familia, cuyo amor incondicional y apoyo constante me han sostenido en los momentos de dificultad. A mis profesores y mentores, quienes han compartido su conocimiento y experiencia con generosidad y me han guiado en el camino hacia el conocimiento y el crecimiento personal.

*FRANCHESKA DAYANNA IÑIGUEZ
HERRERA*

El presente proyecto se lo dedico a mi madre, Mónica Herrera, y a mi padre, Jorge Iñiguez, quienes con su amor y apoyo constante han sido la razón por la que he llegado hasta aquí; a mis abuelos, cuyos consejos y sabiduría han iluminado mi camino desde pequeña; a mi familia en general, por su inquebrantable respaldo a lo largo de mi vida.

AGRADECIMIENTOS

JOSE JHOAN ZAPATA BECERRA

Agradezco a mi madre Paulina Becerra, por su apoyo inquebrantable, amor y sacrificio a lo largo de mi carrera académica, a mi padre Felipe Zapata, por enseñarme lo que es el trabajo honesto y honrado. A mis hermanos y hermanas, Ricardo Zapata y Andrea Cobos, por su comprensión y aliento constante.

*FRANCHESKA DAYANNA IÑIGUEZ
HERRERA*

Agradezco a Dios, quien me ilumina en cada paso de mi vida; a mi madre, mi pilar fundamental; a mis profesores, cuyos conocimientos y dirección han influido en mi crecimiento como estudiante y persona; y a todos aquellos que han depositado su confianza en mí y han contribuido a mi crecimiento.

DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Jose Jhoan Zapata Becerra y Francheska Dayanna Iñiguez Herrera, damos 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"



Jose Jhoan Zapata Becerra



Francheska Dayanna Iñiguez Herrera

EVALUADORES

María José Ramírez Prado
PROFESOR DE LA MATERIA

José Eduardo Córdova García
PROFESOR TUTOR

RESUMEN

Las soluciones que utilizan el Internet de las Cosas (IoT) crecen exponencialmente, y con ello también surgen nuevos dispositivos que utilizan diferentes protocolos de comunicación y especificaciones, planteando diversos desafíos en términos de interoperabilidad.

Por ello, el presente proyecto desarrolla un sistema integrado que emplea un enfoque de estandarización basado en Project Haystack con el fin de simplificar el monitoreo de datos obtenidos a través de sensores, mediante la implementación de una interfaz de programación de aplicaciones (API) junto con una interfaz gráfica de usuario (GUI) que permiten el acceso a datos recopilados de sensores.

Para lograrlo, se realizó la instalación y configuración de sensores capaces de recolectar los datos, integrando aquellos que posean diferentes marcas y protocolos para asegurar la estandarización, también, se implementó un API que permitió la estandarización de los datos recibidos y se utilizaron protocolos Message Queuing Telemetry Transport (MQTT) y HyperText Transfer Protocol (HTTP) para permitir la comunicación bidireccional y transferencia de datos respectivamente. Además, para el almacenamiento de datos, se utilizó PostgreSQL para datos estructurados y MongoDB para no estructurados.

Como resultado, se logró una correcta exhibición de datos mediante una interfaz de usuario amigable, se realizaron distintas pruebas y análisis de rendimiento que permitieron demostrar una correcta optimización y análisis, logrando una implementación exitosa capaz de estandarizar los datos provenientes de diferentes sensores, facilitando la interoperabilidad, brindando flexibilidad al sistema y permitiendo visualizar los datos de manera sencilla e intuitiva.

Palabras Clave: IoT, Estandarización, Plataforma Web, APIs

ABSTRACT

Solutions using the Internet of Things (IoT) are growing exponentially, and with it also emerge new devices using different communication protocols and specifications, posing several challenges in terms of interoperability.

Therefore, the present project develops an integrated system that employs a standardization approach based on Project Haystack in order to simplify the monitoring of data obtained through sensors, by implementing an application programming interface (API) together with a graphical user interface (GUI) that allows access to data collected from sensors.

To achieve this, sensors capable of collecting data were installed and configured, integrating those with different brands and protocols to ensure standardization. An API was also implemented to standardize the data received, and Message Queuing Telemetry Transport (MQTT) and HyperText Transfer Protocol (HTTP) protocols were used to allow bidirectional communication and data transfer, respectively. In addition, for data storage, PostgreSQL was used for structured data and MongoDB for unstructured data.

As a result, a correct data display was achieved through a friendly user interface, different tests and performance analysis were performed to demonstrate a correct optimization and analysis, achieving a successful implementation capable of standardizing data from different sensors, facilitating interoperability, providing flexibility to the system and allowing to visualize the data in a simple and intuitive way.

Keywords: IoT, Standardization, Web Platform, APIs

ÍNDICE GENERAL

RESUMEN	i
ABSTRACT	iii
ABREVIATURAS	vii
SIMBOLOGÍA	ix
INDICE DE FIGURAS	ix
1 INTRODUCCIÓN	1
1.1 Problemática	2
1.2 Justificación	2
1.3 Objetivos	3
1.3.1 Objetivo General	3
1.3.2 Objetivos Específicos	3
1.4 Alcance y Limitaciones	4
1.5 Marco Teórico	4
1.6 Descripción de Escenarios para la Propuesta	5
1.7 Estado del Arte	6
2 METODOLOGÍA	9
2.1 Materiales y Métodos	12
3 RESULTADOS DE IMPLEMENTACIÓN	15
3.1 Integración de Sensores	15
3.2 Exhibición de datos	17
3.3 Pruebas y Análisis de rendimiento	19
3.3.1 Mejora de Visualización de Gráficos	19

3.3.2 Optimización de Código	20
3.3.3 Análisis de Número de Variables	21
3.3.4 Análisis de Tiempo de Respuesta	22
4 CONCLUSIONES Y LÍNEAS FUTURAS	25
4.1 Conclusiones	25
4.2 Recomendaciones	26
4.3 Líneas Futuras	26
BIBLIOGRAFÍA	27
APÉNDICES	28
Apéndice A: Análisis de Costos	31

ABREVIATURAS

IoT	Internet of Things
ESPOL	Escuela Superior Politécnica del Litoral
API	Application Programming Interface
API REST	Representational State Transfer API
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
PREC	Personalized Recommendations for Efficient Consumption
JSON	JavaScript Object Notation
MQTT	Message Queue Telemetry Transport

SIMBOLOGÍA

s	Segundos
%	Porcentaje
min	Minuto

ÍNDICE DE FIGURAS

2.1 Esquema de Implementación	10
2.2 Pasos para Realizar Envío y Consulta de Datos	11
3.1 Pestaña de Registro de Entidades	15
3.2 Diagrama de Flujo para Registro de Entidades	16
3.3 Búsqueda de Sensores Registrados	17
3.4 Gráfico con Datos del Sensor	18
3.5 Sitios, Espacios y Sensores Registrados	18
3.6 Gráfico - Antes de la Mejora	19
3.7 Gráfico - Después de la Mejora	19
3.8 Antes de Optimización de Código	20
3.9 Después de Optimización de Código	21
3.10 Tiempo de Respuesta Según el Número de Variables Previo a la Optimización	21
3.11 Rendimiento Según Número de Solicitudes	22

CAPÍTULO 1

1. INTRODUCCIÓN

El Internet de las Cosas (IoT) es un campo tecnológico que busca establecer conexiones entre dispositivos y el Internet mediante el uso de tecnologías asequibles y de alta velocidad. De este modo, objetos, elementos cotidianos o dispositivos industriales pueden transmitir datos continuamente utilizando pequeños módulos que incluyen sensores y microprocesadores, (Villamil et al., 2020).

Diversas soluciones e innovaciones de IoT continúan creciendo exponencialmente según lo requieren diferentes áreas tales como la industrial, construcción, seguridad, entre otras, aumentando paralelamente la producción masiva de dispositivos de bajo coste de diversos proveedores. Según las previsiones de Transforma Insights, en 2021 había 11.300 millones de dispositivos IoT activos, y se ha anticipado un aumento para 2030 hasta los 29.400 millones, con una tasa de crecimiento anual del 12%, (Morrish and Arnott, n.d.).

La disponibilidad, asequibilidad, entre otros avances de IoT, han permitido su uso y resaltan la importancia en diversos ámbitos para lograr la conectividad y acceso remoto de los mismos, sin embargo, existen distintos factores que dificultan lo antes mencionado, tales como la compatibilidad, marcas, protocolos de comunicación, e inclusive la falta de protocolos de estandarización, (Zikria et al., 2019).

Este trabajo representa un gran avance en la implementación de un sistema estandarizado IoT y describe el proyecto ESPOLHayIoT. El proyecto se desarrolló en la Escuela Superior Politécnica del Litoral (ESPOL) como mejora del sistema de gestión del consumo energético del campus.

1.1 Problemática

El escenario de IoT ha evolucionado durante los últimos años gracias a la rápida propagación de sus diversas aplicaciones en diversos sectores, (Wijethilaka and Liyanage, 2021), también, las nuevas tecnologías han permitido la generación de un gran volumen de datos en tiempo real, las cuales han permitido mejores conexiones entre objetos físicos. Por esta razón, se han producido avances significativos en cuanto al desarrollo de sensores más precisos, redes de comunicación más rápidas y protocolos de datos más eficientes, (Nižetić et al., 2020).

Sin embargo, a medida que estas tecnologías avanzan, también surgen desafíos complejos en relación con el manejo, almacenamiento y visualización de los datos generados, lo que plantea la necesidad de abordar la problemática que se presenta a continuación.

Las empresas se enfrentan a un entorno en el que cada dispositivo tiene sus propias especificaciones técnicas y formatos de datos, dificultando la interoperabilidad y la integración de los sistemas existentes, esto es debido a la variedad de estándares tecnológicos que incluyen una amplia gama de protocolos de red, protocolos de comunicación, estructuras de datos y estándares de captura de datos, (Lee et al., 2021). También, la falta de reglas comunes para la representación y el almacenamiento de datos dificulta la reutilización y el intercambio de información entre varios dispositivos y aplicaciones IoT, (Al-Qaseemi et al., 2016).

Por esta razón, se ha propuesto el desarrollo de una plataforma que permita la estandarización de datos que se generen por los dispositivos IoT independientemente de los protocolos o estándares que utilicen individualmente, permitiendo establecer un estándar para el almacenamiento y visualización de datos.

1.2 Justificación

Dada la problemática que gira en torno a la diversidad de protocolos, plataformas, y estándares que dificultan la interoperabilidad entre dispositivos, con este proyecto se

busca establecer la interoperabilidad para el desarrollo y gestión de soluciones IoT.

Mediante la estandarización no sólo mejorará la interoperabilidad, sino también el acceso y el análisis de datos generados por los dispositivos, esto es debido a que mediante Haystack¹, la estandarización de estos sistemas permitirá la normalización y estructuración de los datos, facilitando así su acceso y análisis, (Quinn and McArthur, 2021). Esto conducirá a una toma de decisiones más informada y un mayor aprovechamiento del potencial de los datos generados por IoT.

Además, la plataforma está diseñada para ser escalable, es decir, podrá crecer y adaptarse a medida que se añadan nuevos dispositivos y tecnologías IoT en el futuro, garantizando que sea estable a largo plazo y capaz de evolucionar en función de las distintas necesidades que surjan.

Por último, este proyecto también se justifica desde el punto de vista económico, ya que al estandarizar los sistemas IoT mediante Haystack, se evitarán los costos asociados con la creación de interfaces o adaptadores que permitan la integración de sistemas heterogéneos, (Roy et al., 2019). Esto resultará en ahorros significativos en cuanto a términos de costos operativos y de implementación. Además, al simplificar la integración y mejorar el acceso a los datos, se maximizará la eficiencia de los sistemas IoT, y consecuentemente ahorrando costos a largo plazo.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un sistema integrado que emplee un enfoque de estandarización con el fin de simplificar el monitoreo de datos.

1.3.2 Objetivos Específicos

- Implementar una API y una interfaz gráfica que permitan el acceso a datos recopilados de sensores.

¹<https://project-haystack.org/>

- Diseñar una plataforma de estandarización de sensores que exhiba los datos recolectados por los mismos.

1.4 Alcance y Limitaciones

Este documento busca presentar una solución tecnológica que aborde el almacenamiento y la presentación de datos provenientes de diversos tipos de sensores, utilizando Haystack como ontología para lograr una estandarización precisa de los datos.

También, a través de un dashboard, los usuarios podrán acceder a los datos de manera clara y personalizada, según las necesidades específicas que cada uno de ellos requiera.

Sin embargo, es importante destacar que, en su estado inicial, el proyecto se desarrolla en colaboración con la ESPOL. Por lo tanto, los sensores utilizados y validados en su totalidad se encuentran limitados a la disponibilidad proporcionada por la ESPOL. Asimismo, para acceder a la solución, los usuarios deben estar dentro de la red de la universidad, ya que se ha implementado en los servidores de la institución.

Además, se espera que en futuras etapas del proyecto se puedan explorar opciones de ampliación para incorporar otros dispositivos y permitir un acceso más amplio a la solución. De esta manera, se buscará brindar una mayor flexibilidad y escalabilidad, ampliando las oportunidades de utilización de la plataforma a otros entornos y usuarios externos a la ESPOL.

1.5 Marco Teórico

Para abordar el tema mencionado, el marco teórico actual presentará los principios de IoT, estándares de etiquetado de datos, tecnologías de comunicación y los componentes requeridos para el diseño y desarrollo de HayloT.

En primer lugar, se toman en cuenta los conceptos fundamentales del Internet de las cosas (IoT), enfatizando su capacidad para conectar dispositivos físicos y recopilar datos en tiempo real, (Hariri et al., 2019). Las plataformas de Internet de las cosas son cruciales

para administrar los datos generados por los dispositivos. En este contexto, HayloT se presenta como una solución completa que permitirá la gestión eficiente de los datos obtenidos y la gestión centralizada de los dispositivos IoT.

Además, se aborda el estándar de etiquetado de datos de Project Haystack, que se utilizará en la plataforma. Este estándar proporciona una estructura común para estandarizar y organizar los datos de varios dispositivos IoT, (Quinn and McArthur, 2021), al integrarlo en la plataforma, se garantizará que los datos recopilados estén estructurados de manera uniforme, lo que facilitará su integración y análisis posterior.

También se hace referencia al uso de una API REST, la cual es un enfoque ampliamente utilizado en el desarrollo de aplicaciones web y servicios de IoT, (“¿Qué es una API REST? | IBM — ibm.com”, n.d.). REST (Representational State Transfer) se basa en el protocolo HTTP y utiliza métodos (GET, POST, PUT, DELETE) para acceder y gestionar recursos a través de una interfaz uniforme, la implementación de una API REST en la plataforma permitirá el acceso y la gestión de los datos recopilados de los sensores IoT.

Por último, se destaca la importancia de una interfaz gráfica de usuario (GUI) intuitiva, una parte esencial de cualquier sistema o aplicación ya que es la forma en que los usuarios interactúan con la plataforma, (Banerjee, 2017). La GUI permitirá a los usuarios interactuar de manera visual y sencilla, visualizando los dispositivos conectados, así como accediendo a los datos recopilados por cada sensor.

Finalmente, la implementación de una API REST basada en el protocolo HTTP (Hypertext Transfer Protocol) permitirá a los usuarios acceder y realizar consultas de datos de manera segura, esta plataforma ofrecerá una interfaz segura y fácil de usar para administrar y visualizar los datos recopilados de los dispositivos IoT.

1.6 Descripción de Escenarios para la Propuesta

El Proyecto de Monitoreo y Gestión Energética en el Campus Gustavo Galindo (ESPOL) implementó una plataforma llamada "PREC" (Personalized Recommendations for Efficient Consumption) para reducir el consumo eléctrico a través del monitoreo y control de recursos mediante módulos IoT. Se propone el uso del estándar de etiquetado

de datos de Project Haystack para estandarizar los datos no estructurados de los módulos IoT utilizados en sistemas de gestión de Edificios Inteligentes.

Esto permite abstraer el tratamiento de datos para servicios o aplicaciones de terceros que requieran acceder a los datos generados por los módulos IoT, cabe recalcar que esto no sólo se aplica a "PREC", sino que también abarca diversos proyectos en desarrollo que hacen uso de dispositivos IoT.

El acceso a los datos estandarizados se realiza a través de una Interfaz de Programación de Aplicaciones con Transferencia de Estado Representacional (API REST). Esta API recibe solicitudes mediante HTTP a través de rutas definidas, los sensores envían la información de lectura en un objeto JSON (JavaScript Object Notation) según el modelo de etiquetado de datos aplicado previamente (Ontología Haystack).

Se planea implementar distintos proyectos desarrollados en ESPOL de monitoreo y gestión de diversos dispositivos IoT, se utiliza el estándar de etiquetado de datos de Project Haystack para estandarizar los datos de los módulos IoT, y se accede a estos datos a través de una plataforma web, cuyos datos fueron almacenados mediante una API REST utilizando el protocolo HTTP, permitiendo así una mejor interoperabilidad y utilización de los datos generados por los módulos IoT en aplicaciones y servicios externos.

1.7 Estado del Arte

Debido a la problemática antes mencionada en la que se presentan los desafíos relacionados a la interoperabilidad e integración efectiva de dispositivos en un ecosistema IoT, se han creado numerosos proyectos y métodos para abordar la estandarización de datos.

El artículo, (Quinn and McArthur, 2022) realiza una comparación entre dos enfoques para respaldar aplicaciones de edificios inteligentes: Brick y Haystack. Ambos tienen como objetivo proporcionar un marco de trabajo para el modelado de datos y la interoperabilidad en entornos de edificios inteligentes.

El objetivo principal de los autores es facilitar la implementación de aplicaciones de edificios inteligentes para que se logre una puesta en servicio eficiente y continua de los edificios, lo que finalmente resultará en un uso máximo de energía, un costo mínimo y una mejora en la salud y la comodidad de los ocupantes.

Según los resultados obtenidos, recomiendan el uso de Brick, ya que mostró valores de evaluación más altos en cuanto a integridad (59%) y expresividad (100%), en comparación con Haystack, que obtuvo un 43% y un 96% respectivamente. Además, Brick exhibió cinco de las seis cualidades deseables, mientras que Haystack solo mostró tres.

Por otro lado, el artículo "Autonomous convergence mechanisms for collaborative crowd-sourced data-modeling", (Lübben and Pahl, 2022) basa su enfoque en el análisis de modelos existentes con el objetivo de identificar similitudes con nuevos modelos, de esta manera, se busca reutilizar o ampliar los que son similares para evitar la redundancia de información.

La evaluación de este enfoque se realizó utilizando un conjunto de datos que consta de 1200 modelos automáticamente recopilados de las iniciativas de modelado de datos IoT más relevantes, Project Haystack, IoTSchema y BrickSchema. Los resultados de la evaluación demostraron una escalabilidad lineal, lo que significa que el enfoque es capaz de manejar eficientemente un crecimiento en el número de modelos de datos. Además, se obtuvo una alta precisión en la identificación de similitudes entre ellos, lo que respalda la efectividad y confiabilidad del enfoque propuesto.

Finalmente, (Jang et al., 2020) en su artículo denominado "IoT Device Auto-Tagging Using Transformers", proporciona una técnica de "Etiquetado" que identifica automáticamente el dispositivo en función de la información recopilada de dispositivos de IoT o dispositivos de sensores, esta técnica se desarrolló en base a un algoritmo de procesamiento del lenguaje natural basado en el mecanismo de atención entre los modelos de aprendizaje automático.

Esto fue implementado debido a que, en una situación en la que muchas empresas lanzan diversos tipos de dispositivos IoT, no es fácil mantener información sobre todos los dispositivos. Es decir, cuando se registran y gestionan manualmente los puntos del

dispositivo, se lleva a cabo a través de una variedad de procedimientos, sin embargo, si se procede al proceso utilizando la función de Auto-Tagging, se puede esperar reducir el tiempo de trabajo de construcción al reducir la asignación de recursos en comparación con el estado existente.

El presente proyecto se distingue por su enfoque innovador y su capacidad para abordar las limitaciones existentes en los artículos previamente mencionados, una de las características principales es su interfaz gráfica intuitiva, la cual es diseñada para visualizar los datos de una manera clara, permitiendo así analizar y comprender las tendencias de los datos generados por los dispositivos IoT, y consecuentemente facilitar la toma de decisiones informadas.

Además, a diferencia del artículo (Quinn and McArthur, 2022), se emplea el estándar de etiquetado de Project Haystack debido a que se centra en el etiquetado de datos relacionados con sistemas de automatización. A diferencia de Brick, Haystack es una solución más flexible y orientada a integrar datos de distintas fuentes.

También, se maneja un etiquetado manual, en comparación con (Jang et al., 2020), ya que fue considerado como limitante la precisión de los algoritmos de procesamiento dependiendo de la calidad y cantidad de datos, también los cambios en los dispositivos y el proceso de recopilación de información debe manejar grandes volúmenes de datos ya que el sistema es escalable. A diferencia de los trabajos de investigación previamente mencionados, este proyecto ha superado distintas limitaciones y podrá adaptarse cada vez que se agreguen nuevos dispositivos.

CAPÍTULO 2

2. METODOLOGÍA

Se ha estructurado la implementación en distintas fases con el objetivo de establecer un estándar de datos y unificar los módulos y sensores IoT.

En primer lugar, se realiza la instalación y configuración de los módulos y sensores IoT dentro del entorno del proyecto, estos dispositivos están diseñados para recolectar datos de diversas variables.

Para garantizar la estandarización en diferentes escenarios, se propuso la integración de sensores de diferentes marcas, tipos de datos y protocolos de comunicación. Además, para tener en cuenta una variedad de protocolos de comunicación, se integraron sensores de varios fabricantes, estas características seleccionadas demostrarán la correcta centralización y estandarización de los dispositivos.

También, se implementa un API cuyo objetivo principal es estandarizar los datos recibidos provenientes de los diferentes sensores IoT mencionados anteriormente, permitiendo así simplificar el acceso y la integración de los datos de manera más óptima. Como la API sólo acepta datos a través del protocolo HTTP, es necesario crear APIs intermediarias para recopilar los datos mediante los protocolos manejados por cada sensor; una vez obtenidos los datos, se envían a la API principal de HayIoT a través de HTTP. El API emplea dos protocolos de comunicación para la transferencia de datos.

Se utiliza el protocolo MQTT (Message Queue Telemetry Transport), el cual permite una comunicación bidireccional entre los clientes y el broker MQTT, esto permite que los clientes MQTT se suscriban a un tópico específico donde recibirán los datos provenientes de los sensores.

Además, se utiliza el protocolo HTTP para la transferencia de datos. A través de métodos de petición y respuesta propios del modelo cliente/servidor, los datos provenientes de sensores que utilizan este protocolo se envían al API mediante solicitudes POST.

Tanto en el caso del protocolo MQTT como del protocolo HTTP, el API se encarga de registrar las lecturas recibidas en una tabla transaccional en una base de datos PostgreSQL.

Esta tabla incluye un identificador único del sensor y un campo de tipo JSON, que contiene el modelo de datos del sensor. Este modelo de datos se configura previamente mediante una interfaz web administrada por el usuario y sigue el estándar de etiquetado de datos definido por Project Haystack, esta estandarización asegura la estructura consistente de los datos y permite una comprensión sencilla.

La Figura 2.1 ilustra las relaciones y el flujo de datos entre el hardware y el software en todo el sistema. Se observa que todos los elementos y recursos de la infraestructura son gestionados a través del API propuesto, mediante el cual se utilizarán PostgreSQL y MongoDB.

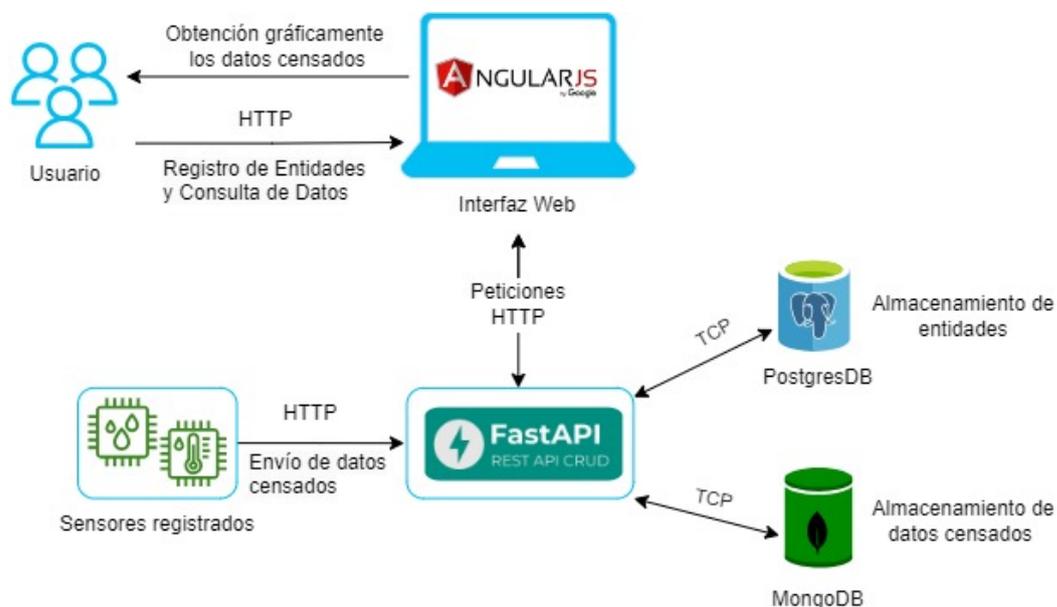


Figura 2.1: Esquema de Implementación

Por otro lado, la Figura 2.2 detalla los pasos esenciales que el usuario debe seguir para lograr una exitosa comunicación entre el sensor y la plataforma. A continuación, se presenta una descripción exhaustiva de cada etapa involucrada:

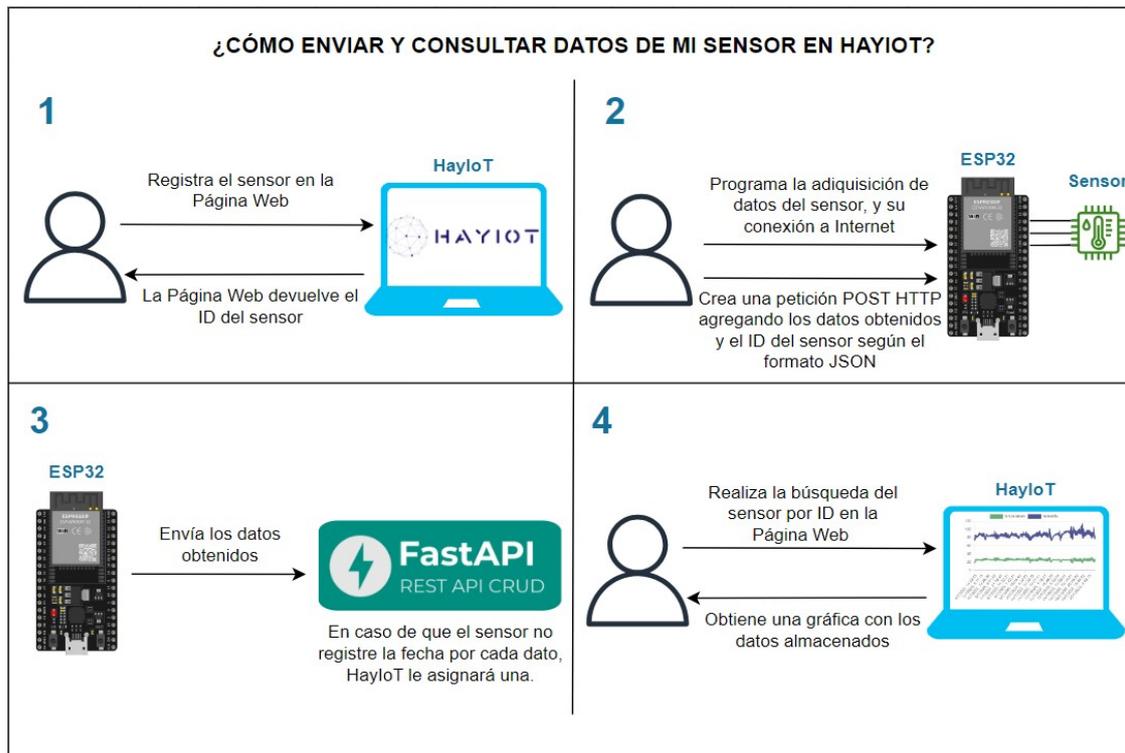


Figura 2.2: Pasos para Realizar Envío y Consulta de Datos

En un principio, el usuario deberá realizar el registro del sensor en la plataforma HayloT. Al completar este proceso, la plataforma generará y proporcionará al usuario un ID único para el sensor, el cual será utilizado para identificar el dispositivo dentro del sistema.

Una vez obtenido el ID del sensor, el siguiente paso consiste en programar la adquisición de datos del sensor y asegurar su correcta conexión a internet. El envío de los datos adquiridos hacia la plataforma HayloT se realiza mediante una petición POST HTTP. En este punto, el usuario deberá crear dicha solicitud incluyendo tanto los datos obtenidos por el sensor como el ID del dispositivo, todo esto siguiendo el formato JSON para facilitar la estructura de la información transmitida.

Una vez que los datos son enviados hacia la plataforma a través de la API correspondiente, en situaciones donde el sensor no registre la fecha asociada a cada dato recopilado, HayloT realizará una asignación automática de fechas a dichos registros, garantizando así una correcta y ordenada catalogación de la información.

Por último, tras completar el proceso de envío de datos, el usuario puede realizar búsquedas en la plataforma para acceder a la información específica de su sensor. HayloT ofrece dos opciones de búsqueda: por ID o por Nombre del sensor. Al seleccionar

una de estas alternativas, el sistema proporcionará una gráfica detallada que representa los datos almacenados del sensor en el rango de fechas escogido, permitiendo así un análisis visual y accesible de la información recopilada.

2.1 Materiales y Métodos

A continuación, se presentarán los componentes, estándares y recursos clave utilizados dentro del proyecto para establecer la infraestructura necesaria y la implementación de la plataforma.

- **Dispositivos Sonoff:** Son interruptores de relé que se conectan a la red eléctrica y permiten controlar de forma remota la energía hacia los dispositivos conectados a ellos, como luces, electrodomésticos o equipos electrónicos controlables a través de una red Wi-Fi, (Sonoff.tech, 2021). Además, los Sonoffs pueden ser modificados o flasheados con firmware personalizado, como Tasmota o ESPHome, que amplían sus capacidades y los hacen compatibles con diferentes protocolos de comunicación, como MQTT, (Valov et al., 2020).

Esto permite la integración de los Sonoffs en sistemas más complejos y personalizados, como HayloT, ya que los datos capturados por los Sonoffs serán publicados en un servidor donde se ha implementado un broker MQTT, estos datos se transmitirán utilizando MQTT bajo un tópico específico.

- **PostgreSQL:** Es un sistema de gestión de bases de datos relacional de código abierto que desempeñará un papel central en el proyecto, (Group, 2021). PostgreSQL permitirá almacenar y organizar eficientemente los datos recopilados por los sensores, ofreciendo un entorno seguro y escalable para su gestión.

Además, la capacidad de PostgreSQL para integrarse con otros componentes, como servidores web y servicios de aplicaciones, (Viloria et al., 2019), permitirá una comunicación fluida entre los dispositivos IoT y la plataforma, facilitando el monitoreo y acceso a los datos a través de la API y la interfaz gráfica.

- **MongoDB:** MongoDB es un sistema de gestión de bases de datos NoSQL de código abierto, a diferencia de las bases de datos relacionales, (MongoDB, 2021), MongoDB utiliza un modelo de datos flexible basado en documentos JSON, lo que lo hace ideal para almacenar y gestionar datos no estructurados o semiestructurados generados por los dispositivos IoT.

Además, la flexibilidad de MongoDB facilitará la evolución y expansión de los datos a medida que se agreguen nuevos sensores y se generen diferentes tipos de información. La integración de MongoDB con la plataforma de estandarización permitirá un acceso rápido y sencillo a los datos mediante consultas flexibles.

- **Project Haystack (Haystack, n.d.):** Se dedica a estandarizar los modelos de datos semánticos y los servicios web con el propósito de simplificar la extracción de valor a partir de la gran cantidad de datos generados por dispositivos inteligentes presentes en hogares, edificios, fábricas y ciudades. Sus aplicaciones abarcan áreas como la automatización, el control, la gestión energética, la climatización, la iluminación y otros sistemas ambientales.

Este estándar permitirá la asignación de etiquetas a cada dispositivo de IoT, proporcionando una descripción completa de toda la información necesaria sobre cada uno de ellos, siendo esta información una valiosa fuente de datos para los usuarios al realizar solicitudes o para otras API. De esta manera, se facilitará la identificación y acceso a los detalles específicos de cada dispositivo IoT, lo que mejorará la experiencia del usuario y permitirá un uso más eficiente de los datos generados en diversas aplicaciones y servicios.

- **Message Queuing Telemetry Transport (MQTT):** Es un protocolo de mensajería ligero y de bajo consumo que hace uso de operaciones de publicación y suscripción para intercambiar información entre clientes y servidores, (V et al., 2022), además, proporciona conexiones sencillas y fiables con un ancho de banda de red limitado.

En este contexto, MQTT se utiliza para facilitar la recopilación de datos entre los sensores y la plataforma. Los sensores pueden enviar datos de forma eficiente a través de este protocolo hacia un broker MQTT para posteriormente a través de un script realizar una petición HTTP y enviarlos a la plataforma de HayIoT. Estos datos

se envían en forma de mensajes, que contienen información específica recopilada por los sensores.

- **Protocolo de Transferencia de Hipertexto (HTTP):** Es un protocolo de comunicación utilizado para la transferencia de información. HTTP permite la comunicación entre un cliente (como un navegador web) y un servidor, facilitando el intercambio de recursos como páginas web, imágenes y otros archivos, (Bressoud and White, 2020).

Este protocolo se basa en el modelo de solicitud y respuesta, donde el cliente envía una solicitud al servidor y este responde con los datos solicitados. Además, HTTP también proporciona métodos para la manipulación de recursos, como GET para obtener información, POST para enviar datos y realizar cambios, entre otros, dentro de la propuesta, a través de un método POST se podrán recibir los datos de aquellos sensores que usen este protocolo.

CAPÍTULO 3

3. RESULTADOS DE IMPLEMENTACIÓN

En este capítulo, se presentan los resultados obtenidos tras la implementación y evaluación de HayloT, junto con las pruebas que se realizan para analizar su rendimiento. A lo largo de este capítulo, se detallarán los resultados obtenidos y pruebas realizadas, proporcionando evidencia concreta y respuestas a los objetivos planteados previamente.

3.1 Integración de Sensores

Para registrar los sensores, se requiere acceder a la pestaña "Registro" como se muestra en la Figura 3.1, aquí, se podrán ingresar los detalles de todas las entidades relevantes, incluyendo Sitio, Espacio y Sensor.

Una vez completado el registro de los datos del sensor, se generará un ID único, el cual es fundamental y será utilizado para las solicitudes POST que se realizan hacia el API.

The image displays three side-by-side registration forms, each titled "Registrar Entidad".

- Form 1 (Sensor):** Features a "Tipo" dropdown menu with "Sensor" selected. Below it are three stacked dropdown menus for "Sitio", "Espacio", and "Descripción". At the bottom left is a "Tipo" dropdown menu. A blue "Registrar" button is positioned at the bottom center.
- Form 2 (Espacio):** Features a "Tipo" dropdown menu with "Espacio" selected. Below it are three stacked input fields for "Sitio", "Identificador", and "Espacio". A blue "Registrar" button is positioned at the bottom center.
- Form 3 (Sitio):** Features a "Tipo" dropdown menu with "Sitio" selected. Below it are two stacked input fields for "Identificador" and "Descripción". A blue "Registrar" button is positioned at the bottom center.

Figura 3.1: Pestaña de Registro de Entidades

La figura 3.2 muestra un diagrama de flujo mediante el cual se ha logrado verificar de manera efectiva el registro de entidades, tales como sitios, espacios y sensores. Durante las pruebas realizadas, se ha constatado que el diagrama de flujo guía de forma clara y precisa a los usuarios en el proceso de registro, asegurando la identificación correcta de cada sensor y su asociación con el sitio y espacio correspondiente.

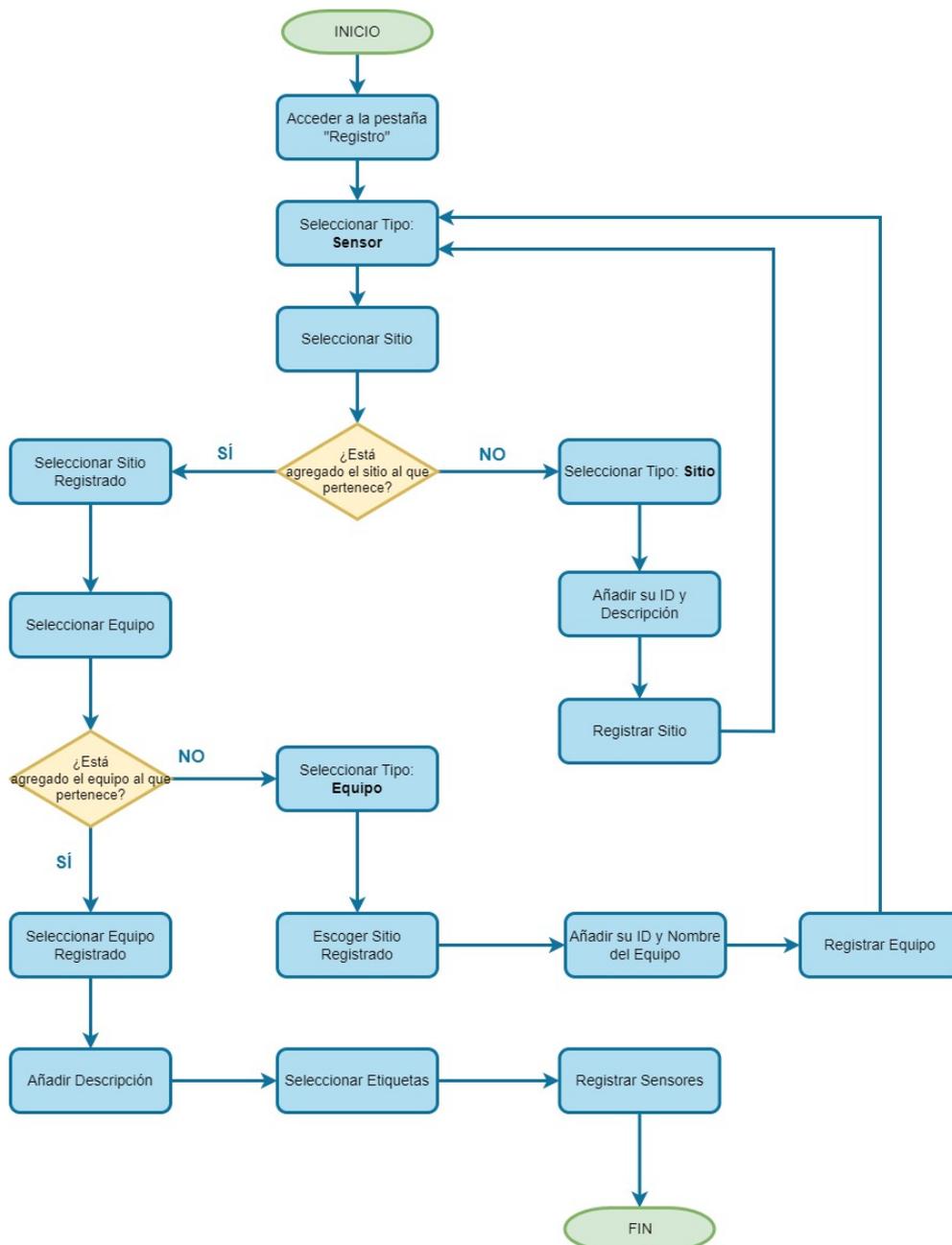


Figura 3.2: Diagrama de Flujo para Registro de Entidades

Los resultados de las integraciones de sensores demuestran que la plataforma es eficiente y versátil en cuanto a la conexión de diversos dispositivos IoT, siendo compatible con diversos tipos de sensores sin importar las tecnologías o protocolos de comunicación

utilizados. Además, la plataforma es capaz de incorporar con éxito diferentes tipos de sensores, como temperatura, humedad, electricidad, y demás.

En la figura 3.3 es posible observar los distintos sensores SonOff conectados dentro del Laboratorio de Sistemas Telemáticos ubicado en la ESPOL, junto con sus características respectivas mediante la opción de búsqueda de sensores, la cual se encuentra dentro de la pestaña "Dashboard" en la plataforma.

Exploración de data

🔍 s31 Nombre Buscar

Seleccione un sensor

Id	Site	Space	Type	Description	Edit
64b5cc45cca9b534b74529c2	11C	LT1	electricity	s31_11	✎
64b5cc2dcca9b534b7452985	11C	LT1	electricity	s31_10	✎
64b5cc16cca9b534b7452948	11C	LT1	electricity	s31_09	✎
64b5cbfecca9b534b74528ed	11C	LT1	electricity	s31_08	✎
64b5cb98cca9b534b74527c0	11C	LT1	electricity	s31_07	✎
648ca96a72114d2f0457eba0	11C	LT1	electricity	s31_06-switch	✎
648ca97f72114d2f0457ebb7	11C	LT1	electricity	s31_05-router	✎
648ca81672114d2f0457eb89	11C	LT1	electricity	s31_03-proyector	✎
648ca97e72114d2f0457ebaf	11C	LT1	electricity	s31_02-monitor	✎
648ca98972114d2f0457ebbf	11C	LT1	electricity	s31_01-cpu	✎

Figura 3.3: Búsqueda de Sensores Registrados

3.2 Exhibición de datos

En cuanto a la exhibición de datos, la interfaz de usuario de la plataforma proporciona una experiencia intuitiva, permitiendo visualizar cambios y tendencias de manera efectiva. Dentro de la plataforma, es posible seleccionar un sensor y posteriormente filtrar los datos por rango de fechas para así poder visualizar toda la información medida y registrada por cada uno de ellos.

La figura 3.4 muestra un ejemplo de la visualización de gráficos mediante el sensor SonOff conectado en el monitor del Laboratorio de Sistemas Telemáticos entre el 17/07/2023 y el 18/07/2023, donde se puede observar claramente las horas en las que el monitor es usado con más frecuencia.

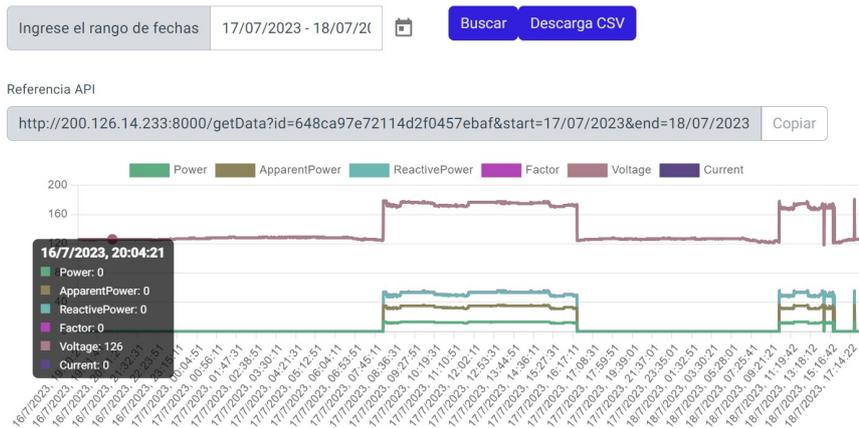


Figura 3.4: Gráfico con Datos del Sensor

Además, a través de la pestaña Entidades de la plataforma, la cual está representada en la figura 3.5, los usuarios tendrán la capacidad de visualizar de manera organizada y clara todos los sitios, espacios y sensores que han sido registrados previamente dentro de la plataforma, y también es posible eliminar entidades siempre y cuando sea un usuario administrador que posea la clave requerida.

Entidades Registradas

Sitios Registrados

Id	Description
2A	Admisiones
12C	FIMCP
11E	FIEC Vieja
11D	Aulas FIEC
11C	Laboratorios Fiec
11A	FIEC Nueva

Espacios Registrados

Identificador	Sitio	Descripcion
MA	11E	Medidor Agua
M1	2A	Medidores
LT1	11C	Papaya
LRA	11C	Sensores

Sensores Registrados

Identificador	Sitio	Espacio	Tipo	Descripción
64b5cc45cca9b534b74529c2	11C	LT1	electricity	s31_11
64b5cc2dcca9b534b7452985	11C	LT1	electricity	s31_10
64b5cc16cca9b534b7452948	11C	LT1	electricity	s31_09
64b5cbfecca9b534b74528ed	11C	LT1	electricity	s31_08
64b5cb98cca9b534b74527c0	11C	LT1	electricity	s31_07
648ca96a72114d2f0457eba0	11C	LT1	electricity	s31_06-switch
648ca97f72114d2f0457ebb7	11C	LT1	electricity	s31_05-router
648ca81672114d2f0457eb89	11C	LT1	electricity	s31_03-proyector
648ca97e72114d2f0457ebaf	11C	LT1	electricity	s31_02-monitor
648ca98972114d2f0457ebbf	11C	LT1	electricity	s31_01-cpu

Figura 3.5: Sitios, Espacios y Sensores Registrados

Esta funcionalidad se convierte en una herramienta indispensable para garantizar un orden adecuado de cada uno de los registros realizados, lo que facilita la identificación de los sensores, espacios y sitios que están siendo ubicados.

3.3 Pruebas y Análisis de rendimiento

El análisis de rendimiento debe demostrar que la plataforma es altamente confiable, incluso bajo múltiples cargas de datos. A continuación, se presentan diferentes pruebas de estrés para probar el tiempo de respuesta cuando se generen grandes volúmenes de datos simultáneamente previo y posteriormente a la optimización realizada.

3.3.1 Mejora de Visualización de Gráficos

En primer lugar, se mejoraron los gráficos de tal manera que no se vean picos que alteren la percepción de los valores en los datos, permitiendo una mejor visualización de los mismos; tal y como se observan en las figuras 3.6 y 3.7, las cuales muestran la gráfica que posee los datos del sensor Shelly.

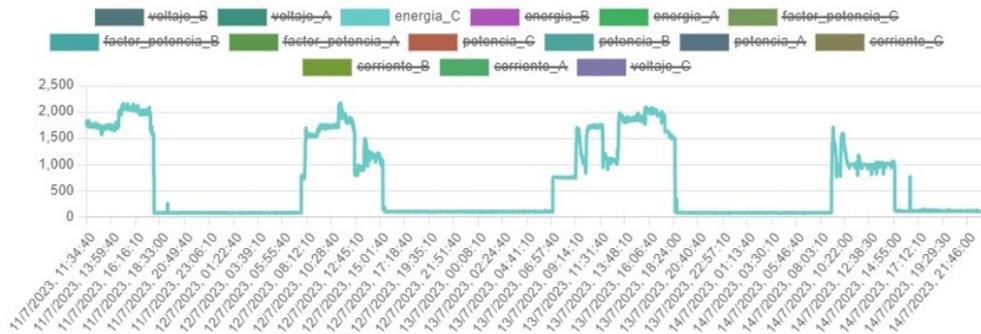


Figura 3.6: Gráfico - Antes de la Mejora

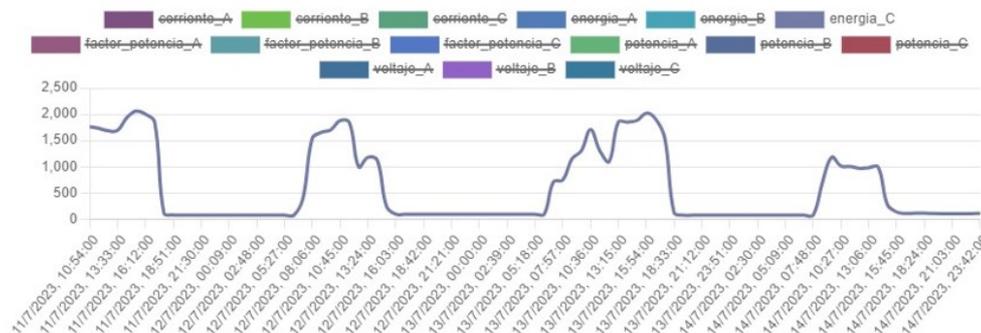


Figura 3.7: Gráfico - Después de la Mejora

3.3.2 Optimización de Código

Para comprobar el correcto funcionamiento de la plataforma, y que los datos se estén cargando en el menor tiempo requerido, se realiza la optimización de código en el Backend, eliminando código innecesario que ocasionaba un mayor tiempo de respuesta al momento de obtener datos de los distintos sensores tanto antes como después de la optimización y muestreo.

Por otro lado, se estaba trabajando con MongoDB tradicional, sin embargo, el tiempo de respuesta y el modo de procesamiento respecto a la cantidad de usuarios no era óptimo, para solucionar este problema se usaron series temporales en MongoDB y Window Functions, logrando un menor tiempo de respuesta.

De esta manera, es posible optimizar el tiempo de respuesta tal y como se muestra en las figuras 3.8 y 3.9, donde se puede visualizar que previo a la optimización de código, el tiempo de respuesta era de 1min 16.62s, y después de este, es de 42.63s; logrando optimizar el tiempo en un 55.64%.

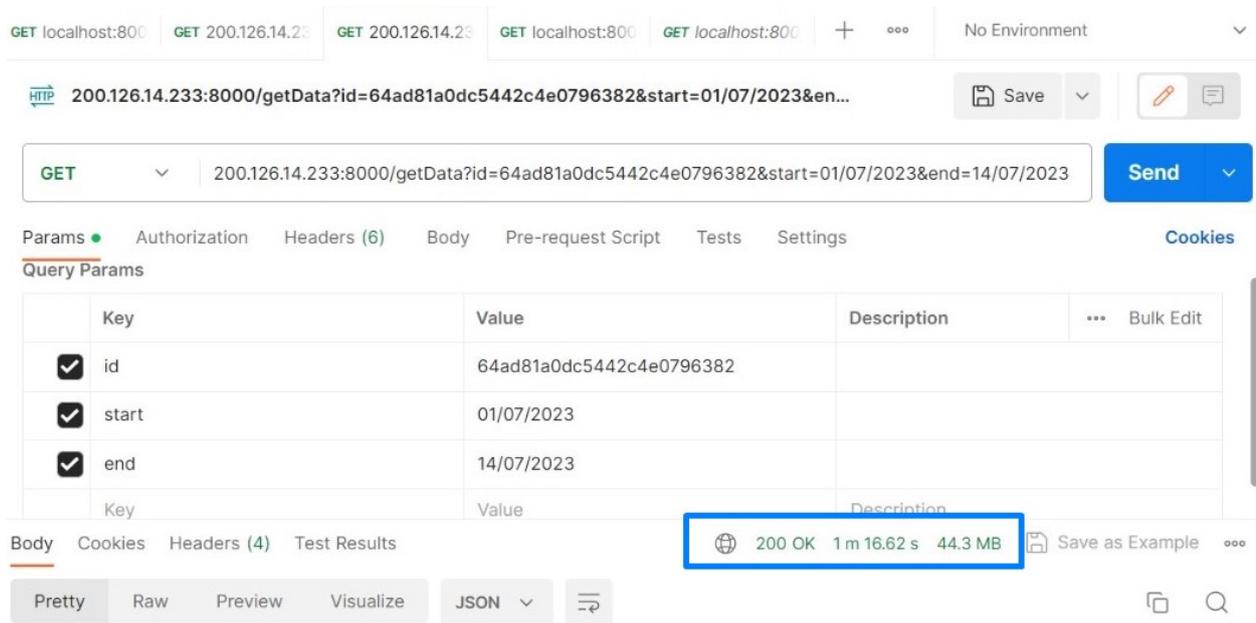


Figura 3.8: Antes de Optimización de Código

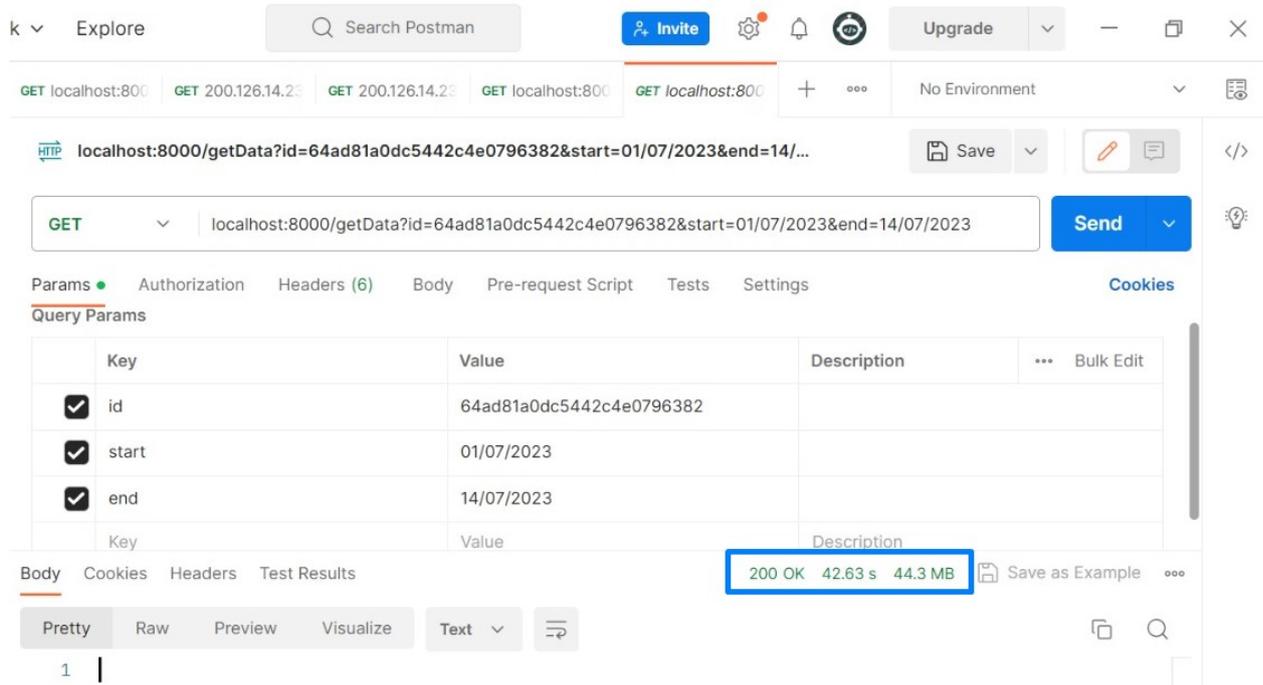


Figura 3.9: Después de Optimización de Código

3.3.3 Análisis de Número de Variables

Con el propósito de evaluar el alcance, se procedió a analizar la influencia del número de variables en cada sensor sobre su tiempo de respuesta, (Fig 3.10). Para lograrlo, se efectuaron solicitudes a sensores que presentaban diferentes cantidades de variables, con el objetivo de observar y comparar sus tiempos de respuesta.

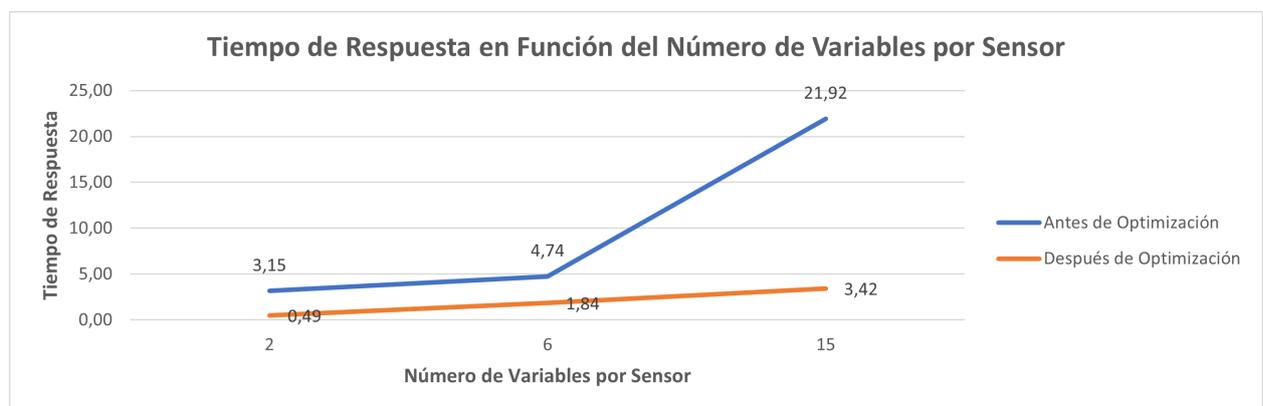


Figura 3.10: Tiempo de Respuesta Según el Número de Variables Previo a la Optimización

Este enfoque proporcionó una visión clara sobre el impacto de dicha variable en la velocidad de respuesta de los dispositivos y permitió determinar cómo la cantidad de

variables afecta el tiempo tanto antes como después de la optimización realizada a la plataforma.

Se observa una relación inversa clara entre la cantidad de variables y el tiempo de respuesta, teniendo sensores de 2, 6 y 15 variables los cuales son el DHT22, Sonoff, y Shelly respectivamente, lo que sugiere que un aumento en la cantidad de variables por sensor se asocia con un incremento en el tiempo de respuesta, esto indica que el procesamiento de una mayor cantidad de datos puede requerir más tiempo. Por otro lado, es posible notar que después de la optimización, los tiempos de respuesta disminuyen significativamente.

3.3.4 Análisis de Tiempo de Respuesta

También, se llevaron a cabo pruebas para evaluar la capacidad de tiempo de respuesta de la plataforma IoT frente a solicitudes simultáneas de lecturas. Estas pruebas se realizaron enviando progresivamente múltiples solicitudes hacia el API, aumentando gradualmente el número de solicitudes de 5 en 5, tal como se observa en la Fig. 3.11.

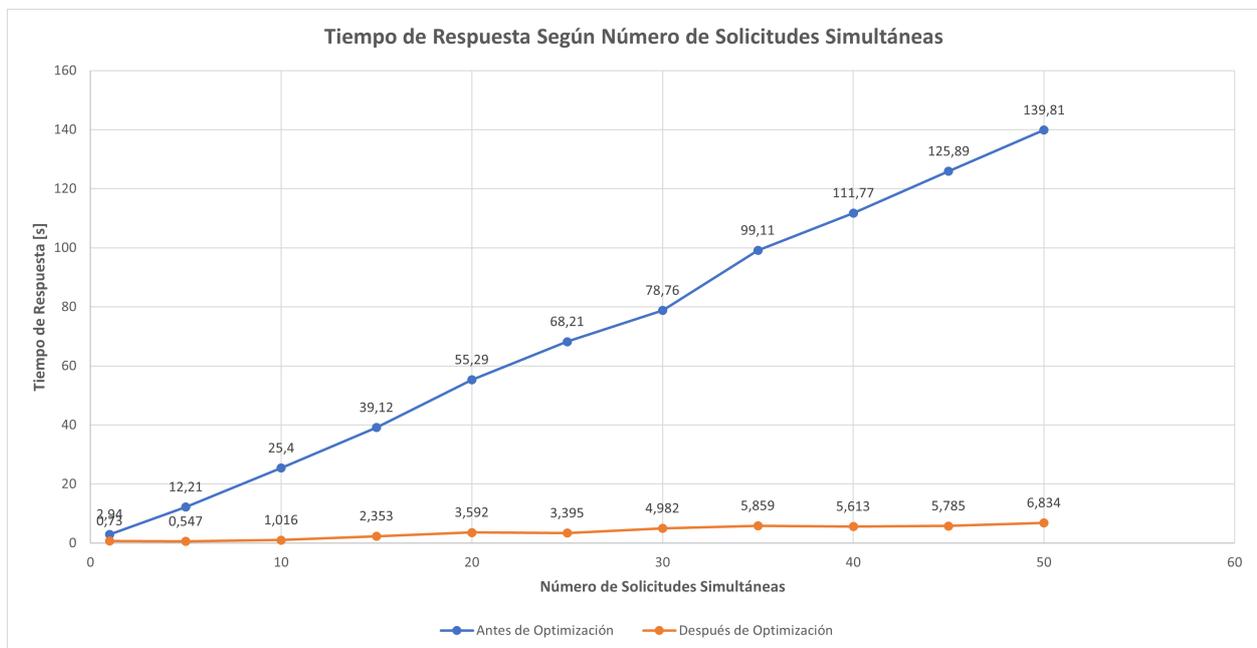


Figura 3.11: Rendimiento Según Número de Solicitudes

Los resultados obtenidos revelaron un aumento significativo en el tiempo promedio de respuesta a medida que aumentaba el número de solicitudes simultáneas previo a la

optimización, estos resultados indican que la plataforma IoT muestra una tendencia a experimentar un incremento en el tiempo de respuesta a medida que la carga de solicitudes aumenta.

Este problema fue solucionado mediante la optimización de la plataforma, donde es posible observar que el tiempo de respuesta según la cantidad de solicitudes disminuyó significativamente, teniendo como tiempo de respuesta máximo 6,83 segundos cuando existen 50 solicitudes simultáneas.

En general, los resultados respaldan la efectividad y utilidad del presente proyecto. La integración exitosa de dispositivos y sensores, junto con la visualización efectiva de datos, valida la capacidad de la plataforma para facilitar la gestión y monitoreo de información en entornos IoT.

CAPÍTULO 4

4. CONCLUSIONES Y LÍNEAS FUTURAS

En este capítulo, se presentan las conclusiones y recomendaciones derivadas del desarrollo de la plataforma de estandarización de Internet de las Cosas HayIoT v2. Además, se proponen nuevas ideas para futuras líneas de investigación que pueden ampliar y mejorar el trabajo realizado hasta ahora.

4.1 Conclusiones

La implementación del API ha sido exitosa para estandarizar los datos provenientes de diferentes sensores, permitiendo una integración óptima y simplificando el acceso a los datos recopilados, esta estandarización ha demostrado ser esencial para facilitar la interoperabilidad y el análisis de datos.

Además, la elección de los protocolos de comunicación ha brindado una flexibilidad significativa al sistema, permitiendo una comunicación efectiva con una amplia variedad de dispositivos IoT, mejorando la adaptabilidad y escalabilidad de la plataforma.

Por otro lado, la interfaz gráfica ha sido diseñada pensando en la experiencia del usuario, lo que ha proporcionado una interfaz intuitiva y amigable para acceder y visualizar los datos, aumentando la facilidad de uso.

Finalmente, se ha logrado la implementación de un sistema escalable, capaz de crecer e incorporar una gran cantidad de dispositivos IoT conforme se necesiten, permitiendo así la visualización de datos en un tiempo relativamente corto.

4.2 Recomendaciones

En base a los resultados y experiencias obtenidas durante el desarrollo de la plataforma, se presentan las siguientes recomendaciones.

Es recomendable asegurar la implementación de mecanismos de seguridad sólidos para proteger la integridad de los datos. La encriptación y autenticación deben ser consideradas para garantizar que los datos sensibles no sean comprometidos.

También, se debe establecer un sistema de monitoreo proactivo para supervisar el estado de los dispositivos y el rendimiento de la plataforma. Asimismo, implementar un plan de mantenimiento periódico para asegurar un funcionamiento óptimo y evitar posibles fallos.

Se recomienda realizar un seguimiento de las visualizaciones en MongoDB, ya que no existen versiones óptimas respecto a series de tiempo.

4.3 Líneas Futuras

Para seguir evolucionando y mejorando la plataforma de estandarización de IoT, se sugieren las siguientes líneas de investigación y desarrollo.

Explorar la incorporación de técnicas de inteligencia artificial y aprendizaje automático para el análisis y detección de patrones en los datos recopilados. Esto permitirá obtener información más profunda y valiosa a partir de los datos generados por los sensores.

Continuar desarrollando nuevas funcionalidades y características en la interfaz gráfica, como paneles personalizados, alertas en tiempo real y visualizaciones avanzadas. Esto permitirá a los usuarios obtener una comprensión más completa y detallada de los datos recolectados.

BIBLIOGRAFÍA

- ¿Qué es una API REST? | IBM — ibm.com. (n.d.).
- Al-Qaseemi, S. A., Almulhim, H. A., Almulhim, M. F., & Chaudhry, S. R. (2016). Iot architecture challenges and issues: Lack of standardization. *2016 Future Technologies Conference (FTC)*, 731–738. <https://doi.org/10.1109/FTC.2016.7821686>
- Banerjee, I. (2017). Chapter two - advances in model-based testing of gui-based software. In A. M. Memon (Ed.). Elsevier. <https://doi.org/https://doi.org/10.1016/bs.adcom.2016.12.003>
- Bressoud, T., & White, D. (2020). The hypertext transfer protocol. In *Introduction to data systems: Building from python* (pp. 609–648). Springer International Publishing. https://doi.org/10.1007/978-3-030-54371-6_20
- Group, P. G. D. (2021). Postgresql.
- Hariri, R. H., Fredericks, E. M., & Bowers, K. M. (2019). Uncertainty in big data analytics: Survey, opportunities, and challenges. *Journal of Big Data*, 6(1), 1–16.
- Haystack. (n.d.). Haystack ontology.
- Jang, H., Choi, S., Kwon, E., & Kwon, C. (2020). Iot device auto-tagging using transformers. *2020 12th International Conference on Advanced Infocomm Technology (ICAIT)*, 47–50. <https://doi.org/10.1109/ICAIT51223.2020.9315384>
- Lee, E., Seo, Y.-D., Oh, S.-R., & Kim, Y.-G. (2021). A survey on standards for interoperability and security in the internet of things. *IEEE Communications Surveys & Tutorials*, 23(2), 1020–1047.
- Lübben, C., & Pahl, M.-O. (2022). Autonomous convergence mechanisms for collaborative crowd-sourced data-modeling. *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, 1–5. <https://doi.org/10.1109/NOMS54207.2022.9789820>
- MongoDB. (2021). MongoDB.
- Morrish, J., & Arnott, M. (n.d.). Global iot forecast report, 2021-2030.
- Nižetić, S., Šolić, P., López-de-Ipiña González-de-Artaza, D., & Patrono, L. (2020). Internet of things (iot): Opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production*, 274, 122877. <https://doi.org/https://doi.org/10.1016/j.jclepro.2020.122877>
- Quinn, C., & McArthur, J. (2021). A case study comparing the completeness and expressiveness of two industry recognized ontologies. *Advanced Engineering Informatics*, 47, 101233.
- Quinn, C., & McArthur, J. (2022). Comparison of brick and project haystack to support smart building applications. *arXiv preprint arXiv:2205.05521*.

- Roy, S. K., Misra, S., & Raghuwanshi, N. S. (2019). Sensnpn: Seamless integration of heterogeneous sensors with iot devices. *IEEE Transactions on Consumer Electronics*, 65(2), 205–214. <https://doi.org/10.1109/TCE.2019.2903351>
- Sonoff.tech. (2021). Sonoff official.
- V, S., A, V., & Pattar, S. (2022). Mqtt based secure transport layer communication for mutual authentication in iot network [International Conference on Intelligent Engineering Approach(ICIEA-2022)]. *Global Transitions Proceedings*, 3(1), 60–66. <https://doi.org/https://doi.org/10.1016/j.gltip.2022.04.015>
- Valov, N., Ivanova, D., & Valova, V. (2020). Increasing energy efficiency of a gas boiler using remote access control. *2020 7th International Conference on Energy Efficiency and Agricultural Engineering (EEAE)*, 1–4. <https://doi.org/10.1109/EEAE49144.2020.9279063>
- Villamil, S., Hernández, C., & Tarazona, G. (2020). An overview of internet of things. *Telkomnika (Telecommunication Computing Electronics and Control)*, 18(5), 2320–2327.
- Viloria, A., Acuña, G. C., Franco, D. J. A., Hernández-Palma, H., Fuentes, J. P., & Rambal, E. P. (2019). Integration of data mining techniques to postgresql database manager system. *Procedia Computer Science*, 155, 575–580.
- Wijethilaka, S., & Liyanage, M. (2021). Realizing internet of things with network slicing: Opportunities and challenges. *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 1–6.
- Zikria, Y. B., Kim, S. W., Hahm, O., Afzal, M. K., & Aalsalem, M. Y. (2019). Internet of things (iot) operating systems management: Opportunities, challenges, and solution. *Sensors*, 19(8), 1793.

APÉNDICES

Apéndice A: Análisis de Costos

Se proporciona un análisis de costos, diferenciando entre los costos de implementación y los costos operativos. Es importante tener en cuenta que la plataforma se comercializará como un servicio.

Tabla 1: Costos Operativos

Descripción	Precio
Seguridad y API Gateway	\$1.300,00
Almacenamiento y Respaldos	\$700,00
Soporte	\$600
Total	\$2.600,00

Tabla 2: Costos de Implementación

Descripción	Precio
Desarrollo	\$3.000,00
Total	\$3.000,00

Los costos de implementación se pagan una única vez, mientras que los costos operativos se pagan con frecuencia anual. Es decir, durante el primer año el costo será de \$5,600, mientras que para los siguientes años será de \$2,600.