Desarrollo de red de comunicación para estación de carga orientado a vehículos eléctricos

Marco de Jesús Maldonado Flores, Román Jared Méndez Barrera, Noé Urbina Villa, M. en C. Alberto Jesús Alcántara Méndez, M. en C. Raúl Santillán Luna
Escuela Superior de Cómputo I.P.N. Ciudad de México, México.
Tel. 57-29-6000 ext. 52000 y 52021.
mmaldonadof1900@alumno.ipn.mx
rmendezb1600@alumno.ipn.mx
mp.noe.urbina@gmail.com

Resumen — **El presente trabajo terminal tiene como objetivo desarrollar una red de comunicación para estaciones de carga de vehículos eléctricos, utilizando el protocolo MQTT en conjunto con un servicio web diseñado para facilitar el monitoreo y gestión de los registros capturados por los sensores de las estaciones. Esta solución está orientada a permitir la administración de la red sin necesidad de conocimientos avanzados en programación. A lo largo del documento se aborda de manera integral la problemática, el análisis, diseño, implementación y las pruebas del sistema. Asimismo, se presentan las interfaces del servicio web, desarrollado con el framework Django, para ejemplificar su funcionamiento. El sistema propuesto proporciona a los administradores un control accesible sobre los parámetros de cada estación, permitiendo además la configuración de reglas específicas según los requerimientos de cada una.**

Palabras clave — **Django, estaciones de carga, monitoreo, MQTT, python, raspberry pi, servicio web**

I. INTRODUCCIÓN

En la última década, ha surgido un crecimiento notable en la adopción de vehículos eléctricos (VE) como respuesta a los esfuerzos mundiales por reducir las emisiones de carbono y enfrentar el cambio climático[1]. Esta transición hacia la movilidad eléctrica se ha visto impulsada por avances tecnológicos en baterías de alto rendimiento, una mayor conciencia ambiental y regulaciones gubernamentales que promueven la electrificación del transporte. Sin embargo, el éxito continuo de los vehículos eléctricos depende en gran medida de la disponibilidad de una infraestructura de carga robusta y accesible. Las estaciones de carga desempeñan un papel esencial al proporcionar a los conductores de vehículos eléctricos la capacidad de recargar sus vehículos de manera conveniente y eficiente. Más allá de la mera disponibilidad de puntos de carga, la eficacia y la inteligencia en la gestión de estas estaciones son fundamentales para garantizar una transición sin problemas hacia una movilidad eléctrica más amplia.

En este contexto, el desarrollo de una red de comunicación para estaciones de carga de vehículos eléctricos adquiere importancia. Esta red facilita la interacción entre el vehículo y la estación de carga, mientras optimiza el rendimiento operativo, gestiona la energía y mejora la experiencia del usuario. En esta introducción, se adentra en el estado actual del desarrollo de estas redes de comunicación para estaciones de carga de vehículos eléctricos. Se analizan los protocolos de comunicación utilizados, las tecnologías emergentes que están transformando el panorama de la carga de vehículos eléctricos, los desafíos técnicos y operativos que enfrentan, así como las oportunidades futuras para mejorar la eficiencia y la sostenibilidad de estas redes. El objetivo es comprender las complejidades de la infraestructura de carga para vehículos eléctricos e identificar áreas de innovación y mejora que impulsen aún más la adopción de la movilidad eléctrica a nivel mundial. Este proyecto busca desempeñar un papel crucial en la construcción de un futuro más limpio, eficiente y sostenible para todos.

II. METODOLOGÍA

A. Metodología espiral

Para el trabajo terminal propuesto, se ha considerado el uso de la metodología Espiral, que combina las metodologías Cascada e Iterativa. En la bibliografía consultada, se divide el proceso en cuatro etapas: conceptualización, desarrollo, mejora y mantenimiento. Cada una de estas etapas se descompone en varias fases, que incluyen planificación, modelado, construcción y despliegue. Un aspecto clave es la fase de transición entre etapas, que facilita el paso de una fase a la siguiente. El equipo de desarrollo comienza con un conjunto reducido de requisitos y avanza a través de cada fase para ese conjunto. Posteriormente, se agrega funcionalidad en iteraciones sucesivas para los requisitos adicionales, hasta que la aplicación esté lista para su lanzamiento.

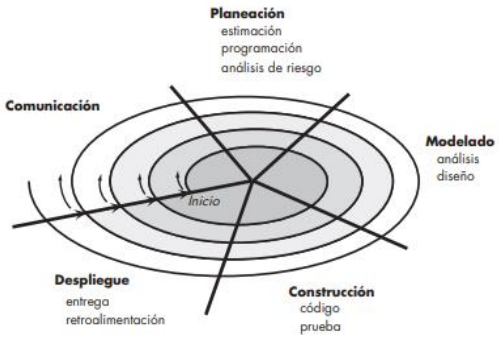

Figura 1. Metodología en espiral

B. Arquitectura del sistema

En la Figura 2 se presenta la arquitectura del sistema, que incluye dos módulos desarrollados en el proyecto: uno para la comunicación MQTT entre el cliente y el servidor, y otro para el servicio web, ambos en una Raspberry Pi 4 Model B[2].

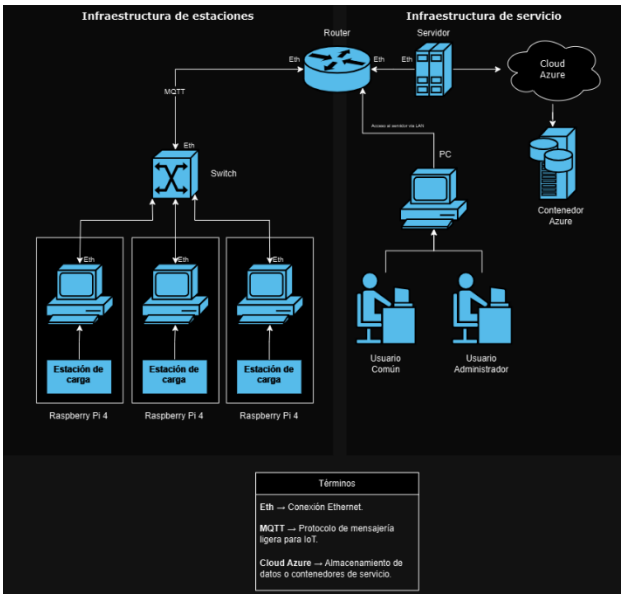

Figura 2. Arquitectura del sistema

1. Módulo de comunicación MQTT
   El módulo de comunicación MQTT realiza un monitoreo constante de la base de datos en MongoDB de las estaciones de carga, reenviando los datos registrados por las estaciones al servidor. Asimismo, se encarga del apagado o encendido remoto de las estaciones cuando detecta que una medición supera los límites establecidos por el usuario administrador.

   Además, este módulo tiene la función de recibir los datos enviados por las estaciones cliente y registrarlos en la base de datos para su posterior análisis.

2. Módulo de servicio web
   El módulo de servicio web permite a los usuarios administradores monitorear las mediciones reportadas por las estaciones y verificar su estado. Además, facilita la revisión de los informes generados cuando las mediciones superan los valores máximos establecidos por las reglas asignadas a cada estación.

   También posibilita la creación y modificación de reglas para nuevas estaciones, funciones que solo puede realizar el usuario administrador, quien debe contar con los conocimientos necesarios para ajustar estos valores sin comprometer la integridad de los componentes de la estación de carga.

   Por último, el módulo incluye una sección de gráficas y tarifas, que presenta los datos de uso de las estaciones de manera visual, permitiendo su análisis e interpretación por parte de los usuarios administradores.

C. Desarrollo del módulo MQTT

El módulo MQTT se desarrolló utilizando Python como lenguaje de programación para establecer las conexiones entre los clientes y el servidor. Para implementar la comunicación mediante el protocolo MQTT, se empleó la biblioteca paho-mqtt[3], y se utilizó el servidor Mosquitto para configurar el bróker en el servidor, el cual “proporciona implementaciones de servidor y cliente conformes con los estándares del protocolo de mensajería MQTT”[4].

Para facilitar la comunicación entre los scripts y la base de datos del servidor en MySQL, se hizo uso del conector MySQL para Python.

Los módulos utilizan una clase personalizada llamada “Estación”, que se encarga de almacenar los registros de mediciones realizadas por la estación de carga y enviarlos al servidor de manera conjunta.

Finalmente, el módulo MQTT genera una señal de apagado cuando se recibe una alerta, ya sea manual o automática. En el caso de la estación, esta señal se representa mediante un LED, dado que no se cuenta con un prototipo funcional de la estación de carga.

D. Desarrollo del módulo de servicio web

Para la implementación del servicio web, se utilizó el framework Django, aprovechando que ya se empleaba Python para el módulo MQTT. Esto facilitó la integración de los scripts de ambos módulos.

El servicio se configuró para adaptarse a la base de datos MySQL, creando los modelos necesarios para garantizar su funcionamiento correcto al insertar o recibir datos. Además, se implementaron transacciones en la creación de nuevos registros para asegurar la integridad y el correcto proceso de inserción.

Finalmente, se definieron dos tipos de usuarios: cliente y administrador. El administrador es el usuario más relevante, ya que es el único que tiene acceso a las interfaces de monitoreo y modificación de datos de las estaciones.

E. Implementación

Se llevaron a cabo pruebas de funcionamiento de ambos módulos para verificar su correcto desempeño. En la Figura 3 se muestra una de las pruebas realizadas para evaluar el funcionamiento del módulo MQTT.

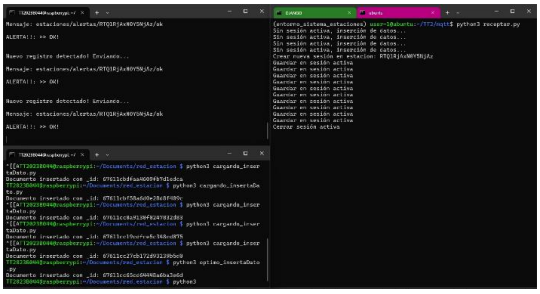

Figura 3. Pruebas del módulo MQTT

III. RESULTADOS

Se utilizaron máquinas virtuales para simular el funcionamiento de las estaciones de carga y transmitir datos al servidor. Para ello, únicamente fue necesario contar con un respaldo de la base de datos en MongoDB, similar al utilizado por las estaciones de carga reales, junto con un script encargado de registrar nuevos datos en la base de manera automatizada.

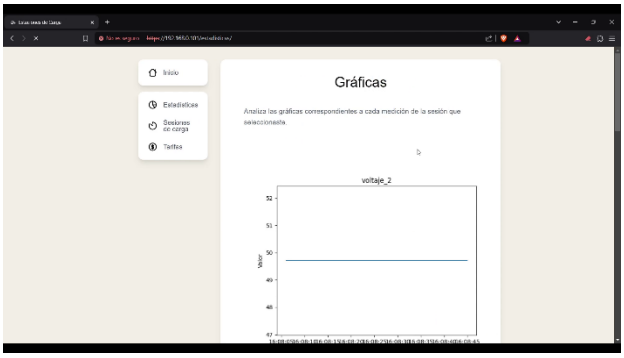

Figura 4. Simulación y transmisión de datos al servidor

Mediante la simulación de los datos, se verificó que la transmisión de los datos del cliente al servidor se realizara correctamente, así como el procesamiento adecuado de las acciones en el backend que dependían de dichos datos. Además, se comprobó la correcta generación de las gráficas por mediciones y la creación de recibos en formato PDF.

IV. CONCLUSIONES

El desarrollo de una red de comunicación para estaciones de carga de vehículos eléctricos representa un avance significativo en la construcción de una infraestructura eficiente, escalable y sostenible para la movilidad eléctrica. A través de la implementación de un sistema robusto, basado en tecnologías como la integración de hardware confiable (Raspberry Pi 4), el protocolo MQTT para comunicación y una base de datos en la nube, se logró consolidar un modelo funcional que cumple con los requerimientos actuales del sector.

El proyecto resolvió problemáticas clave como la interoperabilidad entre estaciones de carga y la optimización de recursos a través de reglas configurables. Estas reglas, entre las que destacan los límites máximos, mínimos y óptimos de carga, permiten adaptar la funcionalidad de las estaciones mientras se promueve la sostenibilidad y se protege la salud de las baterías de los vehículos eléctricos. Estas capacidades aseguran que las estaciones puedan operar con flexibilidad y adaptabilidad según las necesidades del entorno y los usuarios.

V. RECONOCIMIENTOS

Los Autores agradecen a la Escuela Superior de Cómputo del Instituto Politécnico Nacional por el apoyo recibido y las facilidades otorgadas para el desarrollo del presente trabajo terminal.

VII. REFERENCIAS

[1]. L. A. Sánchez Gaspariano y C. I. Martínez Gómez, «Automoción eléctrica en México», RDI, vol. 9, n.º 26, pp. 1–12, may 2023.

[2]. R. P. Ltd, “Buy a Raspberry Pi 4 Model B – Raspberry Pi,” Raspberry Pi. Accedido: nov. 28, 2024. [En línea]. Disponible: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[3]. “client module — Eclipse paho-mqtt documentation”. Projects Gateway | The Eclipse Foundation. Accedido el 23 de diciembre de 2024. [En línea]. Disponible: https://eclipse.dev/paho/files/paho.mqtt.python/html/client.html

[4]. R. A. Light, "Mosquitto: server and client implementation of the MQTT protocol," *Journal of Open Source Software*, vol. 2, no. 13, p. 265, 2017.