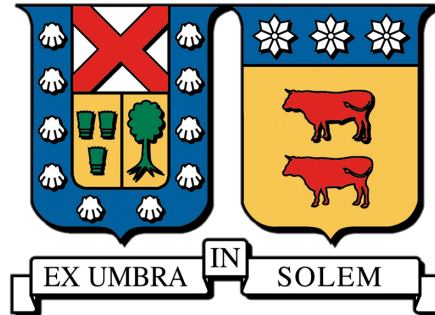


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



**“PLATAFORMA IoT PARA COMPARTIR
ESTACIONES DE CARGA DE VEHÍCULOS
ELÉCTRICOS”**

JAVIER ALONSO MARTÍNEZ FERNÁNDEZ

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO
CIVIL TELEMÁTICO**

PROFESOR GUÍA:

JOSÉ MARTÍNEZ

PROFESOR CORREFERENTE:

MOHAMED ABDELHAMID

MARZO 2026



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título; Tesis de Postgrado;

Título del trabajo: Plataforma IoT para Compartir Estaciones de Carga de Vehículos Eléctricos

Nombre del candidato(a): Javier Alonso Martínez Fernández

Carrera / Grado: Ingeniería Civil Telemática

Campus: Casa Central Valparaíso ; **Departamento:** Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Jose Manuel Martínez Verdugo, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

6 meses; 12 meses; 2 años; 3 años; 5 años; 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 12/03/2026 ; **Firma:** 

Estudiante o Candidato(a):

Fecha: 06/03/2026 ; **Firma:** 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradecimientos

Quiero agradecer a todas las personas que me apoyaron en mi etapa estudiantil, desde mis profesores jefes en el colegio hasta mi profesor guía José Manuel Martínez y profesor correferente Mohamed Abdelhamid, que me ayudaron a desarrollar el pensamiento crítico necesario para abordar el desafío. Y a Francisca Soto, que me ayudó en un principio a evaluar distintas alternativas para abordar el tema, desde la factibilidad técnica y el impacto que tiene este proyecto.

A mi familia que me apoyaron desde el principio y facilitaron que me fuera a estudiar a otra ciudad, especialmente a mi mamá que me apoyó siempre, con las llamadas de teléfono más largas cuidándome y aconsejándome a la distancia, también a mi papá, que me ayudó a forjar carácter y motivándome a siempre seguir adelante sin bajar los brazos.

A mis compañeros de clases, con los que pasé la mayoría del tiempo de esta etapa, tanto en las salas de clases como en otras ocasiones más informales. Gracias a ellos no me sentí solo y me sentí cómodo fuera de mi casa. Especialmente a Enrique, con el que siempre hice grupo de trabajo y nos apoyamos mutuamente en la vida universitaria y personal, no pude haber tenido un mejor compañero.

Finalmente, a Arantza, quien fue un pilar fundamental en estos últimos años, acompañándome los días buenos y malos. Siempre estando junto a mí, en los momentos buenos y no tan buenos. Me hiciste sentir seguridad y confianza cuando más lo necesitaba.

Resumen

El crecimiento sostenido del mercado de vehículos eléctricos ha generado una creciente demanda por infraestructura de carga, la cual en muchos países, incluido Chile, aún resulta insuficiente para cubrir las necesidades de los usuarios. La expansión de estaciones de carga públicas implica altos costos de instalación, mantenimiento y adecuación a las condiciones del sistema eléctrico, lo que ralentiza su despliegue. En este contexto, esta memoria propone el desarrollo de una plataforma basada en Internet de las Cosas (IoT) destinada a facilitar el uso compartido de estaciones de carga privadas, permitiendo ampliar la oferta disponible mediante un modelo colaborativo.

La solución integra dispositivos embebidos, sensores y servicios web para monitorear en tiempo real el estado de las estaciones y su disponibilidad, junto con una aplicación que permite a los usuarios localizar, reservar y utilizar estaciones de carga de manera eficiente. El sistema contempla una arquitectura modular compuesta por capas de percepción, red, servicio, aplicación y negocio, además de mecanismos de autenticación y gestión de reservas para garantizar seguridad y trazabilidad. Asimismo, se incorpora un sistema de recomendación que considera factores como tiempo de carga y preferencias del usuario para optimizar la asignación de recursos.

Los resultados muestran la viabilidad de la plataforma para mejorar el acceso a infraestructura de carga, promoviendo un modelo distribuido y colaborativo que contribuye a reducir las limitaciones actuales en la disponibilidad de estaciones y favorece el desarrollo de la electromovilidad.

***Palabras Clave:* Vehículos eléctricos, Estaciones de Carga, Internet of Things (IoT), Infraestructura de carga inteligente, Estaciones de carga compartidas.**

Abstract

The sustained growth of the electric vehicle market has generated increasing demand for charging infrastructure, which in many countries, including Chile, is still insufficient to meet user needs. The expansion of public charging stations involves high costs for installation, maintenance, and adaptation to the conditions of the electrical system, which slows down their deployment. In this context, this report proposes the development of a platform based on the Internet of Things (IoT) designed to facilitate the shared use of private charging stations, allowing the available supply to be expanded through a collaborative model.

The solution integrates embedded devices, sensors, and web services to monitor the status and availability of stations in real time, along with an application that allows users to locate, reserve, and use charging stations efficiently. The system features a modular architecture composed of perception, network, service, application, and business layers, as well as authentication and reservation management mechanisms to ensure security and traceability. It also incorporates a recommendation system that considers factors such as charging time and user preferences to optimize resource allocation.

The results show the platform's viability in improving access to charging infrastructure, promoting a distributed and collaborative model that helps reduce current limitations in station availability and encourages the development of electromobility.

***Keywords:* Electric Vehicles, Charging Stations, Internet of Things (IoT), Smart Charging Infrastructure, Charging Station Sharing.**

Índice de figuras

2.1. Cantidad de Cargadores Públicos y Privados, SEC.	12
2.2. Distribución de cargadores a nivel nacional, SEC	13
2.3. Asociaciones comerciales de WEcharge.	14
2.4. Flujo de trabajo del plan BBC propuesto.	15
2.5. Componentes del sistema.	16
2.6. Distribución de las categorías.	17
2.7. Modelo funcional de <i>Smart Wireless EV charging station</i> usando una Raspberry Pi.	18
2.8. Arquitectura SIMA.	19
2.9. Esquema propuesto.	19
2.10. Conexión CPO-SCSP y arquitectura EMS.	20
2.11. Arquitectura proupuesta.	21
2.12. Un esquema de los componentes de un sistema V2G y las comunica- ciones de la red de flujo de energía referenciado en.	22
2.13. Integración de energía renovable e IoT para la transferencia inalámbrica avanzada de energía en EV.	23
2.14. Ocho opciones de flujo energético entre las que puede elegir la gestión energética (SPEM no ocupa las rojas).	24
2.15. Diagrama de flujo.	25
2.16. Arquitectura de sistema blockchain.	26
2.17. Flujo del proceso de carga.	27

2.18. Flujo operacional del sistema de cobranza.	28
2.19. Arquitectura general del sistema online de detección de intrusiones propuesto.	29
2.20. Estructura del modelo GAN-DL propuesto.	30
3.1. Esquema general del sistema	34
3.2. Arquitectura del sistema y relación entre módulos	35
3.3. Vista sencilla de usuarios y sus necesidades.	42
3.4. Modelo de navegación del FrontEnd de la plataforma IoT dividido en secciones, la sección violeta representa la interfaz disponible para los usuarios dueños de vehículos eléctricos (EV), la sección verde la interfaz disponible para el administrador de estaciones de carga (CS) y por último la sección naranja que representa lo posible para el usuario administrador de la aplicación.	43
4.1. Diagrama explicativo del funcionamiento general de las tecnologías utilizadas en el proyecto con sus respectivas capas asociadas.	46
4.2. Esquema de la Base de datos con las colecciones explicadas con anterioridad, se presentan sus relaciones.	50
4.3. Inicio de sesión en la plataforma.	59
4.4. Formulario de registro.	60
4.5. Modificación de perfil del usuario.	61
4.6. Funcionamiento de panel de notificaciones.	63
4.7. Tablero de administración general.	66
4.8. Funcionamiento de la gestión detallada del administrador de la aplicación.	68
4.9. Vista general del panel de estaciones de carga.	70
4.10. Funcionalidad de tarjetas de cargador.	72
4.11. Funcionalidades del buscador de cargadores.	75
4.12. Mapa interactivo de cargadores con marcadores de ubicación.	76

4.13. Calendario de planificación de reservas de estaciones de cargadores de vehículos eléctricos de estaciones de carga.	78
4.14. Historial de sesiones de carga con detalles de vehículos, fechas, energía y duración.	79
4.15. Gráficas interactivas de ocupación y consumo energético de la estación de carga.	81
4.16. Pantalla de reservas.	83
4.17. Estación seleccionada en el mapa.	84
4.18. Opciones disponibles para la reserva de estación desde el modal.	86
4.19. Preferencias disponibles en la opción de reserva rápida.	87
4.20. Panel derecho en la vista de ocupante EV.	88
4.21. Historial de carga de vehículo seleccionado.	89
4.22. Calendario de reserva por estación.	90
4.23. Diagrama de flujo para la recomendación de reserva.	92

Índice de tablas

2.1. Tabla comparativa entre el estado del arte y solución propuesta.	31
---	----

Índice general

1. Introducción	5
1.1. Contexto	5
1.2. Problema a resolver	7
1.3. Acercamiento a la solución	7
1.4. Objetivos	8
1.4.1. Objetivo General	8
1.4.2. Objetivos Específicos	8
1.5. Estructura	9
2. Estado del Arte	11
2.1. Actualidad en Chile	11
2.2. Contexto de infraestructura y democratización del acceso	13
2.3. Plataformas IoT y sistemas de monitoreo de estaciones	16
2.4. Arquitecturas de carga inteligente e interoperabilidad	20
2.5. Gestión energética, optimización y energías renovables	22
2.6. Seguridad, autenticación y resiliencia de la infraestructura	25
3. Desarrollo de la plataforma	33
3.1. Diseño de Arquitectura del Sistema	33
3.1.1. Esquema General del Sistema	33
3.1.2. Arquitectura del Sistema	35
3.1.3. Descripción de Módulos	36

3.2. Módulos del Sistema	38
3.2.1. M1: Capa de Percepción	38
3.2.2. M2: Capa de Red	38
3.2.3. M3: Capa de Servicio	39
3.2.4. M4: Capa de Aplicación	40
3.2.5. M5: Capa de Negocio	41
3.3. Diseño de Interfaces	42
3.3.1. Modelo de Navegación	42
4. Implementación	45
4.1. Implementación de Base de datos	45
4.2. Implementación de API	51
4.3. Implementación de Interfaz gráfica	57
4.4. Algoritmo de recomendación	91
4.4.1. Open Source Routing Machine (OSRM)	92
4.4.2. Performance	94
5. Resultados	97
5.1. Despliegue de la plataforma	97
5.1.1. Frontend	97
5.1.2. Backend	98
6. Conclusiones	99

Capítulo 1

Introducción

1.1. Contexto

El mundo está sufriendo un cambio drástico con respecto a la energía, con una conciencia ambiental mayor que impulsa el uso de energías renovables. Por lo que se está buscando un reemplazo o alternativa a los combustibles fósiles. Es por esta razón que los fabricantes de automóviles empezaron a diseñar modelos híbridos, que ocupan tanto combustibles fósiles como electricidad para poder funcionar, y totalmente eléctricos [1].

La venta de automóviles eléctricos (EV) en Chile ha ido en aumento año tras año de forma exponencial [2], por lo que se espera que para 2035 se vendan exclusivamente automóviles eléctricos. Esto implica que también tiene que haber una disponibilidad de electrolinerías suficiente para la demanda de carga por parte de los vehículos. Este año 2024, se vendieron un total de 4.500 unidades en Chile, correspondientes a la mitad de las unidades en circulación.

Según ANAC [3], actualmente hay un total de 1.240 cargadores para autos eléctricos livianos y medianos. Siendo 336 unidades destinadas a la carga pública, mientras que 904 para carga privada. La diferencia entre una estación de carga pública y una privada, es el acceso al público que estas poseen. La instalación de estaciones de carga

es un proceso lento y costoso, no solo en temas de costo de instalación/mantenimiento [4] sino también de un tema energético [5].

Este último tema es crucial de evaluar, al momento de realizar una instalación de una estación de carga, ya que tiene que adecuarse a las condiciones que proporciona el sistema eléctrico de la localidad. La tensión en el sistema eléctrico acá en Chile es de 220 voltios con un amperaje variable entre 10, 16 o 25 Amperes[6, 7].

Con respecto a las estaciones de carga (CS), hay diversos modelos [8, 9, 10] que varían en la potencia desde los 2,2, 7, 22 y hasta 400 kW, por lo que hay muchas opciones de potencia con las que trabajar; esto afecta a la velocidad de carga y la energía que necesita para poder funcionar eficazmente.

Además de las distintas potencias con las que trabajan las estaciones de carga. Existen 4 modos en los que se se pueden conectar los EV al sistema eléctrico. El modo 1 es una conexión a una toma de la corriente alterna (AC) de una instalación eléctrica existente, como lo puede ser un enchufe en una casa, esta corriente se transforma en corriente directa (DC) con un inversor AC/DC que está dentro del vehículo; por la incertidumbre de la instalación eléctrica, es el modo menos recomendable y está limitada la potencia que puede alcanzar la carga; alcanza hasta los 3.5 kW. El modo 2, se puede diferenciar del modo 1, ya que se incorpora un sistema de seguridad, este comprueba que haya toma a tierra, posee protecciones y tiene la posibilidad de seleccionar la velocidad de carga, este también se enchufa a una instalación AC y el vehículo se incorpora al control de la carga; tiene una potencia hasta 2.2 kW monofásico o 11 kW trifásico. Por otro lado, el modo 3 se conecta mediante un cable dedicado, a un equipo que es el encargado de controlar de forma inteligente la carga y la seguridad en el proceso, este modo también se conecta a AC; alcanza los 3.5 kW monofásico y 44 kW trifásico. Mientras que el modo 4 se conecta a una fuente DC, el cargador tiene un rectificador AC/DC, por lo tanto puede alcanzar potencias sobre 40 kW y poseen un alto nivel de control de carga; este tipo de instalaciones son las más costosas.

1.2. Problema a resolver

La actual oferta de estaciones de carga **no será suficiente** para satisfacer el ritmo de crecimiento en la demanda de vehículos eléctricos. En Chile, la mayoría de los hogares cuenta con una red eléctrica monofásica de 220 Volts y un máximo de 25 Amperes, lo que permite una potencia máxima teórica de 5.5 kW. Sin embargo, para evitar sobrecargas o cortes frecuentes, las instalaciones domésticas suelen limitarse a potencias cercanas a los 2.2 kW. Este tipo de carga lenta implica que una recarga completa (de 0 % a 100 %) puede tardar varias horas, lo que hace necesario contar con mecanismos de **planificación y gestión del tiempo de carga** para no afectar la experiencia del usuario.

1.3. Acercamiento a la solución

Con el objetivo de mitigar la limitada disponibilidad de estaciones de carga, se propone una solución basada en la transformación de estaciones de carga privadas en puntos de acceso público. Esta conversión se gestionará mediante una plataforma de *Internet of Things* (IoT), que permitirá a los propietarios compartir sus estaciones con usuarios externos, ampliando así la oferta disponible y sorteando las restricciones actuales del sistema.

La implementación de esta plataforma requerirá diversos componentes tecnológicos, entre ellos microcontroladores, sensores y otros dispositivos embebidos, así como servicios web, bases de datos, servidores y aplicaciones móviles. Estos elementos se integrarán para conformar una arquitectura robusta, escalable y funcional.

Como funcionalidad principal, se busca un sistema de emparejamiento, que sirva tanto a los usuarios que necesiten cargar su vehículo como a dueños o administradores de estaciones de carga, que les muestre ofertas de estaciones de carga *ad hoc* a las preferencias y necesidades del EV. Esto consideraría la demora de la carga como el costo de uso.

1.4. Objetivos

1.4.1. Objetivo General

Implementar una plataforma IoT para compartir estaciones de carga para vehículos eléctricos junto con un sistema de reservas para estas mismas.

1.4.2. Objetivos Específicos

- **Monitorear las CSs y EVs:** Monitorear las estaciones de carga (CS) y los vehículos eléctricos (EV) en tiempo real, utilizando sensores de corriente para registrar el flujo de carga y sensores de proximidad para detectar la ocupación de la estación.
- **Desarrollar un sistema de reservas de las estaciones:** Desarrollar un sistema de reservas para las estaciones de carga, considerando los distintos casos de uso y posibles usos indebidos, con el objetivo de garantizar eficiencia y equidad en la asignación de recursos.
- **Incentivar el compartir las estaciones privadas:** Diseñar un sistema de incentivos para motivar a los propietarios de estaciones privadas a compartir sus estaciones, considerando aspectos de seguridad, privacidad y beneficios económicos u operacionales.
- **Levantar servidor para el despliegue de la aplicación:** Implementar la infraestructura necesaria para desplegar la aplicación, evaluando servicios de servidor en la nube o propios, con el fin de garantizar la transmisión de datos y la conexión entre los distintos módulos del sistema.
- **Integrar con métodos de autenticación el control de acceso:** Desarrollar un sistema de autenticación para validar la identidad de los usuarios al momento de usar o reservar una estación de carga, garantizando trazabilidad y seguridad en el acceso.

- **Transmitir los datos recolectados al servidor:** Se transmiten los datos de forma segura y con una tecnología de comunicación establecida, asegurando un correcto funcionamiento y protección de la información.

1.5. Estructura

Este documento se encuentra constituido de la siguiente manera:

- **Capítulo 1 – Introducción:** Se presenta la motivación del trabajo, el planteamiento del problema, los objetivos generales y específicos, los alcances de la memoria y la organización del documento.
- **Capítulo 2 – Estado del Arte:** Se revisan y analizan trabajos previos, soluciones existentes y antecedentes técnicos relevantes relacionados con la problemática abordada, estableciendo el contexto teórico y tecnológico del proyecto.
- **Capítulo 3 – Modelo Propuesto:** Se describe el modelo conceptual y técnico desarrollado para la solución del problema, detallando su arquitectura, componentes principales y criterios de diseño utilizados.
- **Capítulo 4 – Implementación:** Se expone el proceso de implementación del modelo propuesto, incluyendo las herramientas utilizadas, configuraciones realizadas y consideraciones técnicas relevantes.
- **Capítulo 5 – Resultados:** Se presentan los resultados obtenidos a partir de la implementación, junto con su análisis y evaluación en función de los objetivos planteados.
- **Capítulo 6 – Conclusiones:** Se presentan las conclusiones generales del trabajo, las limitaciones identificadas y posibles líneas de trabajo futuro derivadas del proyecto desarrollado.

Capítulo 2

Estado del Arte

2.1. Actualidad en Chile

En Chile, la compra de automóvil ha ido aumentando en los últimos años. Ya que es el medio más cómodo para transportarse, porque reduce los tiempos de viaje, te acompaña con distintos servicios (música, radio, Waze, entre otros) y poseen más comodidades como lo son el no compartir el espacio con gente desconocida (seguridad), aire acondicionado, asientos cómodos. Mientras se tenga el dinero para adquirir un automóvil, este va a ser la preferencia con respecto al transporte público o los medios de transporte más ligeros, como la bicicleta o scooter.

Junto con esto, la sociedad ha empezado a promocionar el uso de energías renovables o eco-friendly, por lo que el negocio de los automóviles eléctricos ha estado en ascenso. Durante el año 2024 hubo un incremento del 84 % en la venta de este tipo de modelos con respecto al año anterior, alcanzando así un 6.2 % del total de autos vendidos [2].

Acompañando el crecimiento de dueños de automóviles eléctricos, tiene que venir una mejora en la infraestructura tanto pública como privada para el uso de estos mismos. A diferencia de los automóviles que funcionan con combustible fósil, los autos híbridos o totalmente eléctricos requieren estaciones de carga especializadas para po-

der funcionar. En Chile se ha tomado con conciencia este tema y hay varias empresas que han invertido en nuevas estaciones de carga como lo son COPEC [11], o con su nuevo ingreso al mercado chileno, Tesla también se ha encargado de poner estaciones de carga rápida, llamadas "superchargers" [12].

Gracias a [13] y SEC se puede saber la cantidad de estaciones de carga que se han instalado en el país. Un dato relevante es que más de la mitad de las estaciones son de carácter privado, por lo que no pueden ser utilizadas por el público en general. Lo que abre las puertas a que se implemente una metodología/tecnología/negocio para que estas estaciones ayuden a suplir la demanda de por parte de los EVs.

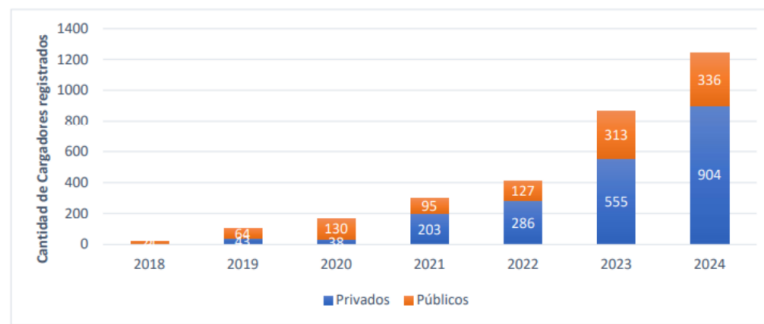


Figura 2.1: Cantidad de Cargadores Públicos y Privados, SEC.

Además, se sabe que los dueños de vehículos eléctricos encuentran un problema al momento de viajar grandes distancias, por la poca disponibilidad de cargadores fuera de la ciudad. Esto implica que tiene que haber una planificación casi minuciosa de donde se puede parar a cargar y que tipo de cargador te vas encontrar (en términos de potencia). Esto hace que no sea trivial salir a vacacionar o explorar con un vehículo eléctrico. Ahora bien, el que haya una infraestructura, no quiere decir que sea de fácil acceso a los distintos usuarios. Si bien hay una aplicación de parte del gobierno de Chile [14], que está dirigida para los usuarios locales, no tiene buena calificación y está con malas reseñas en las distintas tiendas (Play Store y AppStore). Además de que está enfocada en encontrar las estaciones de carga y su disponibilidad, por lo que está dirigida al usuario dueño/ocupante de un automóvil eléctrico, por lo que aún falta una aplicación dirigida a los dueños de estaciones de carga. Hay servicios que se encargan

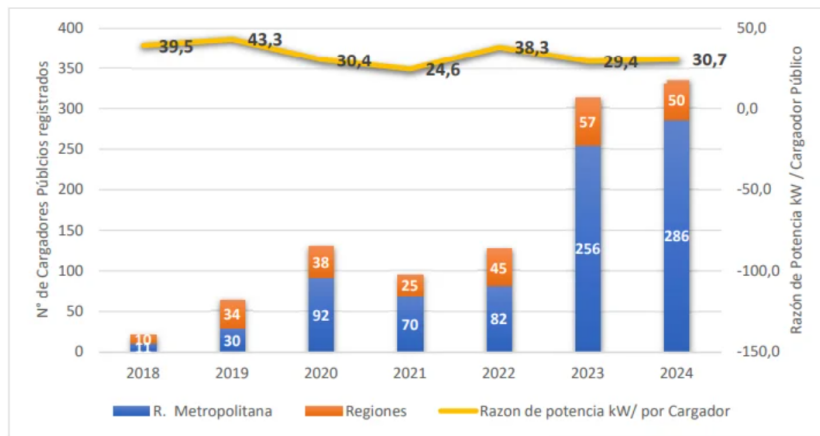


Figura 2.2: Distribución de cargadores a nivel nacional, SEC .

de esto en otros países como Brasil [15], por lo que es una buena oportunidad para instalar un modelo de negocio que pueda beneficiar a ambos tipos de usuario acá en Chile.

2.2. Contexto de infraestructura y democratización del acceso

El crecimiento acelerado en la venta y circulación de EV ha generado una presión significativa sobre la infraestructura de carga disponible. La adopción de esta nueva movilidad, viene impulsada por una mentalidad medio-ambientalista donde se busca reducir la huella de carbono y el uso de combustibles fósiles como lo son los automóviles a gasolina (ICEV). Pero esto tiene que venir acompañado de la infraestructura necesaria para soportar este cambio. En este caso corresponde a la disponibilidad de CS para que se puedan cargar los EV.

En este contexto, la literatura reciente ha comenzado a replantear el concepto tradicional de estación de carga, proponiendo modelos donde la infraestructura no es exclusivamente pública, sino un recurso distribuido y compartido entre usuarios. El artículo [16] propone una plataforma orientada a la democratización de la infraestructura de

carga, permitiendo que propietarios de cargadores privados ofrezcan sus estaciones a terceros mediante mecanismos de descubrimiento, reserva y control de acceso. El aporte central del trabajo es demostrar que una parte relevante de la capacidad instalada permanece subutilizada, y que su integración al ecosistema público puede aumentar la cobertura espacial sin requerir grandes inversiones adicionales en infraestructura física. No obstante, el enfoque depende fuertemente de la participación voluntaria de los usuarios y de la existencia de incentivos adecuados que motiven la compartición.

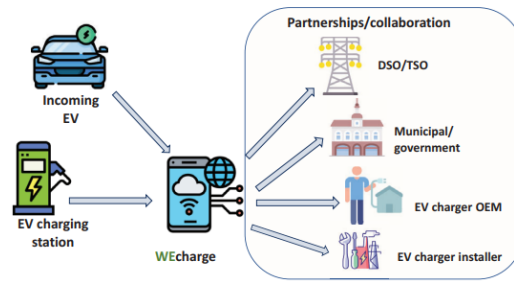


Figura 2.3: Asociaciones comerciales de WEcharge.

Además de mostrar este nuevo modelo de negocio, también aporta con una fórmula que se utiliza para el emparejamiento entre EV y CS según sus características. Este emparejamiento lo utilizar para reservar la estación de carga y medir la utilización de la misma. En la ecuación 2.1, se busca una estación de carga con mejor rendimiento. entre todas, tomando en cuenta la distancia entre ella y el EV, el tiempo que se demora en cargar, el tiempo de espera para que esté disponible y el costo de la energía respectivamente. Todos estos parámetros se normalizan y se ponderan según pesos que son ajustables.

$$P_i = \left(\frac{w_1}{\bar{w}} \frac{d_i}{\max_{i \in \{1, \dots, N\}} d_i} + \frac{w_2}{\bar{w}} \frac{t_i}{\max_{i \in \{1, \dots, N\}} t_i} + \frac{w_3}{\bar{w}} \frac{s_i}{\max_{i \in \{1, \dots, N\}} s_i} + \frac{w_4}{\bar{w}} \frac{c_i}{\max_{i \in \{1, \dots, N\}} c_i} \right) \quad (2.1)$$

Sin embargo, la compartición de infraestructura introduce desafíos asociados a la

confianza y la cooperación entre actores que no necesariamente se conocen. En este sentido, el artículo [17] aborda el problema desde una perspectiva de redes vehiculares cooperativas, proponiendo un esquema de *charging pile sharing* soportado por blockchain. La contribución principal consiste en incorporar mecanismos de autenticación, reputación y registro inmutable de transacciones que permiten a los propietarios ofrecer sus recursos con mayores garantías de seguridad. Así se puede corroborar la autenticación del usuario no solamente por la estación que esté utilizando sino también por otras pertenecientes a esta red como por otros usuarios. La eficacia de este método se verifica dentro de la publicación, logrando transacciones íntegras. Por otra parte, no asegura que sea un sistema eficiente bajo un alto flujo de vehículos y no asegura escalabilidad ni eficiencia con respecto a nuevos escenarios.

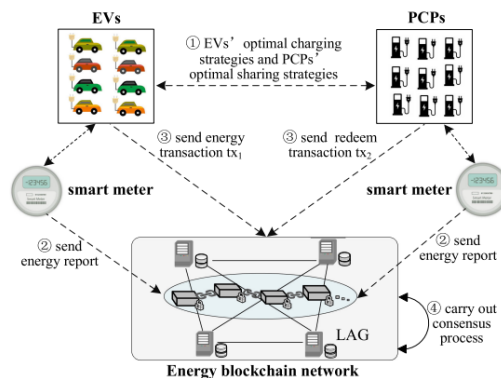


Figura 2.4: Flujo de trabajo del plan BBC propuesto.

Complementariamente, el artículo [18] amplía el paradigma de compartición hacia un modelo completamente *peer-to-peer* (P2P), donde cualquier usuario puede actuar tanto como proveedor como consumidor de energía. La propuesta integra autenticación soportada por *edge computing* y un sistema de facturación basado en criptomonedas, eliminando la dependencia de operadores centrales. Su aporte al contexto de infraestructura es conceptual y operativo, al redefinir la estación de carga como un nodo dinámico dentro de una red distribuida. Esto permite escalar la infraestructura de forma orgánica conforme aumenta la adopción de EV. Lo principal, es que con este modelo las

transacciones son rápidas entre el EV y la CS, ya que se realizaría de forma automática después de generar el bloque para la *blockchain*. Sin embargo, el enfoque asume la existencia de capacidades de cómputo en el borde y aceptación de esquemas de pago no tradicionales, lo que puede limitar su adopción en escenarios reales con restricciones regulatorias o tecnológicas.

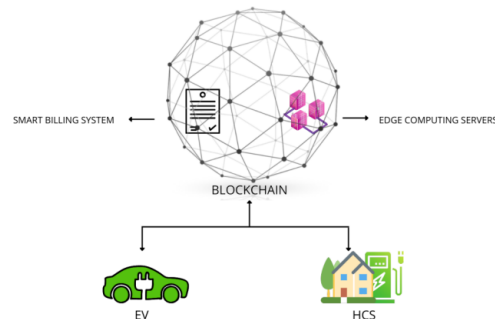


Figura 2.5: Componentes del sistema.

En conjunto, estos trabajos muestran una transición desde modelos centralizados hacia ecosistemas colaborativos donde la infraestructura de carga es concebida como un recurso distribuido. La democratización del acceso permite ampliar la cobertura, reducir la ansiedad de autonomía y disminuir los costos de expansión, aunque introduce nuevos desafíos relacionados con incentivos, confianza, escalabilidad y aceptación por parte de los usuarios. Estos elementos establecen el marco sobre el cual se desarrollan soluciones más integradas para la gestión moderna de estaciones de carga EV.

2.3. Plataformas IoT y sistemas de monitoreo de estaciones

La operación eficiente de una infraestructura de carga para vehículos eléctricos depende cada vez más de la integración de plataformas IoT que permitan supervisar, controlar y optimizar el funcionamiento de las estaciones en tiempo real. Estas plataformas habilitan la recolección continua de datos desde sensores, cargadores y dispositivos de

usuario, facilitando la visualización del estado del sistema, la gestión remota y la mejora de la experiencia del usuario. En este contexto, los sistemas IoT no solo soportan la conectividad, sino que se convierten en un componente central para la observabilidad y la automatización de la red de carga.

El artículo [19] propone una plataforma IoT orientada a mejorar las estaciones de carga mediante la integración de sensores, servicios de movilidad y una aplicación para el usuario final. Su principal aporte es la construcción de un ecosistema donde la infraestructura física se conecta con una capa digital que permite monitorear disponibilidad, consumo energético y estado de las estaciones en tiempo real. Los autores muestran la cantidad de aplicaciones móviles/web que están relacionadas con la electromovilidad en general, y evalúan la posibilidad de migrar algunas de estas aplicaciones a las propias CS. Con esto se puede obtener una mejor experiencia de usuario y aprovechamiento de la capacidad de cómputo que tienen las estaciones. Esto agilizaría todo el proceso de carga, tanto su configuración como el proceso de pago. Sin embargo, esto rompería con el estándar actual y puede que haya un proceso de adaptación grande por parte de los fabricantes y administradores de estaciones de carga.

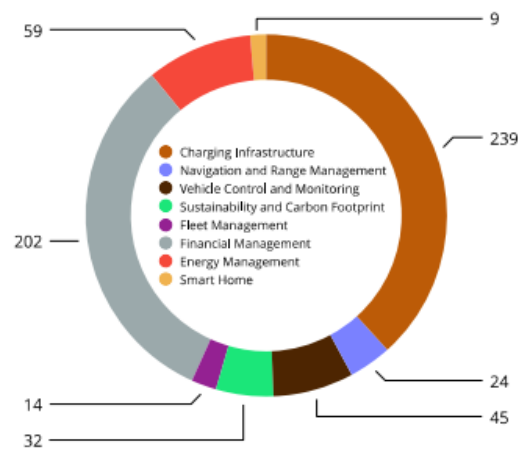


Figura 2.6: Distribución de las categorías.

Desde una perspectiva más prototípica, el artículo [20] presenta el diseño de una estación de carga inalámbrica inteligente basada en Raspberry Pi con monitoreo web. El aporte principal radica en demostrar cómo hardware de bajo costo puede integrarse en

una plataforma IoT para controlar el proceso de carga, recolectar datos operacionales y visualizarlos mediante una interfaz web. Este enfoque aborda otra forma de monitorear la estación de carga, no tomando únicamente las variables relacionadas a la carga, sino también la temperatura y la humedad. Estas últimas son variables que pueden ser útiles para observar las condiciones en la que opera la estación o prevenir algún tipo de accidente/incidente. No obstante, al tratarse de un prototipo, el trabajo no aborda en profundidad la integración con redes de estaciones a gran escala ni los desafíos de interoperabilidad entre múltiples dispositivos.

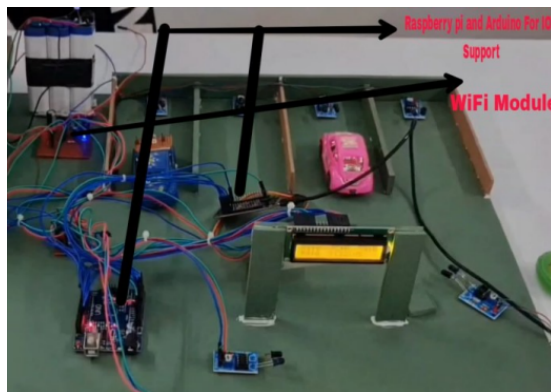


Figura 2.7: Modelo funcional de *Smart Wireless EV charging station* usando una Raspberry Pi.

El artículo [21] desarrolla una plataforma IoT para monitorear continuamente estaciones de carga. Esta plataforma obtiene los datos del protocolo OCPP de la estación para ver sus datos y transacciones. Se plantea IoT como una solución a la complejidad que lleva administrar cada una de las estaciones de carga y más cuando estas varían en tecnología/manufactura. Para probar, tomaron el caso de uso un sistema llamado SIMA, el cual está compuesto por estaciones de carga, buses eléctricos, una plataforma IoT, un back-end OCPP y front-end web. Es una investigación pertinente al desafío que se busca resolver, ya que demuestra que teniendo acceso a OCPP, se pueden monitorear sin problema las estaciones de carga, despreciando si es que la plataforma IoT pueda tener conectividad con el servidor principal.

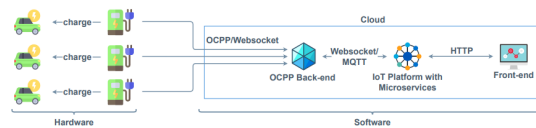


Figura 2.8: Arquitectura SIMA.

Por otra parte, el artículo [22] propone un sistema inteligente basado en dispositivos IoT para el monitoreo y control de la carga residencial de vehículos eléctricos. Aquí van más allá del monitoreo de la CS, sino que implementan un sistema capaz de controlar el flujo de la carga, para así optimizar la carga. Esto puede tener en consideración el gasto energético que está ocurriendo en la casa, luz del día (en caso de tener energía solar) u otras variables ambientales. A esto le llama *smart charging*. Un siguiente paso en la carga de vehículos eléctricos, algo más controlado y eficiente. Lamentablemente no lo implementan con una estación de carga, sino que con otros electrodomésticos que consumen mucha energía como lo es el aire acondicionado y simulaciones.

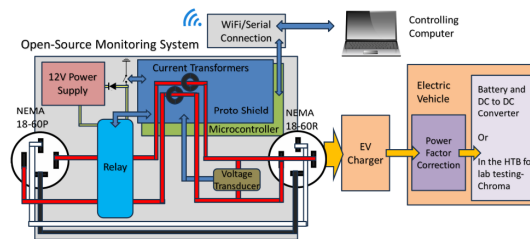


Figura 2.9: Esquema propuesto.

En conjunto, estos trabajos evidencian que las plataformas IoT y los sistemas de monitoreo constituyen un pilar fundamental para la operación moderna de la infraestructura de carga. La instrumentación mediante sensores, la conectividad y las interfaces de visualización permiten mejorar la observabilidad del sistema, optimizar el uso de las estaciones y facilitar la interacción con los usuarios. No obstante, persisten desafíos asociados a la escalabilidad, la interoperabilidad entre dispositivos heterogéneos y la robustez de las comunicaciones en entornos reales de despliegue masivo.

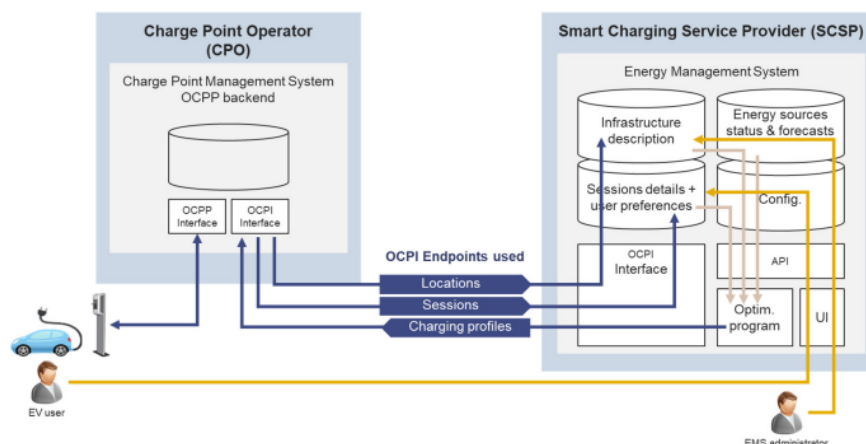


Figura 2.10: Conexión CPO-SCSP y arquitectura EMS.

2.4. Arquitecturas de carga inteligente e interoperabilidad

El despliegue masivo de infraestructura de carga para vehículos eléctricos no solo depende de la disponibilidad física de estaciones, sino también del diseño de arquitecturas inteligentes que permitan coordinar actores, sistemas y flujos de información de forma interoperable. En este contexto, la estandarización de protocolos, la definición de capas funcionales y la integración entre operadores, usuarios, vehículos y redes eléctricas se convierten en elementos clave para garantizar escalabilidad, eficiencia y adopción sostenible. El artículo [23] aborda la carga inteligente utilizando el protocolo Open Charge Point Interface (OCPI) como mecanismo de interoperabilidad entre operadores de movilidad eléctrica y proveedores de servicios. Su principal aporte consiste en demostrar cómo OCPI permite desacoplar la operación de las estaciones de carga de las plataformas de gestión, facilitando el roaming, la autenticación de usuarios y el intercambio de información entre distintos actores del ecosistema. Este enfoque favorece un modelo abierto donde múltiples proveedores pueden integrarse sin depender de soluciones propietarias. Como limitación, el trabajo se centra principalmente en la interoperabilidad a nivel de servicios, dejando menos desarrollados aspectos de optimización energética y control fino del proceso de carga. Desde una perspectiva ar-

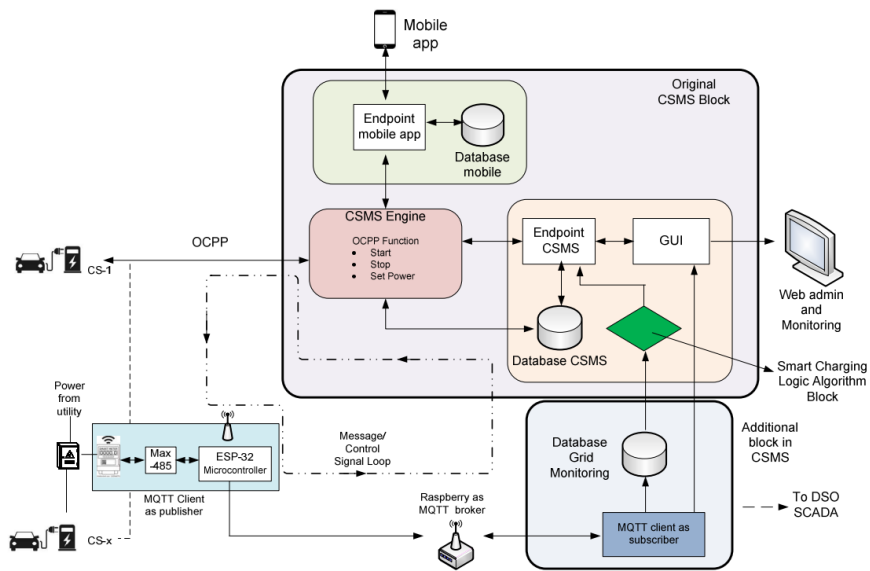


Figura 2.11: Arquitectura propuesta.

quitectónica más amplia, el artículo [24] propone un diseño de arquitectura inteligente para la carga de vehículos eléctricos a batería, estructurando el sistema en capas que incluyen dispositivos físicos, comunicación, gestión y aplicaciones. El aporte central es la formalización de un modelo lógico que permite integrar estaciones, vehículos, red eléctrica y plataformas de control dentro de una misma infraestructura coordinada. Esta aproximación facilita la incorporación de mecanismos de gestión de demanda, monitoreo y control distribuido. No obstante, el enfoque es principalmente conceptual, por lo que su validación práctica en entornos reales de gran escala queda como un desafío abierto. Complementariamente, el artículo [25] presenta una revisión integral sobre infraestructura de carga inteligente para electromovilidad. Su contribución al capítulo radica en ofrecer una visión global del estado del arte en arquitecturas, estándares, modelos de operación y tecnologías habilitantes. El trabajo identifica tendencias como la integración con redes inteligentes, la interoperabilidad entre plataformas, la gestión dinámica de carga y la coordinación con energías renovables. Además, expone brechas existentes en términos de estandarización y compatibilidad entre soluciones comerciales. Como limitación, al tratarse de un estudio de revisión, no propone una arquitectura concreta, sino que sintetiza enfoques existentes y desafíos futuros. En conjunto, estos

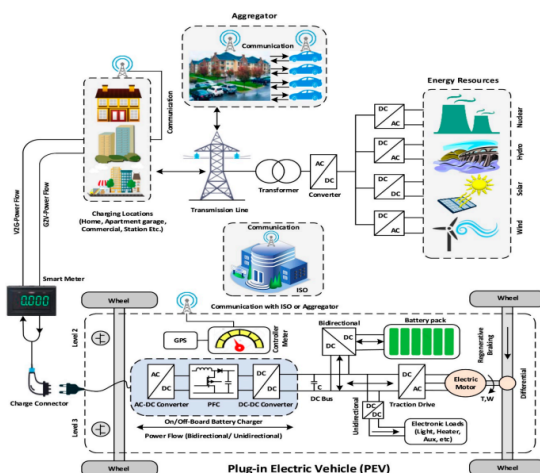


Figura 2.12: Un esquema de los componentes de un sistema V2G y las comunicaciones de la red de flujo de energía referenciado en.

trabajos muestran que la evolución de la infraestructura de carga requiere pasar de soluciones aisladas a arquitecturas inteligentes e interoperables. La adopción de protocolos abiertos como OCPI, junto con modelos arquitectónicos bien definidos, permite integrar múltiples actores y tecnologías dentro de un ecosistema común. Sin embargo, aún persisten retos relacionados con la validación práctica de estas arquitecturas, la armonización de estándares y la coordinación entre servicios, red eléctrica y usuarios finales en escenarios de despliegue a gran escala.

2.5. Gestión energética, optimización y energías renovables

El crecimiento de la electromovilidad impone nuevos desafíos en la gestión energética de la infraestructura de carga, donde no solo se busca abastecer vehículos eléctricos, sino también optimizar el uso de recursos, minimizar impactos sobre la red y aprovechar fuentes de energía renovable. En este contexto, la carga inteligente debe incorporar estrategias de control, predicción y coordinación que permitan balancear demanda, generación y capacidad del sistema.

El artículo [26] propone una estrategia de carga inalámbrica inteligente y sostenible que integra energías renovables con plataformas IoT para la gestión del proceso de carga. Esta nueva forma de cargar un vehículo es flexible, adaptable y de fácil uso. Propone todo un ecosistema alrededor de la carga inalámbrica, que permite monitorear tanto la estación de carga como la batería del EV (mediante la instalación de microcontroladores y sensores apropiados) con un aplicación en el teléfono. Lo anterior permite obtener las métricas de uso y variables de la carga. Además de esta ventaja, busca dejar de ocupar los cables de carga, que según el estudio, requieren una mantención o reemplazo constante, además de que así se aseguran una forma universal para cargar cada vehículo. Por otra parte, el estudio hizo un prototipo a escala para realizar las pruebas, por lo que falta la implementación en un escenario real. Aunque algo que lo diferencia de otras investigaciones, es que hace un estudio de costos de implementación y gasto energético para una más expedita evaluación en la integración de un sistema así.

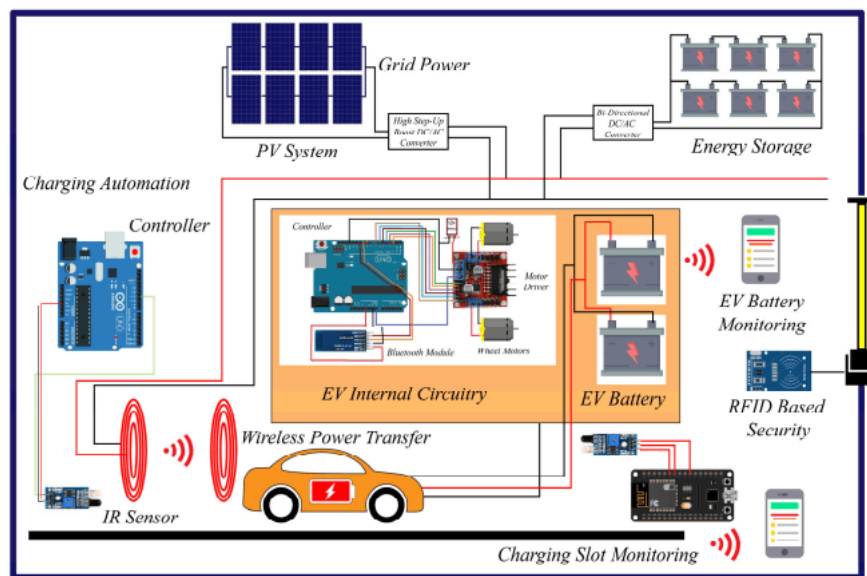


Figura 2.13: Integración de energía renovable e IoT para la transferencia inalámbrica avanzada de energía en EV.

Desde una perspectiva de operación de flotas, el artículo [27] presenta una plataforma IoT para la gestión de estacionamientos solares destinados a la carga inteligente

de flotas de vehículos eléctricos, utilizando datos en tiempo real y pronósticos de producción fotovoltaica. El aporte central consiste en coordinar la demanda de carga con la generación renovable disponible, permitiendo desplazar consumos hacia períodos de mayor producción solar. Esto lo hace con un software llamado SPEM, el cual cambia dinámicamente el flujo de las cargas en tiempo real. Este enfoque reduce la dependencia de la red y mejora la eficiencia energética global del sistema. SPEM se compara directamente con HOME@GRID, que también es un software de control de flujo, pero ocupa estadísticas y datos predictivos para funcionar, por lo que no es totalmente eficiente y queda corto en días borde como lo pueden ser las festividades. No obstante, el modelo depende fuertemente de la calidad de los pronósticos y de la infraestructura solar instalada, lo que limita su aplicabilidad en contextos con baja penetración de energías renovables.

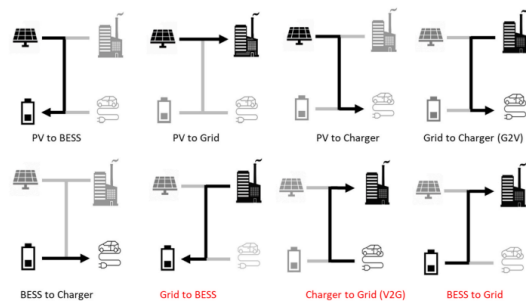


Figura 2.14: Ocho opciones de flujo energético entre las que puede elegir la gestión energética (SPEM no ocupa las rojas).

El artículo [28] aborda el problema de la carga eficiente de flotas de vehículos eléctricos mediante estrategias de optimización que consideran restricciones operacionales, capacidad de la red y requerimientos de los vehículos. Su contribución radica en la formulación de modelos que permiten planificar horarios y potencias de carga para minimizar costos y picos de demanda. Este tipo de gestión energética resulta clave para evitar sobrecargas en la infraestructura y mejorar el aprovechamiento de recursos disponibles. No reinventaron la rueda, sino que se enfocaron en mejorar la solución *open-source* "open e-Mobility", mediante pequeños añadidos que iban agregando a lo largo del tiempo. Logrando así crear su extensión llamado "Smart Charging System". Co-

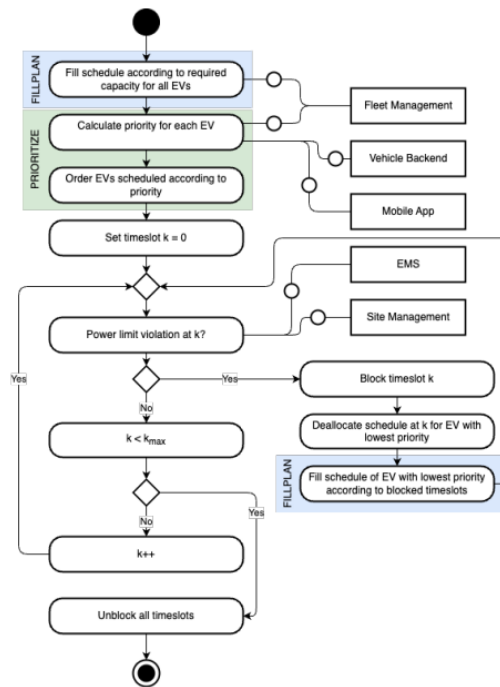


Figura 2.15: Diagrama de flujo.

mo limitación, el trabajo se enfoca principalmente en flotas organizadas, por lo que su adaptación a entornos abiertos con usuarios heterogéneos requiere extensiones adicionales. En conjunto, estos trabajos muestran que la gestión energética en infraestructura de carga debe ir más allá del simple suministro eléctrico. La incorporación de energías renovables, la predicción de generación y demanda, y la optimización de horarios y potencias permiten construir sistemas más eficientes y sostenibles. Aun así, persisten retos relacionados con la integración a gran escala, la incertidumbre en la producción renovable y la coordinación entre múltiples actores dentro de la red eléctrica.

2.6. Seguridad, autenticación y resiliencia de la infraestructura

La infraestructura de carga para vehículos eléctricos, al integrarse con plataformas IoT, servicios en la nube y mercados energéticos, se expone a múltiples riesgos de se-

guridad, privacidad y disponibilidad. La protección de los procesos de autenticación, facturación, intercambio de datos y control operativo se vuelve esencial para garantizar la confianza de los usuarios y la resiliencia del sistema frente a fallas o ataques. En este contexto, surgen enfoques que combinan tecnologías distribuidas, aprendizaje automático y mecanismos criptográficos para fortalecer la seguridad de la red de carga.

El artículo [29] propone un marco de gestión de carga soportado por blockchain que busca asegurar las transacciones y la coordinación entre estaciones, usuarios y operadores. Su aporte principal es el uso de un libro distribuido para registrar eventos de carga, autenticación y pagos de manera inmutable, reduciendo la dependencia de entidades centrales. Este enfoque incrementa la transparencia y la confianza en el sistema. Sin embargo, la incorporación de blockchain puede introducir sobrecarga computacional y latencias que afectan la operación en tiempo real cuando el número de nodos crece significativamente.

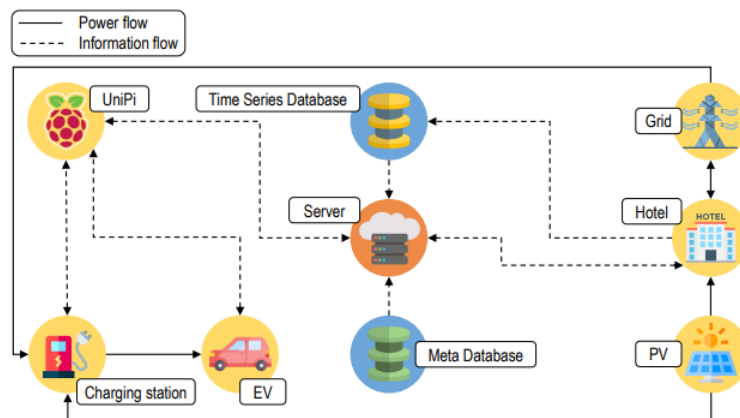


Figura 2.16: Arquitectura de sistema blockchain.

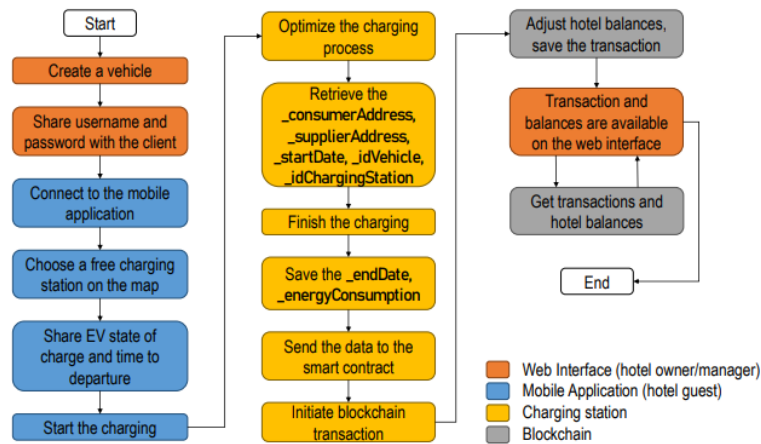


Figura 2.17: Flujo del proceso de carga.

Desde una perspectiva colaborativa, el artículo [17] presenta un esquema seguro y cooperativo para la compartición de puntos de carga privados mediante blockchain. El aporte central radica en permitir que propietarios de cargadores participen en la red de forma confiable, utilizando contratos inteligentes para gestionar acceso, autenticación y compensación. Implementan un sistema de evaluación para saber la confiabilidad de los entes pertenecientes a la red, así los participantes pueden reconocer más fácil la manipulación en alguna parte de la transacción (como lo puede ser el flujo de energía). Con un enfoque descentralizado se reduce la posibilidad de ataque desde el servidor central y se puede evidenciar quien está tratando de hacer cobros falsos, entre otras mejoras importantes de seguridad. Aunque aún falta la evaluación de los EVs, por lo que solo se puede hacer el seguimiento de las CS.

El artículo [18] introduce un método de autenticación soportado por blockchain en el borde (edge) junto con un sistema de facturación basado en criptomonedas para esquemas de carga P2P. Su contribución se centra en desplazar parte de la lógica de seguridad hacia nodos cercanos a las estaciones, reduciendo latencias y mejorando la escalabilidad del proceso de autenticación. Además, integra mecanismos de pago automatizados que refuerzan la confianza entre actores. Como limitación, el enfoque depende de capacidades computacionales adicionales en el borde, lo que puede incrementar los costos de despliegue.

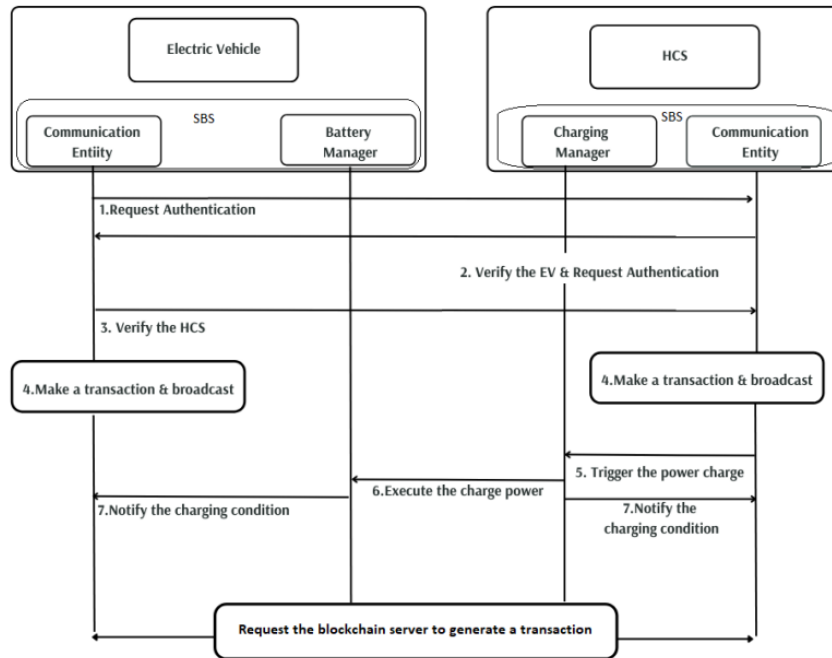


Figura 2.18: Flujo operacional del sistema de cobranza.

Complementariamente, el artículo [30] propone técnicas de aprendizaje automático en línea para la detección de intrusiones en sistemas distribuidos. En el contexto de estaciones de carga, su aporte consiste en analizar flujos de datos operacionales provenientes de la infraestructura para identificar comportamientos anómalos en tiempo real. El modelo tiene dos funcionamientos: uno de clasificación binaria para identificar tráfico malicioso, y uno multiclase que identifica que tipo de ataque es. Entrenaron y evaluaron el modelo con el dataset CICEVSE2024 entre otros, el cual tiene tráfico de tanto ISO15118 y OCPP, lo que es una buena aproximación a un escenario real. Estadísticamente tiene muy buenos resultados en ambos tipos de clasificación, pero utilizaron un ordenador de gama media-alta, por lo que falta evaluar en como incorporar el modelo dentro de las redes privadas.

Por último, el artículo [31] aborda la detección de ciberataques en sistemas de carga de vehículos eléctricos utilizando redes generativas adversarias basadas en Remaining Useful Life. Su principal aporte es modelar el comportamiento normal de la infraestructura para identificar desviaciones que indiquen fallas o ataques, incluso cuando existen

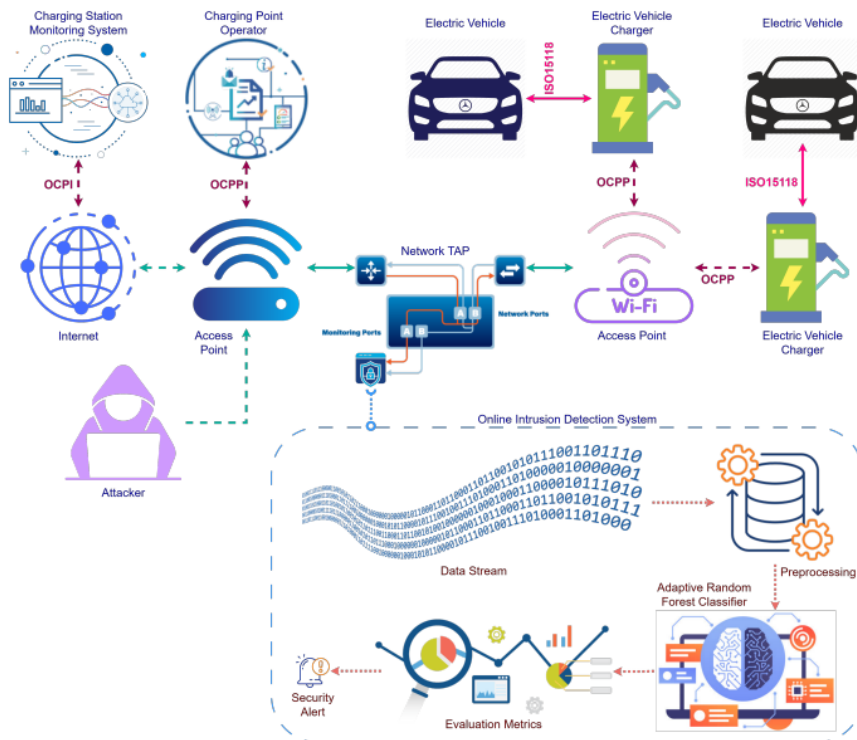


Figura 2.19: Arquitectura general del sistema online de detección de intrusiones propuesto.

pocos ejemplos etiquetados. Este enfoque resulta especialmente útil en redes de gran escala donde la supervisión manual es inviable. Como limitación, la efectividad del método depende fuertemente de la calidad y completitud de los datos recolectados por la plataforma.

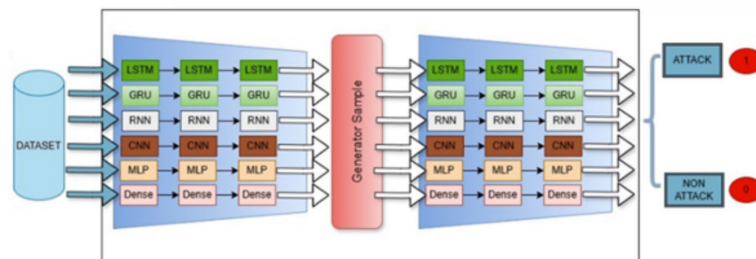


Figura 2.20: Estructura del modelo GAN-DL propuesto.

En conjunto, estos trabajos muestran que la seguridad de la infraestructura de carga debe abordarse desde múltiples capas: autenticación, transacciones, detección de intrusiones y resiliencia operacional. La combinación de blockchain, edge computing y aprendizaje automático permite construir sistemas más confiables y autónomos. No obstante, persisten desafíos relacionados con la escalabilidad, los costos de implementación y la gestión de datos heterogéneos en entornos reales de despliegue masivo.

Tabla 2.1: Tabla comparativa entre el estado del arte y solución propuesta.

Ref.	Capa de percepción	Capa de red	Capa de servicio (middleware)	Capa de aplicación	Capa de negocio
[16]	-	-	-	✓	✓
[17]	-	✓	✓	-	✓
[18]	-	✓	✓	-	✓
[19]	-	✓	-	✓	-
[20]	✓	✓	✓	✓	✓
[21]	✓	✓	✓	✓	✓
[22]	✓	✓	-	-	-
[23]	-	✓	✓	✓	✓
[24]	✓	✓	✓	-	✓
[25]	✓	✓	-	-	✓
[26]	✓	✓	✓	✓	✓
[27]	-	✓	✓	-	✓
[28]	✓	✓	✓	-	✓
[29]	✓	✓	✓	✓	✓
[30]	-	✓	✓	-	-
[31]	-	-	✓	-	✓
Nuestra propuesta	-	-	✓	✓	✓

Capítulo 3

Desarrollo de la plataforma

3.1. Diseño de Arquitectura del Sistema

3.1.1. Esquema General del Sistema

El sistema va a tener 6 entes clave: la estación de carga o vehículo eléctrico, un microcontrolador u objeto inteligente, el servidor/nube, y los tres tipos de usuario (usuario EV, administrador CS y administrador de la plataforma). Los cuales son representados en las interacciones mostradas en la figura 3.1. La estación de carga representa al objeto físico el cual se va a medir, estación de carga (CS) o vehículo eléctrico (EV), los datos medidos van a ser recopilados por un microcontrolador (MCU) u otro objeto inteligente y enviados a la nube para su procesamiento. Por parte externa del sistema, está el usuario EV que quiere buscar una estación de carga para cargar su vehículo, el administrador de la estación de carga, que presta la disponibilidad y la carga en sí, y el administrador de la plataforma que se encarga de monitorear y solucionar problemas con los usuarios y/o el sistema.

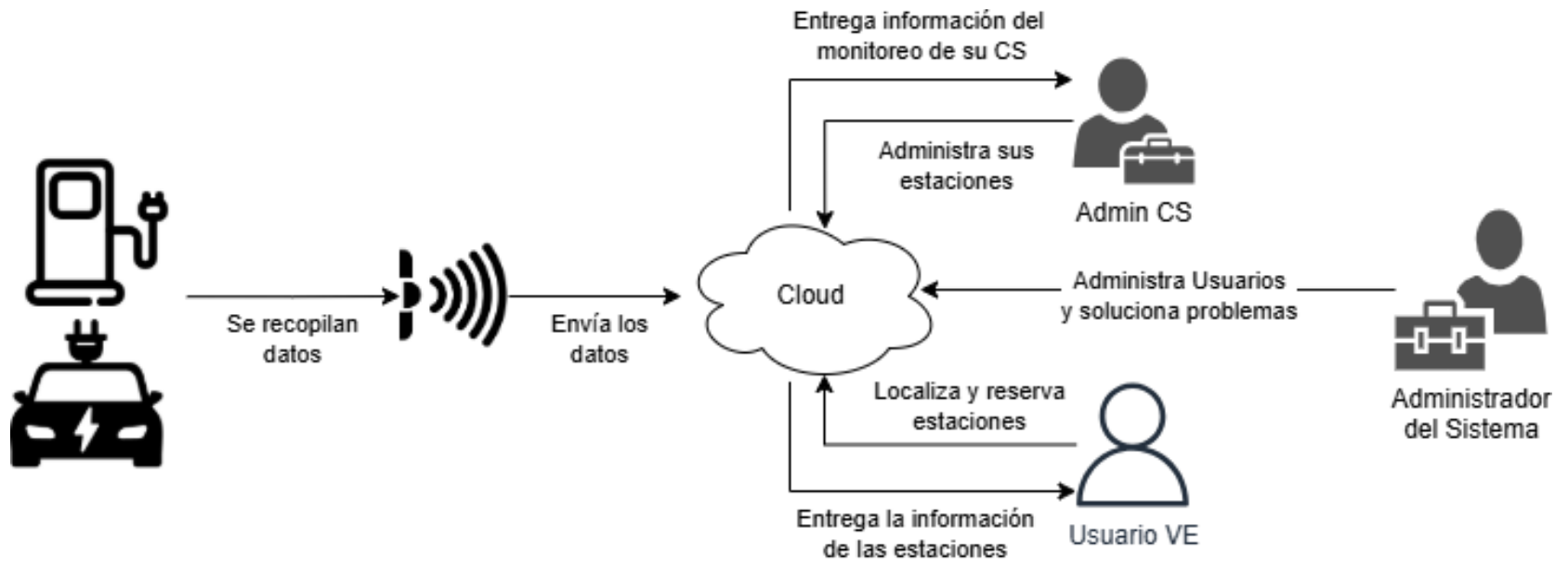


Figura 3.1: Esquema general del sistema

3.1.2. Arquitectura del Sistema

En la figura 3.2, se puede apreciar de forma más detallada la arquitectura del sistema, reconociendo a grandes rasgos los componentes que tendrá cada módulo y cómo interactúan entre sí.

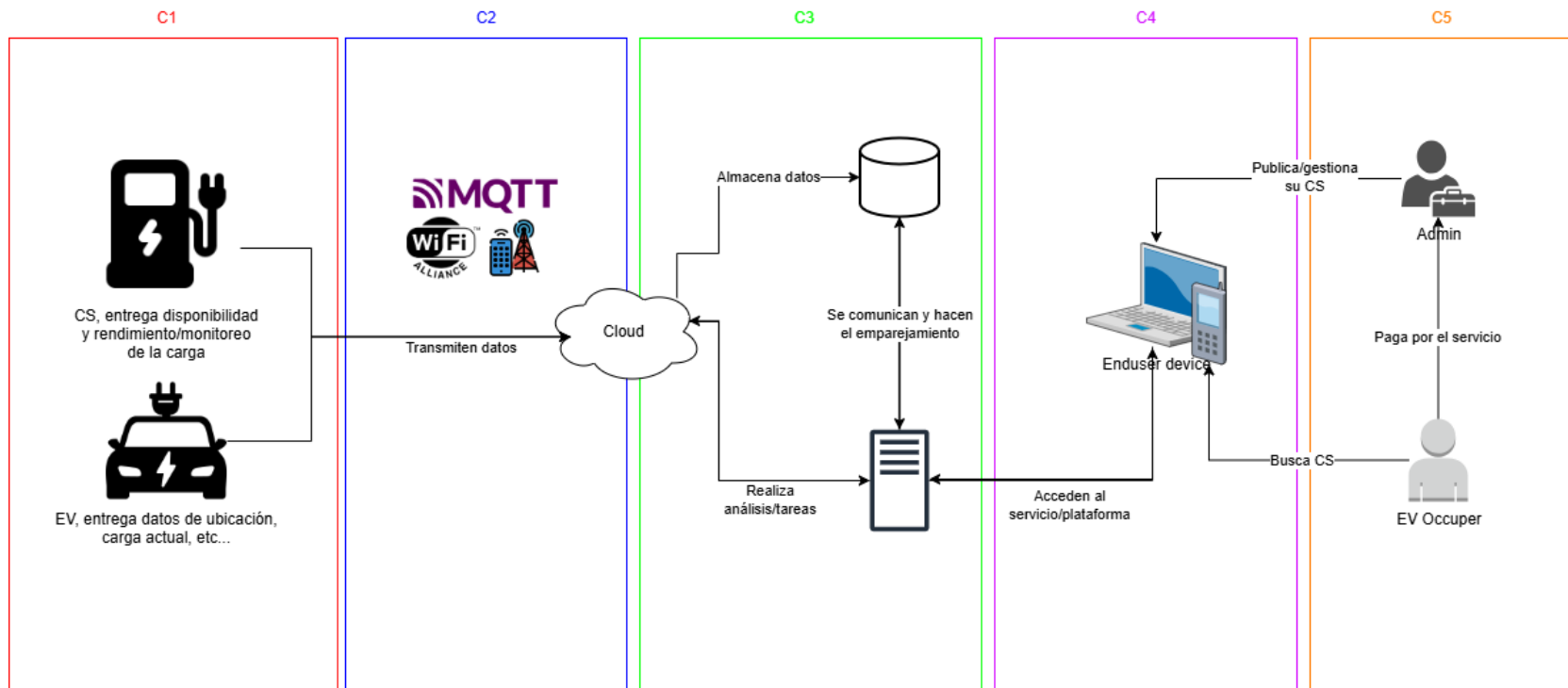


Figura 3.2: Arquitectura del sistema y relación entre módulos

3.1.3. Descripción de Módulos

A continuación, se pueden identificar los módulos que componen la arquitectura del sistema.

- **Capa de Percepción (M1):** Capa compuesta por los sensores y actuadores del sistema, la capa se encarga de recopilar información del mundo real necesaria para permitir a la plataforma adaptarse a las limitaciones existentes o plasmar la información recabada de forma útil para el usuario. Dentro de la capa de percepción se encuentran los vehículos eléctricos y las estaciones de carga.
- **Capa de Red (M2):** Capa compuesta por los medios de red y protocolos utilizados para permitir el funcionamiento del sistema en Internet, su propósito es permitir la comunicación entre todos los componentes del sistema. Los componentes físicos se comunican entre sí con protocolos propios, pero la conexión al servidor puede ser tanto por WiFi, con datos móviles o con otras tecnologías en el caso de las estaciones de carga. Ocupando protocolos tanto MQTT como HTTP/HTTPS.
- **Capa de Servicio (M3):** Capa compuesta por todos los elementos de backend que garantizan la funcionalidad del sistema, su propósito es el manejo y almacenamiento de datos ya anteriormente recopilados desde la capa de percepción o desde la capa de aplicación. Para el proyecto se propone el uso de brokers MQTT, API expressJS y una base de datos MongoDB.
- **Capa de Aplicación (M4):** Capa compuesta por elementos de visualización y presentación para la plataforma IoT, es decir es la capa de visualización gráfica y depende totalmente de los datos de la capa de servicio, su propósito es presentar grafica y intuitivamente al usuario los datos recopilados de una forma intuitiva y útil, además de manejar elementos propios del usuario, como características definidas por él (Por ejemplo, nombre de usuario, correo, tipo de cargador, etc.)

que son enviadas a la capa de servicio para su almacenamiento. Para el proyecto se propone el uso de React para el manejo de esta capa.

- **Capa de Negocio (M5):** Capa responsable de definir las políticas, reglas de operación, modelos comerciales y estrategias que rigen el funcionamiento y evolución de la plataforma IoT. Es la capa donde se definen los flujos de negocio, es decir, cómo los distintos actores del sistema (usuarios, propietarios de estaciones, administradores, etc.) interactúan con la solución, y qué beneficios o servicios se derivan de esas interacciones. Esto incluyen las reglas para el sistema de reserva, la monetización del servicio y el modelo de participación para los dueños de las estaciones privadas.

3.2. Módulos del Sistema

3.2.1. M1: Capa de Percepción

- **Propósito:** Este módulo es lo tangible de la solución, está encargado de recopilar la información correspondiente al cargador. Tanto su corriente, voltaje, potencia y disponibilidad del espacio.
- **Alcance:** Es la base para que se pueda compartir la estación de carga, realiza el monitoreo preciso del consumo energético.
- **Dependencias:** Se necesita la conectividad a internet presente en la capa de red para que se evidencie su funcionamiento.
- **Supuestos:** Se pueden obtener los datos desde las estaciones de carga y/o vehículos eléctricos.
- **Restricciones:** Este módulo necesita recursos que no se tienen disponibles para este proyecto.
- **Estructura General:** Los vehículos eléctricos y las estaciones de carga tienen sus propios métodos y protocolos para obtener su información respectiva. Esto puede ser la ubicación del vehículo, la energía transmitida entre los entes para una carga, la batería restante, etc...

3.2.2. M2: Capa de Red

- **Propósito:** Es el módulo en donde se transmiten los datos del sistema.
- **Alcance:** Sirve para transmitir los datos entre el microcontrolador y el servidor.
- **Dependencias:** Depende del módulo de percepción y el de servicio, ya que tiene que haber un acuerdo de que protocolo utilizar.

- **Supuestos:** Todas las estaciones de carga dentro del sistema, tienen conectividad a internet de alguna u otra forma.
- **Restricciones:** Es susceptible a cualquier tipo de cambio en el servidor, puede que se requiera hacer una actualización en cascada.
- **Estructura General:** Son los protocolos y tecnología que utilizan las estaciones de carga y los vehículos eléctricos para transmitir datos al servidor.

3.2.3. M3: Capa de Servicio

- **Propósito:** Procesar los datos entregados por el hardware, es el servicio en nube o servidor.
- **Alcance:** Es el intermediario entre los datos recopilados por el hardware y todo lo que ocurre con las interacciones del usuario.
- **Dependencias:** Depende del módulo de aplicación, ya que sino el usuario no podría interactuar con el sistema, y que el hardware esté funcionando correctamente. Además depende del módulo de negocio, ya que aquí se ve implementada la lógica que hará atractiva la propuesta.
- **Supuestos:** La capa de servicio siempre va a estar disponible para tomar los datos de cada estación de carga y resolver las solicitudes de los usuarios.
- **Restricciones:** Está sujeto a la cotización de un servidor en la nube o el levantamiento de un servidor propio.
- **Estructura General:** Esta capa está constituida por la nube/servidor para los procesos y además está conectada o contiene la base de datos. La base de datos tiene que almacenar los datos enviados del hardware, como la potencia, voltaje, corriente y ocupación de la estación de carga y datos duros, como lo son el costo de la utilización de cada estación y las métricas de cada instalación eléctrica (sin contar los datos de cada usuario). Además es la encargada de hacer el

emparejamiento de los usuarios con una estación de carga que cumpla con sus preferencias/necesidades.

Para el algoritmo de emparejamiento, vamos a ocupar el mostrado en el paper WEcharge, donde hace un emparejamiento implícito[32] que tiene en cuenta datos duros que son fijos (como el tipo de cargador o la potencia) y datos blandos como el tiempo de espera, el costo de uso, el tiempo de carga y la distancia entre la estación y el usuario.

Por último, debe confirmar la autorización de ocupar la estación de carga cuando se requiera.

3.2.4. M4: Capa de Aplicación

- **Propósito:** Gracias a este módulo, el usuario puede interactuar con el sistema.
- **Alcance:** El módulo se puede presentar como una aplicación móvil o web, en donde el usuario administrador de CS o ocupante de EV pueda interactuar con el sistema.
- **Dependencias:** Depende de la capa de servicio, sin servicio ni acceso a datos el usuario no puede interactuar con el sistema.
- **Supuestos:** El servidor está funcionando correctamente.
- **Restricciones:** No hay un diseñador UX en el equipo, por lo que la interfaz puede no ser lo más intuitiva de usar o amigable con los usuarios
- **Estructura General:** Está compuesta por una aplicación de React Vite montada en un servidor en la nube, la aplicación de React se comunica únicamente con la API que maneja los datos desde la base de datos. La aplicación también posee un capa de traducción para dispositivos móviles con “Capacitor”, la cual permite conectarse al servidor que contiene el Frontend de forma rápida y eficiente en dispositivos móviles.

3.2.5. M5: Capa de Negocio

- **Propósito:** Es el uso per se, lo que hace el usuario en la plataforma IoT.
- **Alcance:** Son las interacciones que puede tener el usuario con el sistema a través de la aplicación
- **Dependencias:** Depende de la capa de aplicación, ya que sin esta, el usuario no puede interactuar con el sistema.
- **Supuestos:** La red de usuarios es grande, para los dos usuarios que se definieron se necesita que su contraparte tenga actividad. No sirve de nada tener estaciones de carga disponibles si es que no hay nadie que las quiera ocupar, ni gente que busque estaciones de carga si es que no hay oferta (problema a resolver).
- **Restricciones:** Este módulo está limitado por el flujo de usuarios. Es importante que el servidor debe escalar en términos de procesamiento según la demanda.
- **Estructura General:** Es el modelo de negocio del sistema, dentro de este módulo se encuentra el cómo un administrador de estaciones de carga puede ver el funcionamiento actual y acumulado de sus estaciones, la reserva de las misma y que usuarios las han, están o irán a ocupar. Por parte de los ocupantes de EV, está en la búsqueda de la estación ad hoc, más allá del algoritmo que hay por detrás; el usuario tiene el deber de ingresar sus preferencias en la plataforma para que el servidor pueda ofrecerle una estación de parámetros similares a lo que se pide. Además está el incentivo para cada uno de ellos, el cobro por el servicio y el poder cargar su vehículo respectivamente. Es importante mencionar que al ser estaciones de carga privadas, el administrador de la estación es el responsable de que el usuario general tenga acceso a la estación de carga y pueda ocupar el servicio.

3.3. Diseño de Interfaces

3.3.1. Modelo de Navegación

Para definir un modelo de navegación hay que tener en cuenta los dos usuarios principales y el administrador de la plataforma y las necesidades de cada uno, por ello se realiza el siguiente esquema que representa las necesidades del usuario administrador de estaciones de carga, el usuario general que busca disponibilidad de estaciones de carga y del usuario administrador general de la plataforma. El usuario administrador de CS necesita saber el consumo en sus estaciones de carga, su ocupación y un informe de la utilización y reservas de estas, el usuario general necesita encontrar estaciones de carga cerca suyo y reservarlas o ver su disponibilidad y el usuario administrador de la plataforma necesita poder gestionar a todos los usuarios de la plataforma y sus características.

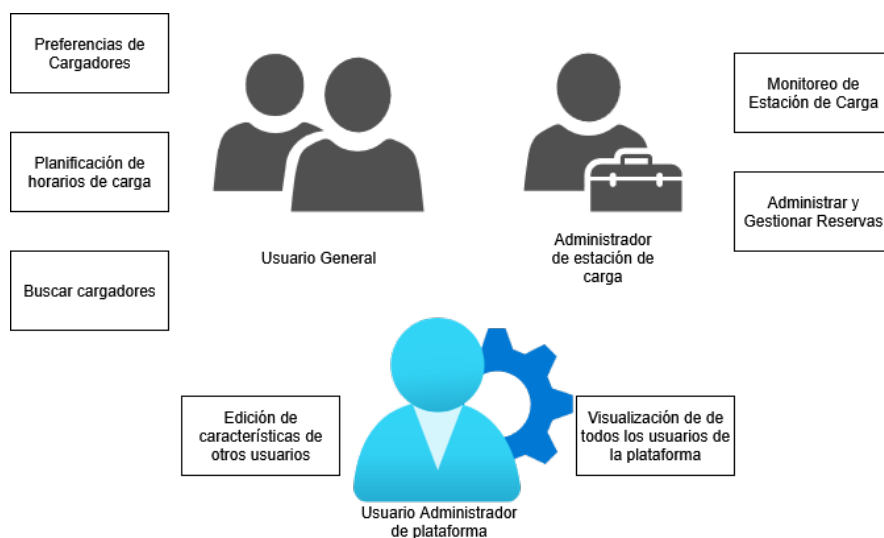


Figura 3.3: Vista sencilla de usuarios y sus necesidades.

Teniendo en cuenta esto se diseña el modelo de navegación de la sección web de la plataforma IoT que presenta distintas vistas para un usuario administrador de CS y un usuario general.

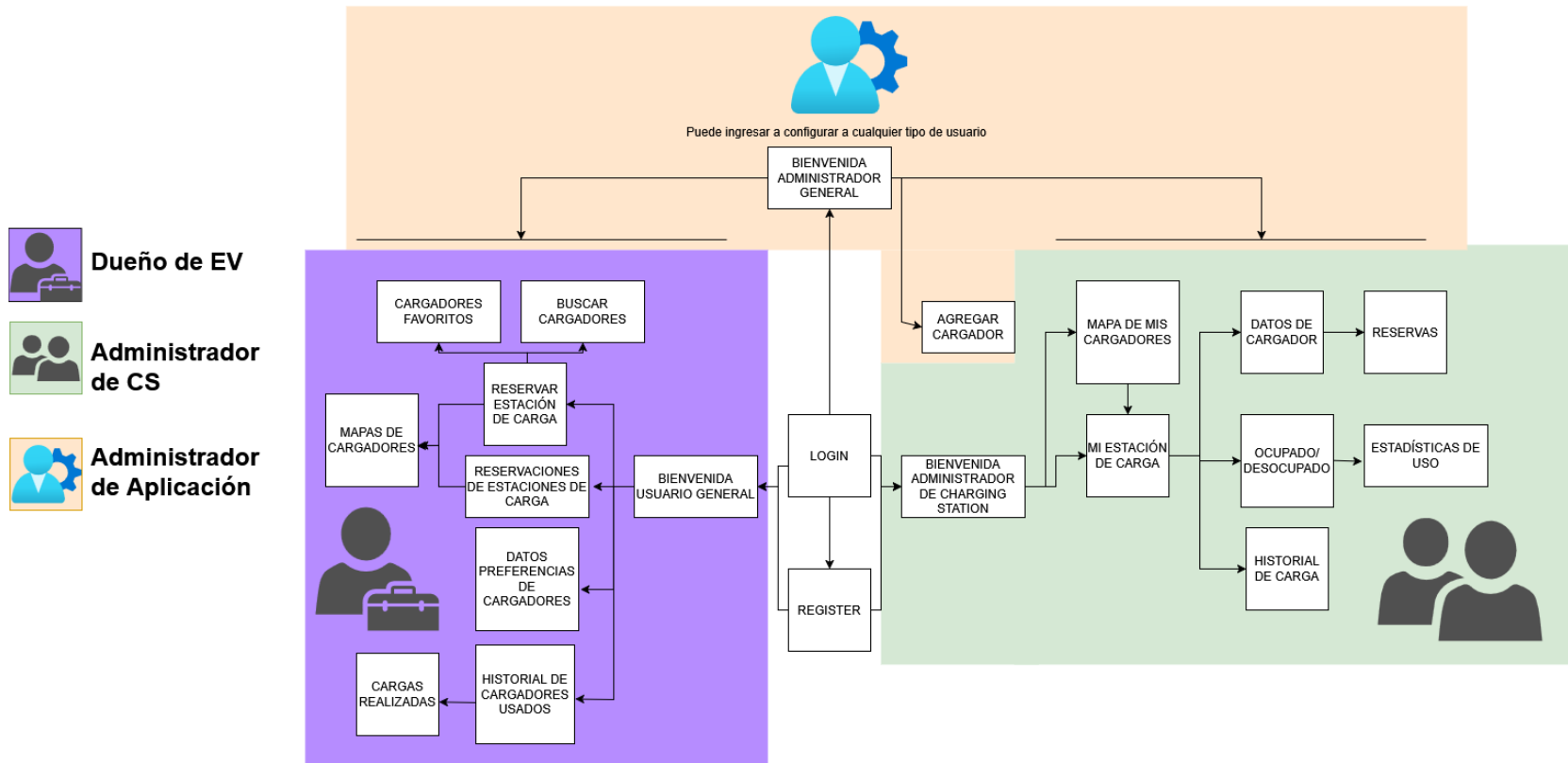


Figura 3.4: Modelo de navegación del FrontEnd de la plataforma IoT dividido en secciones, la sección violeta representa la interfaz disponible para los usuarios dueños de vehículos eléctricos (EV), la sección verde la interfaz disponible para el administrador de estaciones de carga (CS) y por último la sección naranja que representa lo posible para el usuario administrador de la aplicación.

En el modelo se describe el funcionamiento general y comportamiento esperado de la página web/móvil, partiendo desde el centro con un Login o Registro si es que el usuario no se encuentra registrado con anterioridad que culmina en dos bifurcaciones, una para el usuario general y otra para el usuario administrador de estaciones de carga. Si es que el usuario que ingresa es un usuario general, entonces verá las estaciones de carga cercanas a él, podrá notar sus disponibilidades, precios o distancias y agendar un horario. Si el usuario es un administrador de estaciones de carga, entonces tendrá opciones de monitoreo de su estación como potencia cargada, ocupación y ubicación de cada estación de carga en un mapa interactivo. Por último el Administrador de la plataforma tiene los privilegios para administrar las propiedades de todos los usuarios de la plataforma y agregar o eliminarles características como el por ejemplo, añadir una estación de carga al Administrador de estaciones de carga.

Capítulo 4

Implementación

Para la implementación del proyecto se hace enfoque en la sección de Software de la aplicación IoT, es decir, en la intercomunicación entre una base de datos, una API intermediaria y una interfaz gráfica, para aquello se realizan las siguientes tareas.

4.1. Implementación de Base de datos

Para la implementación de la base de de datos se decide ocupar MongoDB dada su flexibilidad y fácil modificación de estructura al implementar nuevos cambios. La estructura planteada para la base de datos es la siguiente:

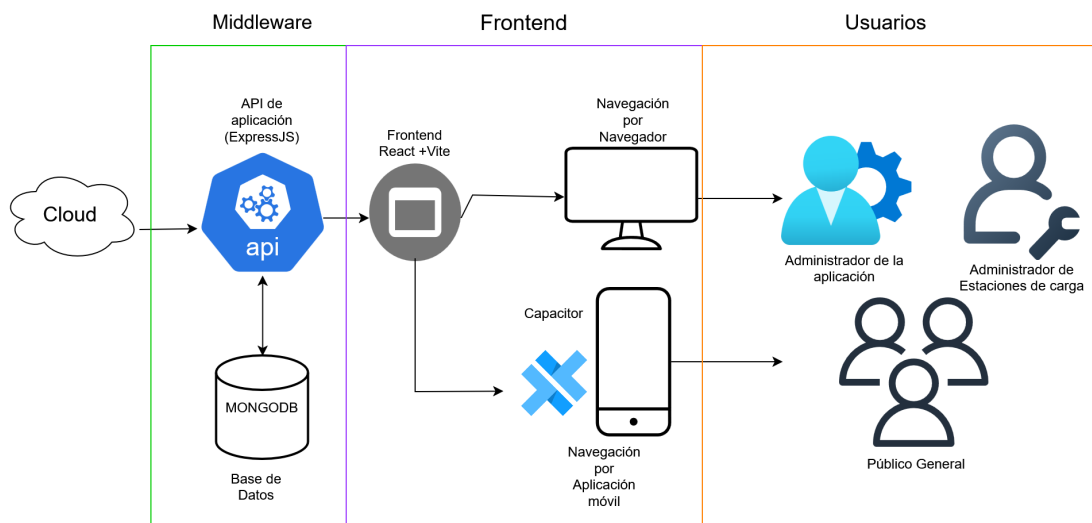


Figura 4.1: Diagrama explicativo del funcionamiento general de las tecnologías utilizadas en el proyecto con sus respectivas capas asociadas.

- **Cargadores:** Colección de todos los cargadores de vehículos eléctricos registrados en la aplicación.
 - **ID:** Identificación asociada a cargador.
 - **Nombre:** Nombre asignado al cargador.
 - **Ubicación:** Ubicación geográfica del cargador.
 - **Tipo de Cargador:** Especifica modelo del cargador.
 - **Potencia:** Potencia de carga del cargador.
 - **Costo de energía:** Cuanto cuesta cada kWh.
 - **Costo de estacionamiento:** Cuanto cuesta por minuto, el ocupar la estación de carga sin estar cargando el vehículo
 - **Estado:** Refleja si el cargador está disponible, ocupado o en mantenimiento.
 - **Reservas:** Lista de Reservas asociadas al cargador.
 - **Historial de Ocupación:** Lista de momentos en las que el cargador cambió de ocupado a disponible y viceversa.

- **Historial de Carga:** Lista de cargas registradas con fecha de realización.
- **Sesiones de Carga:** Colección de cargas planificadas o en realización.
 - **ID de Cargador:** Identificación de cargador asociada a la sesión.
 - **ID de Vehículo:** Identificación de vehículo asociada a la sesión.
 - **Tiempo de Inicio:** Fecha de inicialización de la carga.
 - **Tiempo de finalización:** Fecha de finalización de la carga.
 - **Energía entregada:** Carga realizada hasta el momento.
 - **Duración:** Minutos de carga registrados.
 - **Costo:** Costo calculado de carga.
 - **Activo:** Indica si la carga está sucediendo o si está planificada.
- **Notificaciones:** Colección de notificaciones generadas por el sistema para los usuarios.
 - **Usuario:** Identificación del usuario asociado a la notificación.
 - **Título:** Título descriptivo de la notificación.
 - **Mensaje:** Contenido detallado de la notificación.
 - **Tipo:** Clasificación de la notificación (informativa, alerta, recordatorio, etc.).
 - **¿Fue leído?:** Indica si la notificación fue visualizada por el usuario.
 - **Fecha de Creación:** Fecha en la que la notificación fue generada.
 - **Fecha de Modificación:** Fecha de la última actualización de la notificación.
- **Reservaciones:** Colección de reservas de cargadores realizadas por los usuarios.
 - **ID de Cargador:** Identificación del cargador reservado.
 - **ID de Vehículo:** Identificación del vehículo asociado a la reserva.
 - **ID de Usuario:** Identificación del usuario que realizó la reserva.

- **Tiempo de Inicio:** Fecha y hora de inicio de la reserva.
 - **Tiempo de finalización:** Fecha y hora de finalización de la reserva.
 - **Finalización Estimada:** Fecha y hora estimada de finalización de la carga.
 - **Estado:** Estado actual de la reserva (activa, cancelada, completada).
 - **Tiempo de Carga estimado:** Tiempo estimado necesario para completar la carga.
- **Usuarios:** Colección de usuarios registrados en la aplicación.
 - **ID:** Identificación única del usuario.
 - **Nombre:** Nombre completo del usuario.
 - **Email:** Dirección de correo electrónico del usuario.
 - **Contraseña:** Credenciales de acceso del usuario.
 - **Rol:** Indica si el usuario es administrador de la aplicación, administrador de estaciones de carga o dueño de vehículos eléctricos.
 - **Cargadores:** Cargadores pertenecientes o administrados por el usuario.
 - **Vehículos:** Lista de vehículos eléctricos asociados al usuario.
 - **Estaciones favoritas:** Lista de estaciones favoritas del usuario.
 - **Fecha de creación:** Fecha de registro del usuario en la aplicación.
 - **Vehículos:** Colección de vehículos eléctricos registrados en la aplicación.
 - **ID:** Identificación única del vehículo.
 - **Modelo:** Indica el tipo o modelo del vehículo eléctrico.
 - **Tipo de Cargador:** Indica el tipo de cargador compatible con el vehículo.
 - **Capacidad de batería:** Capacidad total de la batería del vehículo.
 - **Carga Actual:** Nivel de carga actual de la batería (SOC).

- **Historial de Carga:** Lista de cargas realizadas por el vehículo con sus respectivas fechas.

En la figura 4.2 se explica en un esquema relacional el funcionamiento interconectado de la base de datos.

Para poblar la base de datos se utiliza el dataset de estaciones de carga de la universidad de Georgia [33] como base y se reestructura para crear nuevos usuarios asociados a vehiculos eléctricos presentes en el dataset, nuevos cargadores según los datos recopilados del dataset y un historial de cargas según las cargas registradas en el dataset.

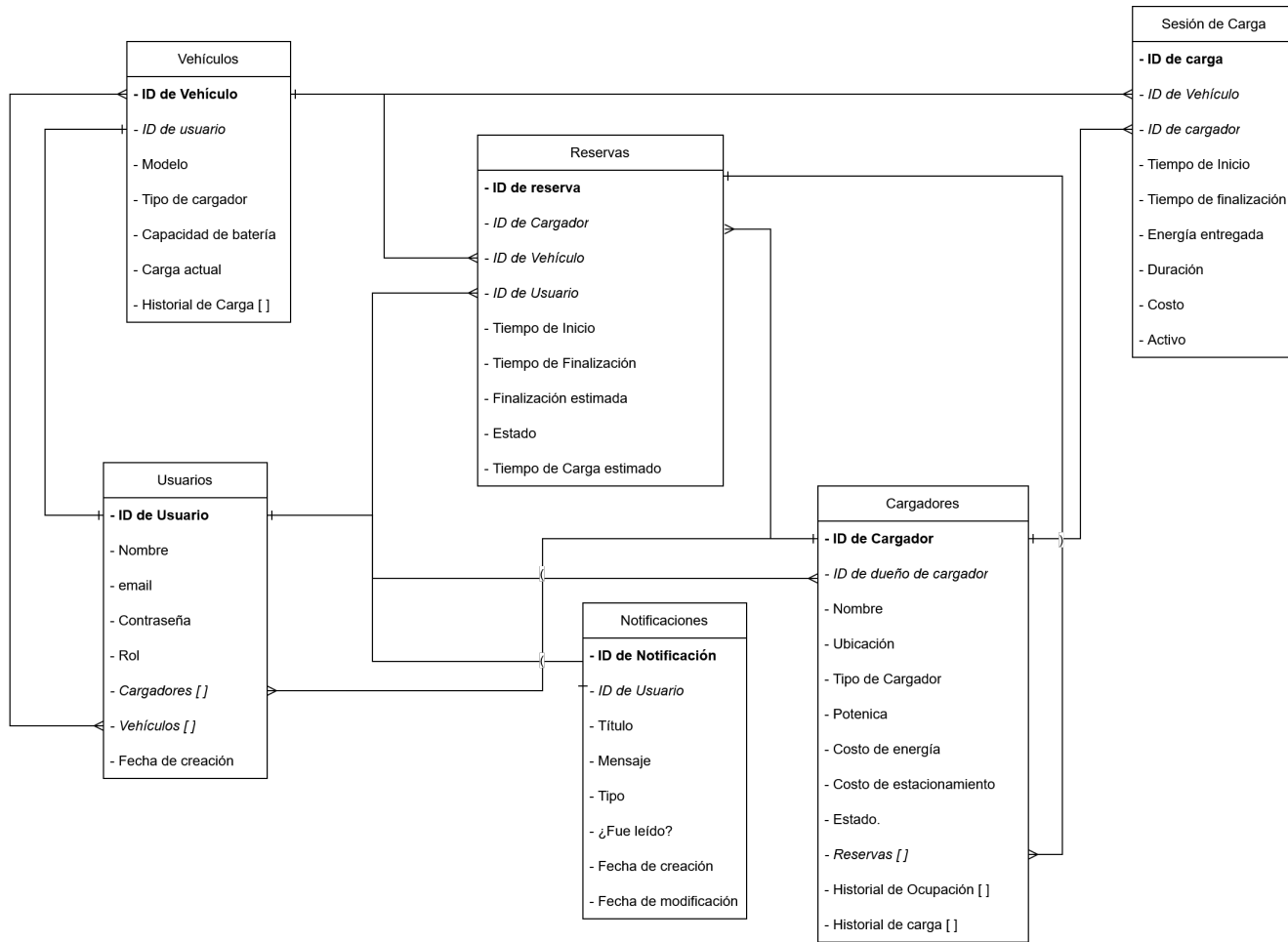


Figura 4.2: Esquema de la Base de datos con las colecciones explicadas con anterioridad, se presentan sus relaciones.

4.2. Implementación de API

Se busca utilizar la API como Middleware entre la base de datos y la interfaz gráfica de la aplicación, por lo que ésta se encarga de mostrar a la interfaz web información relevante de la base de datos mientras permite la modificación de ciertas configuraciones específicas.

La API se compone de las siguientes tecnologías

- **ExpressJS:** Framework para Node.js que permite la creación de APIs REST de manera sencilla y estructurada, facilitando la gestión de rutas, solicitudes HTTP y middleware.
- **Mongoose:** Biblioteca de modelado de objetos (ODM) para MongoDB que permite definir esquemas, validar datos y gestionar la comunicación entre la base de datos y la aplicación de forma eficiente.
- **Socket.IO:** Biblioteca que permite la comunicación bidireccional en tiempo real entre el servidor y los clientes, utilizada para la actualización instantánea de estados como disponibilidad de cargadores o notificaciones.
- **JWT:** Mecanismo de autenticación basado en tokens (JSON Web Tokens) que permite verificar la identidad de los usuarios de forma segura sin necesidad de mantener sesiones en el servidor.
- **bcryptjs:** Biblioteca utilizada para el cifrado de contraseñas, asegurando que las credenciales de los usuarios se almacenen de forma segura mediante funciones hash irreversibles.

Para la implementación de la API se diseñan diferentes tipos de Endpoints con funciones de autenticación, autorización, consulta de datos, modificación de datos, simulación de procesos, calendarización de eventos, cálculo y entrega de estadísticas y manejo de eventos en tiempo real. A continuación se enumeran los Endpoints diseñados por el estudiante junto con una breve descripción sobre su función.

Autenticación & Autorización

- **POST /api/auth/login:** Inicia sesión con email y contraseña; devuelve token JWT y datos del usuario.
- **GET /api/auth/profile:** Recupera el perfil del usuario autenticado (Permite continuar conectado con la cuenta en la aplicación hasta que se detecte inactividad).
- **PUT /api/auth/profile:** Actualiza nombre y/o contraseña del usuario autenticado.
- **middleware authenticateToken:** Verifica y decodifica JWT (Clave que válida una sesión activa del usuario).

Entrega de datos (consultas / GET)

- **GET /apichargers:** Lista de todos los cargadores con sus especificaciones (permite filtros por estado y/o tipo).
- **GET /apichargers/:id:** Detalles de un cargador específico.
- **GET /apichargers/:id/usage-history:** Historial de uso de un cargador (rango de fechas).
- **GET /apichargers/:id/usage-stats:** Estadísticas de uso por periodo (día/semana/mes/año).
- **GET /apichargers/nearby:** Obtiene las estaciones cercanas a una ubicación.
- **GET /apicharging-sessions/active/by-reservation/:reservationId:** Retorna la sesión de carga con la reserva.
- **GET /apifavourites/:userId:** Obtiene las estaciones favoritas del usuario.
- **GET /apireservations:** Obtiene reservas del usuario autenticado.

- **GET /api/notifications:** Obtiene todas las notificaciones del usuario (permite filtrar por solo no leídas).
- **GET /api/vehicles:** Lista de vehículos (opcionalmente filtrable por ID de usuario).
- **GET /api/vehicles/:id/actual:** Retorna las reservas actuales y futuras de un vehículo, no entrega las que ya terminaron.
- **GET /api/vehicles/:id/charging-history:** Obtiene el historial de cargas de un vehículo.
- **GET /api/calendar/:chargerId:** Disponibilidad / calendario de reservas de un cargador en un rango de fechas.
- **GET /api/simulator/status:** Estado de carga de vehículos simulados activos.
- **GET /api/realtime/chargers:** Estado en tiempo real de todos los cargadores (incluye sesión activa).
- **GET /api/users:** (Función para administrador de la plataforma) Listado de todos los usuarios y sus detalles.

Modifican datos (crear / actualizar / eliminar)

- **POST /api/auth/registerEvUser:** Crea un nuevo usuario de tipo ocupante de EV.
- **POST /api/chargers:** Crear un nuevo cargador con detalles especificables (requiere auth y permisos).
- **PATCH /api/chargers/:id/name:** Actualizar nombre de un cargador según su ID.
- **DELETE /api/chargers/:id:** Eliminar un cargador según su ID (y limpiar referencias en otros objetos).

- **POST /api/favourites/:userId/:stationId:** Agrega una estación como favorita para un usuario.
- **DELETE /api/favourites/:userId/:stationId:** Quita una estación de favoritos para un usuario.
- **POST /api/reservations:** Crear una nueva reserva (valida conflictos entre cargador/vehículo).
- **DELETE /api/reservations/:id:** Cancelar una reserva según su ID.
- **POST /api/reservations/:id/accept:** Aceptar una reserva (usuario, propietario o admin).
- **POST /api/reservations/:id/cancel:** Cancelar una reserva con motivo (registra motivo y quién canceló).
- **POST /api/reservations/update-statuses:** Actualiza estados de reservas según hora actual (accionado por scheduler).
- **POST /api/notifications:** Crear notificación y emitir en tiempo real.
- **POST /api/notifications/:id/read:** Marcar una notificación como leída.
- **POST /api/notifications/read-all:** Marcar todas las notificaciones como leídas.
- **DELETE /api/notifications/:id:** Eliminar notificación segun su ID.
- **POST /api/vehicles:** Crear vehículo nuevo según parámetros.
- **PATCH /api/vehicles/:id:** Actualizar detalles de vehículo (campos: model, chargerType, batteryCapacity, currentChargeLevel).
- **DELETE /api/vehicles/:id:** Eliminar un vehículo según su ID.
- **POST /api/simulator/create:** Crear un nuevo vehículo simulado para empezar proceso de simulación de carga.

- **POST /api/simulator/start**: Iniciar sesión de carga simulada (modifica estado del simulador y del vehículo simulado).
- **PATCH /api/users/:id/role**: (Función para administrador de la plataforma) Cambiar rol de un usuario.

Tiempo real / WebSockets

- **utils/socket.init(server, origin)**: Inicializa websocket en el servidor; registra eventos y origen CORS para notificaciones en tiempo real.
- **utils/socket.emitToUser(userId, event, data)**: Emite un evento a un usuario específico (notificaciones, updates).

Reservas

- **reservationScheduler.startReservationScheduler()**: Programador que se ejecuta cada 5 minutos, busca reservas que inician en los próximos 15 minutos y envía recordatorios.

Simulador de vehículos (lógica y controlador)

- **Class EVSimulator / ev-simulator.js**
 - **startCharging(vehicleId, chargerId, targetCharge)**: Inicia sesión de carga simulada; actualiza nivel de carga periódicamente (cada 5 segundos), calcula incremento según potencia y capacidad; crea sesión final al terminar; emite eventos de progreso.
 - **stopCharging(vehicleId)**: Detiene la carga simulada para un vehículo.
 - **getStatus()**: Retorna estado de vehículos que están cargando (nivel actual, objetivo, etc.).

- **simulator-controller.js (gestión de instancias):**
 - **createSimulatedVehicle(vehicleData):** Crea vehículo para simulación.
 - **startSession(vehicleId, chargerId):** Inicia sesión usando el controlador (coordina instancias).
 - **stopSession(vehicleId):** Detiene sesión simulada.
 - **getActiveSimulations():** Lista simulaciones activas.

Estadísticas & Recomendaciones

- **GET /api/stats/overview:** Conjunto de métricas globales (vehículos, cargadores, sesiones, energía, reservas).
- **GET /api/stats/charger-types:** Estadísticas por tipo de cargador.
- **GET /api/stats/energy-monthly:** Consumo total de energía por mes para el administrador de la aplicación.
- **GET /api/stats/top-energy-vehicles:** Top de vehículos por consumo de energía para el administrador de la aplicación.
- **GET /api/stats/usage-by-hour:** Uso por hora del día de todos los cargadores en general para el administrador de la aplicación.
- **POST /api/recommendations:** Calcula y devuelve cargadores recomendados según ubicación y vehículo (score, distancia, tiempo estimado).

Dadas estas tecnologías y Endpoints diseñados se puede realizar una comunicación con la base de datos y obtener características almacenadas en esta o modificarlas, lo anterior siendo accesible para los usuarios de la plataforma posteriormente mediante el uso de una interfaz gráfica.

4.3. Implementación de Interfaz gráfica

Para el desarrollo de la interfaz gráfica de la aplicación se opta por utilizar el Framework de React-Vite para el fácil despliegue de la plataforma, dada su versatilidad y popularidad es una de las opciones más sencillas de desarrollar.

En específico el frontend de la aplicación utiliza las siguientes tecnologías:

- **React 19.1.0:** Framework de JavaScript para construir interfaces de usuario basadas en componentes. Se utiliza en toda la aplicación para crear componentes reutilizables.
- **TypeScript 5.8.3:** Superset de JavaScript que añade tipado estático. Utilizado en todos los componentes y contextos de la aplicación para mejorar la mantenibilidad, prevenir errores en tiempo de compilación y proporcionar autocompletado en el IDE.
- **Vite 6.3.5:** Herramienta de compilación y servidor de desarrollo de nueva generación. Permite compilación individual de módulos durante el desarrollo (evita recompilación completa del proyecto) y optimización del proyecto para producto final.
- **Tailwind CSS 4.1.11:** Framework CSS utility-first para diseño intuitivo y amigable de interfaces. Utilizado extensivamente en todos los componentes mediante clases semánticas personalizadas (cards, botones, inputs, badges, modales, layouts responsivos).
- **Axios 1.11.0:** Cliente HTTP basado en promesas para realizar peticiones a la API REST del backend. Utilizado en componentes como `AdminDashboard` para operaciones CRUD de usuarios, vehículos y estaciones.
- **Leaflet 1.9.4 y React Leaflet 5.0.0:** Biblioteca de mapas interactivos open-source en conjunto con su wrapper para React. Utilizada para visualizar ubicaciones de cargadores, seleccionar coordenadas y mostrar marcadores interactivos.

- **Chart.js 4.5.0** y **React ChartJS 2 5.3.0**: Biblioteca de gráficos en conjunto con su wrapper para React. Utilizada para visualizar estadísticas de uso, ocupación de cargadores y métricas de sesiones de carga.
- **date-fns 4.1.0**: Librería moderna para manipulación de fechas. Utilizada en componentes de calendario para formateo de fechas, cálculo de rangos y validaciones temporales.
- **Fuse.js 7.1.0**: Librería de búsqueda difusa (fuzzy search). Implementada en buscadores dentro de la aplicación para búsqueda tolerante a errores por nombre y ubicación de cargadores.
- **Socket.IO Client 4.8.1**: Cliente WebSocket para comunicación en tiempo real. Utilizado para recibir actualizaciones en vivo de estado de cargadores, notificaciones push y sincronización de sesiones de carga activas.
- **Capacitor 7.4.0**: Framework para construir aplicaciones nativas multiplataforma desde código web. Proporciona la infraestructura para compilar la aplicación React como APK de Android, puede trabajar en conjunto con plugins de geolocalización y de notificaciones locales en dispositivos móviles para mayor integración en el entorno móvil. Es utilizada para compilar una versión compatible con Android de la aplicación.
- **@mantine/notifications 8.3.5**: Sistema de notificaciones. Utilizado en componentes para mostrar mensajes temporales de éxito, error o información al usuario.

Para el desarrollo de las vistas de la aplicación se diseñan las siguientes páginas de uso general, uso del administrador de la aplicación y uso del administrador de estaciones de carga con sus mencionadas funcionalidades:

Uso general

- **Inicio de sesión y acceso:** Pantalla de entrada con autenticación por correo y contraseña. Su diseño en escritorio se presenta como un formulario centrado con panel lateral de marca, en cambio en Android usa un flujo de pantalla completa con botones grandes y teclado optimizado para facilitar la interacción táctil.

Lista de Endpoints utilizados:

- POST /api/auth/login (ver sección 4.2)



Figura 4.3: Inicio de sesión en la plataforma.

- **Registro:** En esta vista el usuario EV puede registrarse junto con su vehículo. Presenta un formulario “estándar”, donde el usuario tiene que ingresar su nombre completo, correo, confirmación de correo, contraseña y datos del vehículo. Formalmente, no se incluyen los datos del vehículo en tiempo real ni se ha implementado un sistema para hacerlo. En los datos del vehículo aparece el nombre del modelo, la capacidad de la batería y el tipo de cargador (que se puede utilizar para filtrar las estaciones). *Lista de Endpoints utilizados:*

- GET /api/vehicles (ver sección 4.2)

- POST /api/auth/registerEvUser (ver sección 4.2)
- POST /api/auth/login (ver sección 4.2)

El formulario de registro, titulado "Crear cuenta", está dividido en dos secciones principales. La primera sección contiene campos para "Nombre completo", "Correo", "Confirmar correo", "Contraseña" (con un botón "Mostrar" a su derecha) y "Confirmar contraseña" (con un botón "Mostrar" a su derecha). La segunda sección, titulada "Vehículo", incluye un menú desplegable con la opción "Otro (ingresar manualmente)", un campo "Vehículo", un campo "Capacidad (kWh)" con el valor "0", y otro menú desplegable "Tipo cargador" con la opción "Selecciona tipo...". Al final del formulario hay dos botones: "Crear cuenta" (de color azul) y "Volver" (de color gris).

Figura 4.4: Formulario de registro.

- **Perfil del usuario:** Tiene como funciones permitir visualizar y actualizar datos personales (nombre, correo, foto), así como configurar preferencias de notificaciones y modo oscuro. Su diseño en escritorio se organiza en dos columnas, en cambio en Android se apilan secciones con botones táctiles amplios para mejor accesibilidad.

Lista de Endpoints utilizados:

- GET /api/auth/profile (ver sección 4.2)

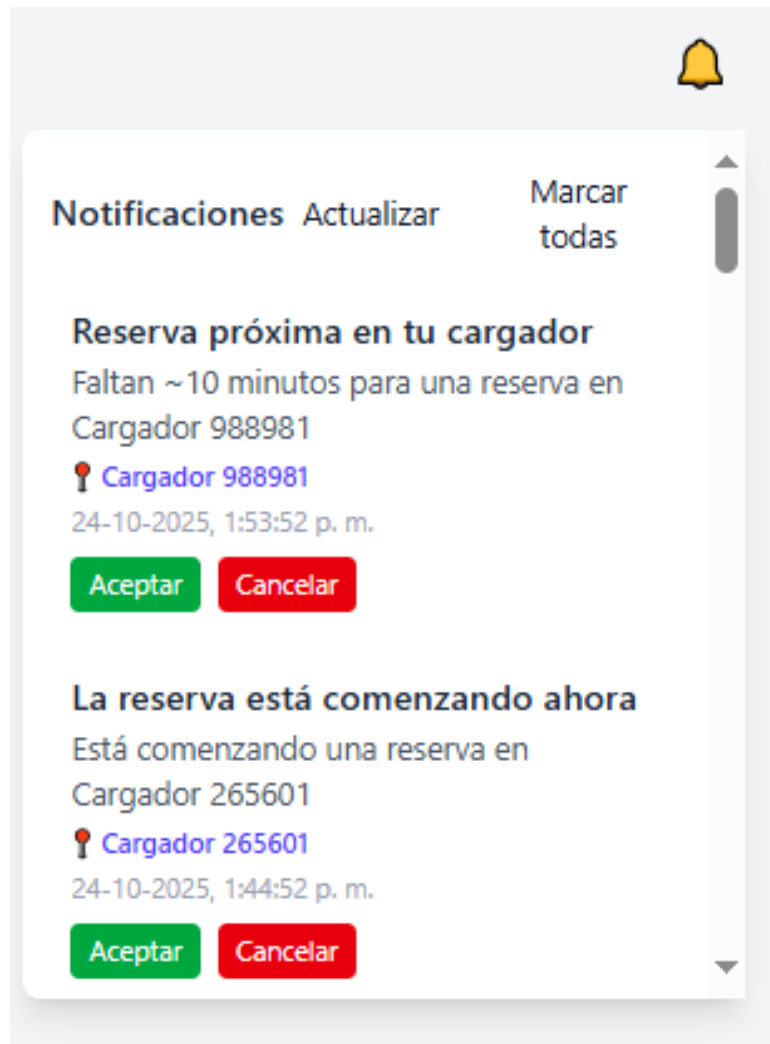
- PUT /api/auth/profile (ver sección 4.2)

Figura 4.5: Modificación de perfil del usuario.

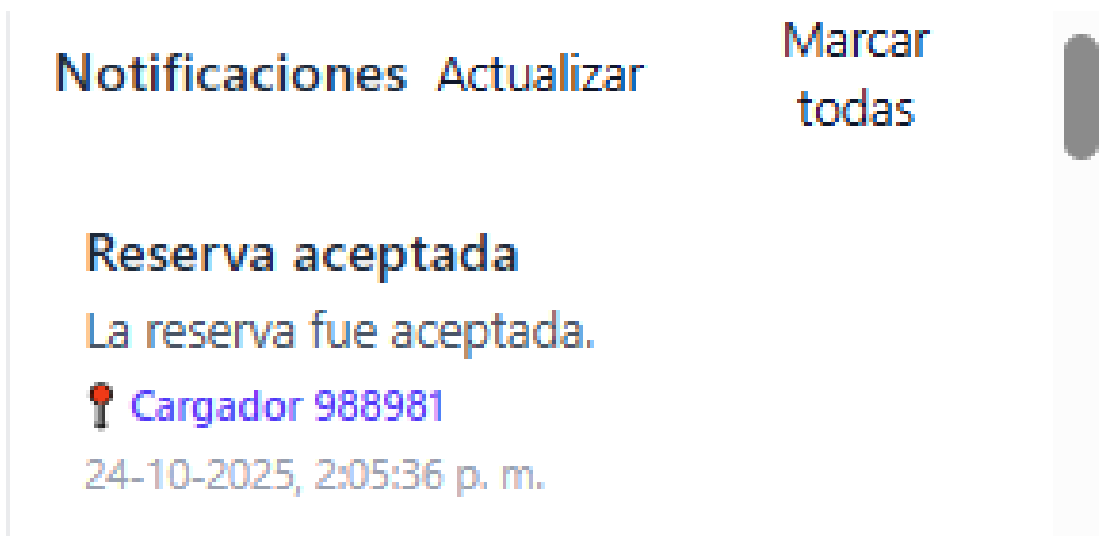
- **Panel de notificaciones:** Listado y campana con contador de pendientes. Tiene como funciones recibir avisos de reservas, confirmaciones de carga y alertas de tiempo de espera mediante WebSockets para actualizaciones en tiempo real y llamadas REST. Su diseño en escritorio se muestra en la barra superior, en cambio en Android se despliega como panel modal de pantalla completa para mayor visibilidad.

Lista de Endpoints utilizados:

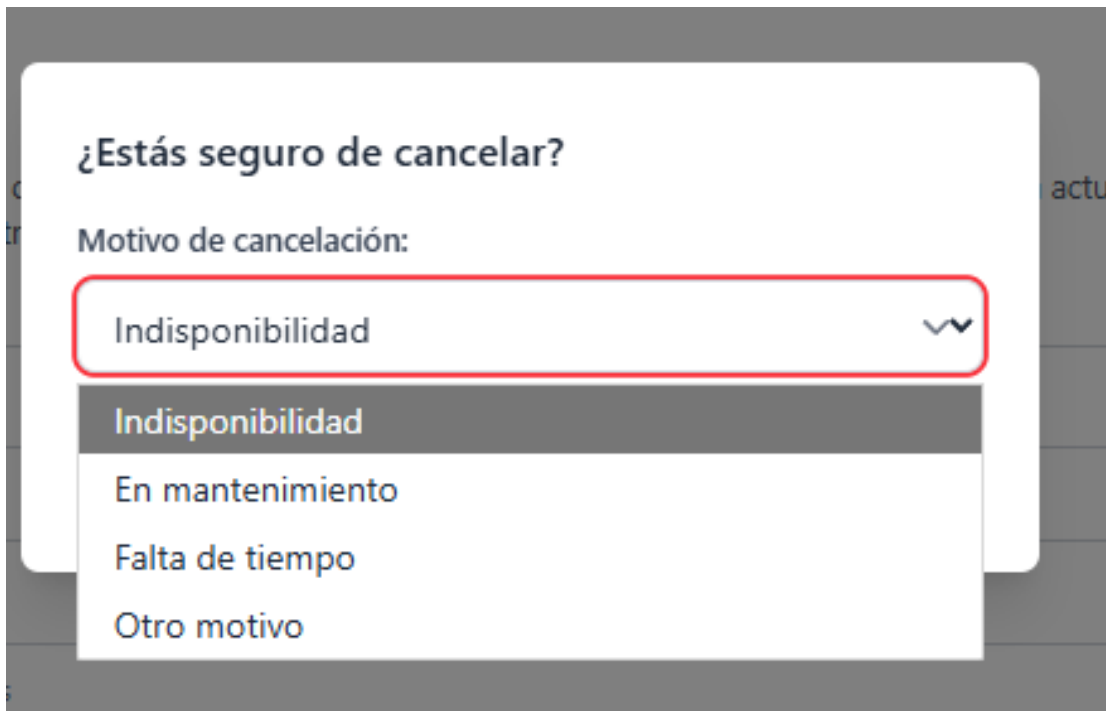
- GET /api/notifications (ver sección 4.2)
- POST /api/notifications/:id/read (ver sección 4.2)
- POST /api/notifications/read-all (ver sección 4.2)
- DELETE /api/notifications/:id (ver sección 4.2)



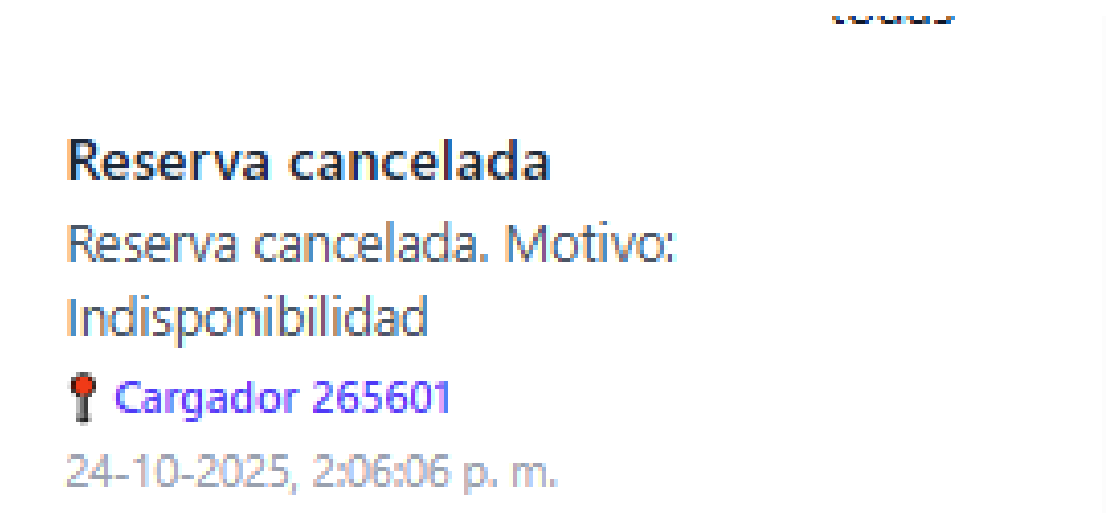
(a) Panel de notificaciones.



(b) Notificación de reserva aceptada.



(c) Confirmación de cancelación con motivo.



(d) Notificación de reserva cancelada.

Figura 4.6: Funcionamiento de panel de notificaciones.

Uso de administrador de la aplicación

- **Tablero de administración general:** Vista para supervisar usuarios, roles y salud de la plataforma. Tiene como funciones permitir filtrar y buscar usuarios, revisar estaciones de carga registradas y permitir acceso rápido a gestión detallada. Su diseño en escritorio consta de una barra de navegación lateral fija y tarjetas de resumen, en cambio en Android se prioriza una navegación por pestañas y listas condensadas para facilitación de la navegación.

Bibliotecas utilizadas:

- **Fuse.js:** Biblioteca utilizada para formar un buscador inteligente de usuarios por nombre o correo electrónico.

Lista de Endpoints utilizados:

- GET /api/users (ver sección 4.2)
- GET /api/chargers (ver sección 4.2)
- GET /api/stats/overview (ver sección 4.2)
- GET /api/stats/charger-types (ver sección 4.2)
- GET /api/stats/energy-monthly (ver sección 4.2)
- GET /api/stats/top-energy-vehicles (ver sección 4.2)
- GET /api/stats/usage-by-hour (ver sección 4.2)

Resumen del sistema

Indicadores globales recopilados desde la API.

Actualizar

VEHÍCULOS REGISTRADOS

85

ESTACIONES ACTIVAS

105

SESIONES HISTÓRICAS

3.395

ENERGÍA ENTREGADA

19.723,7 kWh

RESERVAS TOTALES

1.016

Distribución por tipo de cargador

Top tipos

CCS (Combo)	38 - 36.2%
Mennekes (Tipo 2)	30 - 28.6%
Tesla Supercharger	20 - 19.0%
J1772 (Tipo 1)	11 - 10.5%
CHAdEMO	6 - 5.7%

Energía por mes

MES	ENERGÍA (KWH)	SESIONES
Oct 2020	464,1	95
Sept 2020	4.408	761
Ago 2020	3.991,7	673
Jul 2020	3.452,2	568
Jun 2020	2.296,1	416

Vehículos con mayor consumo

#	VEHÍCULO	ENERGÍA (KWH)	SESIONES
1	Vehículo 35897499 CCS	1.013,3	170
2	Vehículo 98345808 CHAdEMO	1.006,1	192
3	Vehículo 90692118 CCS	870,9	91
4	Vehículo 78908148 Tesla	808,7	54

Uso por hora del día

HORA	SESIONES	ENERGÍA (KWH)
00:00	35	127,6
01:00	9	53,3
02:00	6	53,3
03:00	2	20
04:00	5	64,7

(a) Estadísticas generales de la aplicación.

The screenshot displays a user management dashboard. At the top left, the user is identified as 'Administrador Principal' (App Admin). The main area features a search bar for users and a filter dropdown set to 'Todos'. Below this, a list of users is shown, with the 'Administrador Principal' user highlighted. The detailed view for this user shows 0 vehicles and 0 stations, and includes sections for user data, reservations, and session history, each with an 'Actualizar' button.

Nombre	Rol	Vehículos	Estaciones
Administrador Principal	app admin	0	0
Administrador de Estaciones	station admin	0	105
Ana Rodríguez	ev user	1	0
Laura Sánchez	ev user	1	0
Juan Pérez	ev user	1	0
Ana Rodríguez	ev user	1	0
Diana Fernández	ev user		

(b) Vista principal de gestión de usuarios.

Figura 4.7: Tablero de administración general.

- **Gestión detallada:** Módulo para alta/baja/edición de usuarios, estaciones y vehículos. Tiene como funciones proporcionar formularios validados, mostrar confirmaciones antes de eliminar y sincronizar estados de entidades. Su diseño en escritorio consta de formularios de dos columnas y paneles colapsables, en cambio en Android presenta un flujo paso a paso con campos a pantalla completa para mejor experiencia de usuario.

Lista de Endpoints utilizados:

- PATCH /api/users/:id/role (ver sección 4.2)
- POST /api/chargers (ver sección 4.2)
- PATCH /api/chargers/:id/name (ver sección 4.2)
- DELETE /api/chargers/:id (ver sección 4.2)
- POST /api/vehicles (ver sección 4.2)
- PATCH /api/vehicles/:id (ver sección 4.2)
- DELETE /api/vehicles/:id (ver sección 4.2)

Ana Rodríguez 1 0
user35897499@evcharging.com Vehículos Estaciones

Datos del usuario ^

Nombre

Correo electrónico

Nueva contraseña

[Restablecer](#) [Guardar cambios](#)

Vehículos registrados ^ [Agregar vehículo](#)

Vehículo 35897499 [Eliminar](#)

Tipo de cargador: CCS
 Capacidad: 30 kWh
 Nivel actual: 5%

(a) Vista general de edición de un usuario.

Reservas del usuario ^ [Actualizar](#)

Cargador 425476 upcoming

Vehículo 35897499 · CCS

Inicio: 27-09-2025, 10:55 a. m.
 Fin estimado: 27-09-2025, 6:08 p. m.

[Cancelar reserva](#)

Cargador 711566 upcoming

Vehículo 35897499 · CCS

Inicio: 28-09-2025, 10:09 a. m.
 Fin estimado: 28-09-2025, 7:00 p. m.

[Cancelar reserva](#)

Cargador 300866 upcoming

Vehículo 35897499 · CCS

Inicio: 29-09-2025, 12:37 p. m.
 Fin estimado: 29-09-2025, 7:59 p. m.

[Cancelar reserva](#)

Cargador 894409 upcoming

Vehículo 35897499 · CCS

Inicio: 29-09-2025, 6:24 p. m.
 Fin estimado: 30-09-2025, 2:48 a. m.

[Cancelar reserva](#)

Cargador 738900 upcoming

Vehículo 35897499 · CCS

Inicio: 30-09-2025, 5:10 p. m.
 Fin estimado: 01-10-2025, 1:25 a. m.

[Cancelar reserva](#)

(b) Vista de edición de reservas en la sección de edición de un usuario.

Figura 4.8: Funcionamiento de la gestión detallada del administrador de la aplicación.

Uso de dueño de estaciones de carga

- **Panel de estaciones:** Vista principal con listado filtrable de cargadores y mapa sincronizado. Tiene como funciones permitir búsqueda por texto o ubicación, aplicar filtros por estado y tipo de conector, y permitir selección de un cargador para visualizar sus detalles. Su diseño en escritorio usa dos columnas (lista + mapa interactivo), en cambio en Android usa lista prioritaria y mapa colapsable con gestos para optimizar el espacio de pantalla.

A Administrador de Estaciones
Station Admin

Cargadores

Editar datos de perfil

Cerrar sesión

Gestión de Cargadores

Administra tus estaciones sin desplazamientos adicionales.

Actualizar

Estaciones de Carga

105 cargadores registrados

Filtro por estado: Todos los estados

Filtro por tipo: Todos los tipos

Modo de búsqueda: **Por nombre** Por ubicación

Buscar por nombre...

Cargador 582873 Disponible CCS

2.38 kW CCS

Gráficas Historial Calendario

-33.4392, -71.6201 [Ver en mapa](#) [Ver detalles](#)

Cargador 549414 Disponible CCS

2.10 kW CCS

Gráficas Historial Calendario

Mapa de Cargadores

105 cargadores mostrados en el mapa

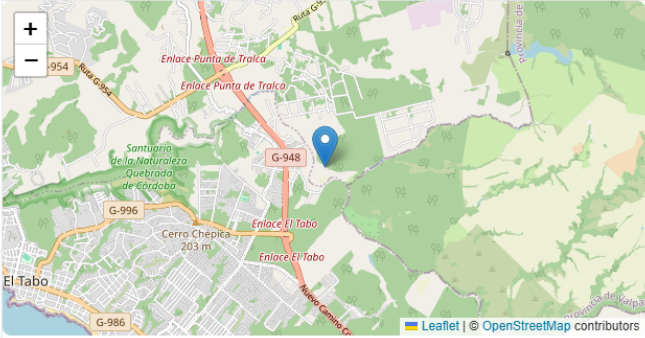


Figura 4.9: Vista general del panel de estaciones de carga.

Componentes del panel:

- **Tarjetas de cargador:** Tarjetas presentes en el panel para acceder a información rápida sobre cada cargador disponible para el administrador de cargadores. Tienen como funciones proporcionar accesos rápidos a gráficas, historial de carga y calendario de reservas, permitir centrar cargador en el mapa, mostrar ubicación aproximada y permitir extender detalles del cargador. Su diseño presenta una subtarjeta expandible con información sobre el tipo de conector, potencia máxima de carga, ocupación actual de la aplicación, ubicación exacta de la estación y opciones para cambiar el nombre del cargador en vez del nombre por defecto.

Lista de Endpoints utilizados:

- GET /api/chargers (ver sección 4.2)
- GET /api/chargers/:id (ver sección 4.2)
- PATCH /api/chargers/:id/name (ver sección 4.2)
- GET /api/realtime/chargers (ver sección 4.2)



(a) Tarjeta de cargador con información rápida y detallada.



(b) Cambio de nombre de cargador.

Figura 4.10: Funcionalidad de tarjetas de cargador.

- **Buscador de cargadores:** Sección que permite el filtrado y búsqueda dinámica de cargadores. Tiene como funciones proporcionar filtrado por sección de estado (Todos, disponible, ocupado, en mantenimiento), aplicar filtro por tipo (Todos, Type1, Type2, CCS, CHAdeMO, Tesla) y ofrecer dos modos de búsqueda: modo por ubicación que permite buscar ubicaciones con recomendaciones de nombre y otorgar una lista descendente de cargadores más cercanos a la ubicación descrita, y buscador por nombre que permite buscar un cargador específico por su nombre. Su diseño permite que automáticamente estas búsquedas y filtros afecten al mapa de la aplicación cambiando sus resultados visibles en tiempo real y centrando automáticamente el primer resultado de la búsqueda en el centro del mapa.

Bibliotecas utilizadas:

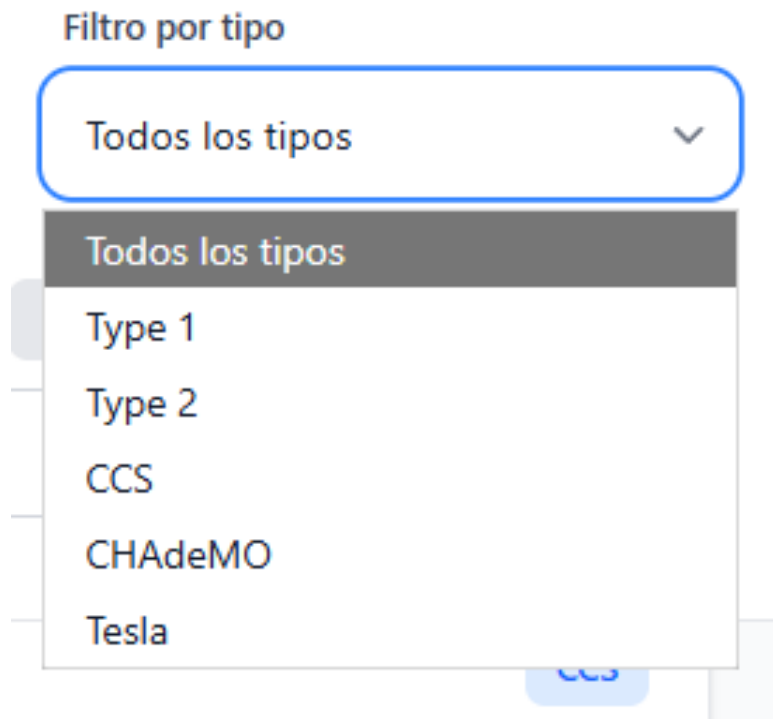
- **Fuse.js:** Biblioteca utilizada para formar un buscador inteligente de estaciones de carga y un buscador por cercanía de ubicación.

Lista de Endpoints utilizados:

- GET /api/chargers (ver sección 4.2)
- POST /api/recommendations (ver sección 4.2)

The image shows a user interface for filtering search results. It features two main sections: 'Filtro por estado' and 'Filtro por tipo'. The 'Filtro por estado' section has a dropdown menu currently set to 'Ocupado', with a list of options below it: 'Todos los estados', 'Disponible', 'Ocupado', and 'Mantenimiento'. The 'Filtro por tipo' section has a dropdown menu set to 'Todos los tipos'. Below these filters, there is a button labeled 'Por ubicación' and a search input field.

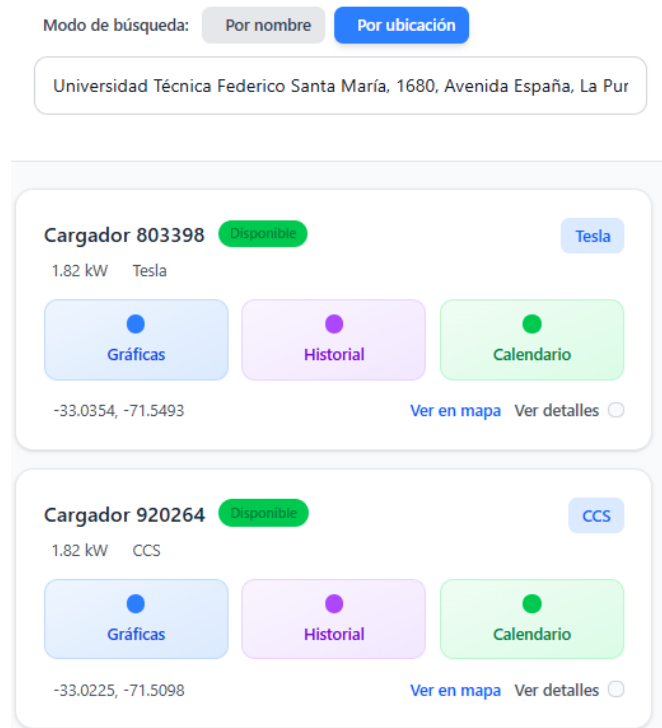
(a) Filtro por ocupación.



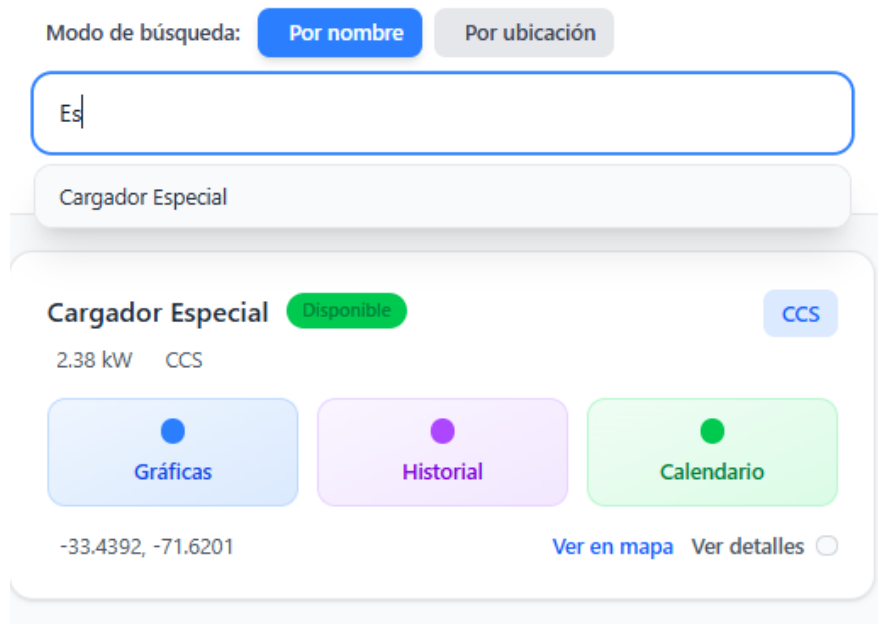
(b) Filtro por modelo de cargador.



(c) Búsqueda por ubicación (recomendaciones).



(d) Búsqueda por ubicación (lista de cargadores cercanos).



(e) Búsqueda por nombre.

Figura 4.11: Funcionalidades del buscador de cargadores.

- **Mapa de cargadores:** Mapa interactivo que permite la visualización dinámica de los cargadores disponibles para el administrador de estaciones de carga. Tiene como funciones mostrar cargadores asignados a modo de muestra a lo largo de la región de Valparaíso, responder dinámicamente a búsquedas y centrados en el mapa, permitir centrar cargadores en la lista de cargadores cuando se selecciona un punto específico y visualizar coordenadas o posición aproximada de cada estación. Su diseño presenta marcadores interactivos que se sincronizan con la lista de tarjetas y responden a filtros y búsquedas en tiempo real.

Bibliotecas utilizadas:

- **Leaflet:** Biblioteca utilizada para generar un mapa de los cargadores disponibles del administrador de estaciones de carga, con marcadores de ubicación añadidos y responsivo a búsquedas e interacciones externas del usuario.

Lista de Endpoints utilizados:

- GET /api/chargers (ver sección 4.2)
- GET /api/realtime/chargers (ver sección 4.2)

Mapa de Cargadores

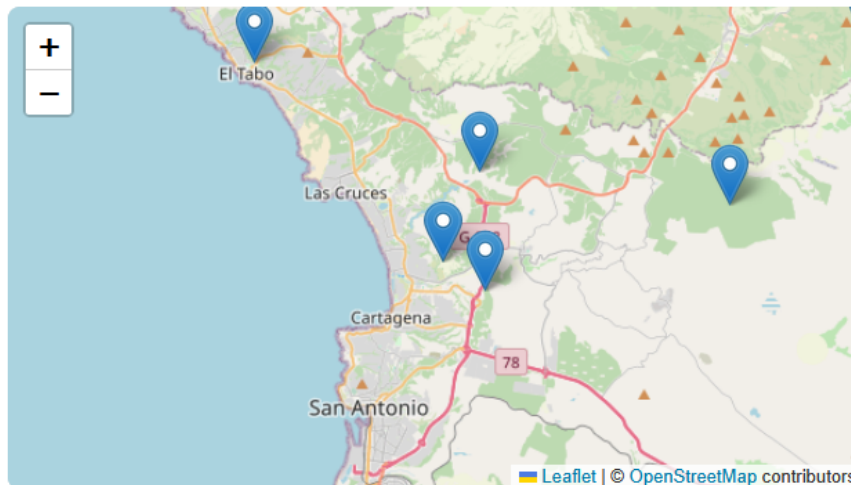


Figura 4.12: Mapa interactivo de cargadores con marcadores de ubicación.

- **Calendario de reservas por cargador (Administrador de CS):** Agenda visual de horarios ocupados y disponibles. Tiene como funciones permitir navegación por mes y semana, mostrar vista detallada de reservas y proporcionar acceso rápido al historial. Su diseño en escritorio muestra cuadrículas amplias con efectos hover, en cambio en Android muestra tarjetas compactas y scroll vertical para facilitar la navegación táctil.

Bibliotecas utilizadas:

- **date-fns:** Biblioteca utilizada para calcular todo tipo de fechas disponibles en el calendario.

Lista de Endpoints utilizados:

- GET /api/calendar/:chargerId (ver sección 4.2)
- GET /api/reservations (ver sección 4.2)
- POST /api/reservations (ver sección 4.2)
- POST /api/reservations/:id/accept (ver sección 4.2)
- POST /api/reservations/:id/cancel (ver sección 4.2)
- DELETE /api/reservations/:id (ver sección 4.2)

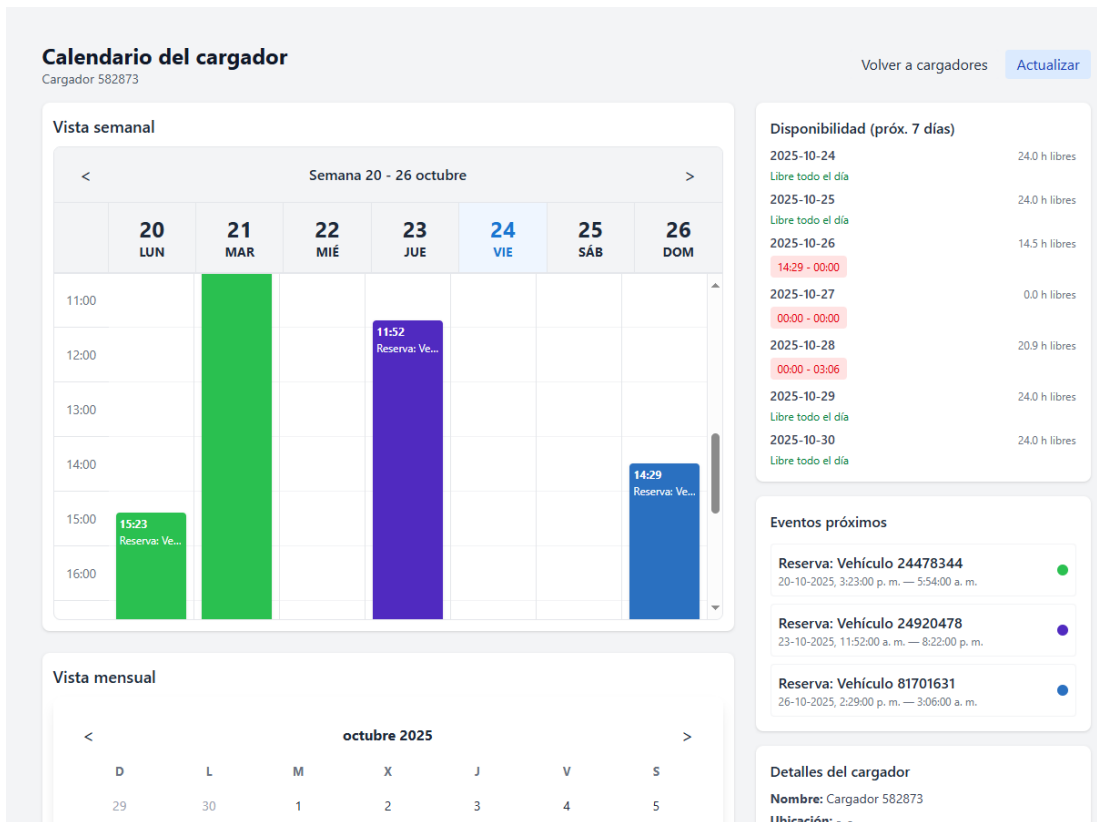


Figura 4.13: Calendario de planificación de reservas de estaciones de cargadores de vehículos eléctricos de estaciones de carga.

- Historial de carga:** Submenú diseñado para permitir al administrador de estaciones de carga revisar información detallada de las sesiones. Tiene como funciones mostrar qué vehículo y qué cargador ha utilizado, desplegar fechas de inicio y fin de las cargas, visualizar energía entregada y calcular duración de cada carga. Su diseño en escritorio presenta tablas laterales con filtros incorporados, en cambio en Android muestra tarjetas compactas con scroll continuo para mejor visualización móvil.

Lista de Endpoints utilizados:

- GET /api/chargers/:id/usage-history (ver sección 4.2)

Historial de sesiones ^

Actualizar

VEHÍCULO	CARGADOR	INICIO	FIN	ENERGÍA (KWH)	DURACIÓN
Vehículo 35897499 CCS	Cargador 944575	01-10-2020, 5:16 p. m.	01-10-2020, 7:34 p. m.	6,3	137.50000002 min
Vehículo 35897499 CCS	Cargador 582873	01-10-2020, 12:06 p. m.	01-10-2020, 1:32 p. m.	3,4	85.60000002000001 min
Vehículo 35897499 CCS	Cargador 944575	01-10-2020, 11:21 a. m.	01-10-2020, 12:01 p. m.	2	39.13333332 min
Vehículo 35897499 CCS	Cargador 280221	21-09-2020, 11:19 a. m.	21-09-2020, 1:19 p. m.	4,2	120 min
Vehículo 35897499 CCS	Cargador 801274	18-09-2020, 4:36 p. m.	18-09-2020, 7:02 p. m.	6,4	146.06666664 min
Vehículo 35897499 CCS	Cargador 738900	17-09-2020, 5:25 p. m.	17-09-2020, 5:57 p. m.	2,3	31.666666680000002 min
Vehículo 35897499 CCS	Cargador 280221	14-09-2020, 5:28 p. m.	14-09-2020, 8:48 p. m.	6,6	199.24999997999998 min

Figura 4.14: Historial de sesiones de carga con detalles de vehículos, fechas, energía y duración.

- Gráficas de Estación de carga:** Gráficas que muestran estadísticas de ocupación de la estación de carga seleccionada y un resumen de la energía entregada por cada sesión, permitiendo interacción del usuario. Tiene como funciones permitir zoom y desplazamiento (pan) en los gráficos, aplicar filtros por periodo temporal y realizar exportaciones básicas de datos. Su diseño en escritorio presenta gráficos grandes e interactivos con todas las funcionalidades disponibles, en cambio en Android muestra gráficos adaptados al tamaño de pantalla para mantener la legibilidad y funcionalidad.

Bibliotecas utilizadas:

- **Chart.js y plugin de zoom:** Biblioteca utilizada para diseñar los diferentes tipos de gráficos en conjunto con la opción de zoom para permitir mejor visualización de los datos.

Lista de Endpoints utilizados:

- GET /api/chargers/:id/usage-stats (ver sección 4.2)

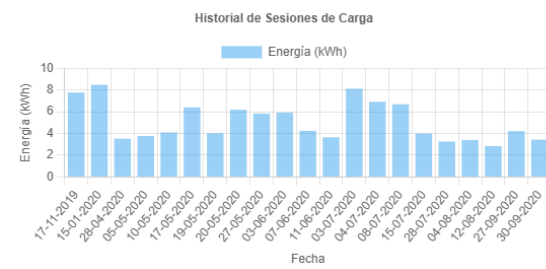
Gráficas del Cargador

[Volver a cargadores](#)

Historial de Ocupación



Historial de Sesiones de Carga



Energía Total
106.87 kWh

Sesiones
22

Tiempo Promedio
2.2 h

Figura 4.15: Gráficas interactivas de ocupación y consumo energético de la estación de carga.

- **Gestión de sesiones activas:** Panel para monitorear y controlar cargas en curso mediante simulador. Tiene como funciones permitir crear vehículos simulados, iniciar procesos de carga y visualizar progreso de carga en tiempo real con actualizaciones dinámicas. Su diseño en escritorio consta de un panel modal grande con badges de estado claramente visibles, en cambio en Android presenta un modal de pantalla completa con botones táctiles grandes para facilitar la interacción.

Lista de Endpoints utilizados:

- POST /api/simulator/create (ver sección 4.2)
- POST /api/simulator/start (ver sección 4.2)
- GET /api/simulator/status (ver sección 4.2)
- GET /api/vehicles (ver sección 4.2)

Uso de ocupante de EV

- **Tablero personal de carga:** Vista donde el usuario puede realizar las reservas de las estaciones de carga. Tiene dos columnas: la primera donde se puede realizar las reservas con los sistemas implementados o directamente en el mapa, y la segunda donde se puede ver la lista de las próximas reservas y también las estaciones favoritas.

Lista de Endpoints utilizados:

- GET /apichargers/nearby (ver sección 4.2)
- GET /api/vehicles/:id/actual (ver sección 4.2)
- GET /api/vehicles/:id/charging-history (ver sección 4.2)
- GET /api/charging-sessions/active/by-reservation/:reservationId (ver sección 4.2)
- POST /api/reservations/:id/cancel (ver sección 4.2)

M María González
Ev User

Vehículo seleccionado
Vehículo 358974

NIVEL DE CARGA: 44%
CAPACIDAD: 99 kWh

MODELO: Vehículo 35897499
TIPO DE CARGADOR: Tesla

Reservar

Historial de cargas

Editar datos de perfil

Cerrar sesión

¡Buenos días! ¿Dónde quieres cargar?

Planifica otra carga como encuentres necesario.

Reservar Cargador

Reserva Rápida Reserva Específica

[Mostrar preferencias](#)

Mapa de cargadores

Próximas reservas Mis Favoritos

JUE 26 feb.
Cargador 623134
Carga de Vehículo Tipo Tesla

10:17 - 11:37

Duración estimada: 7h 55m

Centrar en mapa Cancelar

JUE 26 feb.
Cargador 256233
Carga de Vehículo Tipo Tesla

12:11 - 20:06

Duración estimada: 7h 55m

Centrar en mapa Cancelar

DOM 1 mar.
Cargador 236840
Carga de Vehículo Tipo Tesla

09:36 - 11:37

Duración estimada: 26h 1m

Centrar en mapa Cancelar

DOM 1 mar.

Notificaciones Actualizar Marcar todas

Reserva cancelada
Reserva cancelada. Motivo: Indisponibilidad
Cargador 431796
24/2/2026, 12:00:41

Reserva cancelada
Reserva cancelada. Motivo: Indisponibilidad
Cargador 801274
24/2/2026, 12:00:39

Reserva cancelada
Reserva cancelada. Motivo: Indisponibilidad
Cargador 430455

Figura 4.16: Pantalla de reservas.

Componente de la vista:

- **Mapa de cargadores:** Mapa interactivo donde el usuario puede ver todas las estaciones de carga disponibles. Se puede ver la ubicación de cada estación, junto con su información dura (potencia de carga, el costo de la energía, tipo de cargador, coordenadas). Seleccionando la estación el usuario puede agregarla como estación favorita para poder acceder a ella más fácilmente, y puede acceder a su horario de reservas apretando el botón “Reservar”. *Bibliotecas utilizadas:*

- **Leaflet:** Biblioteca utilizada para generar un mapa de los cargadores disponibles del administrador de estaciones de carga, donde el usuario puede interactuar con cada estación.

Lista de Endpoints utilizados:

- GET /api/favourites/:userId (ver sección 4.2)
- POST /api/favourites/:userId/:stationId (ver sección 4.2)
- DELETE /api/favourites/:userId/:stationId (ver sección 4.2)



Figura 4.17: Estación seleccionada en el mapa.

- **Modal de opciones:** Dentro de esta ventana, el usuario tiene la opción de seleccionar que tipo de recomendación quiere (por carga o por tiempo) o puede reservar directamente una estación en específico e ir a su calendario. Además, para la opción de reserva rápida, el usuario puede ajustar sus preferencias para ajustar los pesos del algoritmo de recomendación. *Lista de Endpoints utilizados:*

- GET /api/recommendations/recommend (ver sección 4.2)
- GET /api/chargers/nearby (ver sección 4.2)
- GET /api/favourites/:userId (ver sección 4.2)
- POST /api/reservations (ver sección 4.2)

¡Buenos días! ¿Dónde quieres cargar?

Planifica otra carga como encuentres necesario.

Reservar Cargador

Reserva Rápida

Reserva Específica

[Mostrar preferencias](#) ?

Propuesta de reserva:

Inicio: 25/2/2026, 13:27:18

Fin: 26/2/2026, 5:50:00

Tiempo estimado de carga: 983 minutos

Tiempo estimado de llegada (ETA): 11 minutos

Estación: Cargador 955429

Potencia: 2.17 kW

Costo estimado de la carga: CLP\$ 13.259

Aceptar

Cancelar

Otra Recomendación

(a) Reserva rápida.

¡Buenos días! ¿Dónde quieres cargar?

Planifica otra carga como encuentres necesario.


Reservar Cargador

Reserva Rápida

Reserva Específica

[Mostrar preferencias](#) ?

Selecciona una estación de carga:

★ Cargador 955429 (2.176059143022068 kW) 

Ir a calendario de reserva

Cancelar

(b) Reserva específica.

Figura 4.18: Opciones disponibles para la reserva de estación desde el modal.

[Ocultar preferencias](#) ?

Porcentaje objetivo de batería (%)

80

Distancia Costo

- + - +

Tiempo de carga Demora

- + - +

Modo de búsqueda

Por Carga Por Tiempo

(a) Recomendación basada en nivel de carga.

[Ocultar preferencias](#) ?

Tiempo disponible (minutos)

30

Distancia Costo

- + - +

Tiempo de carga Demora

- + - +

Modo de búsqueda

Por Carga Por Tiempo

(b) Recomendación basada en tiempo.

Figura 4.19: Preferencias disponibles en la opción de reserva rápida.

- **Columna de próximas reservas/Mis favoritos:** En la parte derecha del panel, se tiene una columna que te da la opción de visualizar tanto las próximas reservas del vehículo como las estaciones favoritas. En esta última pestaña, uno puede dirigirse directamente al calendario de reservas o quitarla de favoritos.

Lista de Endpoints utilizados:

- GET /api/favourites/:userId (ver sección 4.2)
- DELETE /api/favourites/:userId/:stationId (ver sección 4.2)

Próximias reservas Mis Favoritos Ver todas

JUE 26
feb.

Cargador 623134
Carga de Vehículo Tipo Tesla

10:17 - 18:03
Duración estimada: 7h 46m

[Centrar en mapa](#) [Cancelar](#)

JUE 26
feb.

Cargador 256233
Carga de Vehículo Tipo Tesla

12:11 - 20:06
Duración estimada: 7h 55m

[Centrar en mapa](#) [Cancelar](#)

DOM 1
mar.

Cargador 236840
Carga de Vehículo Tipo Tesla

09:36 - 11:37
Duración estimada: 26h 1m

[Centrar en mapa](#) [Cancelar](#)

(a) Pestaña de próximas reservas.

Próximias reservas Mis Favoritos

NOMBRE DE LA ESTACIÓN	DIRECCIÓN (POSICIÓN)	ACCIONES
Cargador 955429	-33.0114, -71.5352	Ir a calendario Quitar de Favoritos
Cargador 371335	-33.0703, -71.5136	Ir a calendario Quitar de Favoritos

(b) Estaciones en la lista de favoritos.

Figura 4.20: Panel derecho en la vista de ocupante EV.

- **Vista del historial de carga:** En esta pantalla se puede ver el historial de búsqueda de un vehículo. La fecha, el cargador que utilizó, la energía transferida y la duración de la carga.

Historial de carga			
FECHA	CARGADOR	ENERGÍA (KWH)	DURACIÓN (MIN)
1/10/2020	Cargador 944575	6.27	138
1/10/2020	Cargador 582873	3.43	86
1/10/2020	Cargador 944575	1.97	39
21/9/2020	Cargador 280221	4.21	120
18/9/2020	Cargador 801274	6.35	146
17/9/2020	Cargador 738900	2.27	32
14/9/2020	Cargador 280221	6.58	199
11/9/2020	Cargador 801274	6.31	213

Figura 4.21: Historial de carga de vehículo seleccionado.

- **Calendario de reservas por cargador (Ocupante de EV):** Aquí el usuario puede escoger un horario en específico de la estación de carga, teniendo la posibilidad de ver y analizar los tiempos en donde la CS está ocupada. Además, puede añadir/quitar la estación de la lista de favoritos, y ver el detalle de la estación, como la potencia, costos, etc... Al seleccionar el horario en donde quiere hacer la reserva, le da un detalle de su reserva. *Lista de Endpoints utilizados:*

- GET /api/calendar/charger/:chargerId (ver sección 4.2)
- GET /api/favourites/:userId (ver sección 4.2)
- POST /api/reservations (ver sección 4.2)
- POST /api/favourites/:userId/:chargerId (ver sección 4.2)
- DELETE /api/favourites/:userId/:chargerId (ver sección 4.2)

Reservar cargador
 Cargador 955429 ★
 Vehículo: Vehículo 35897499 (Tesla)
 Volver

Selecciona el horario de tu reserva
 Haz click en la hora de inicio en el calendario

Semana 23 - 1 marzo

	23 LUN	24 MAR	25 MIÉ	26 JUE	27 VIE	28 SÁB	1 DOM
11:00							
12:00							
13:00							
14:00			14:21 Reserva: V...	13:42 Reserva: V...	13:00 Tu reserva		
15:00							
16:00							
17:00							

Horario seleccionado

Fecha y hora de inicio: 27-02-2026 13:00

Fecha y hora de fin: 28-02-2026 16:00

Duración de la reserva: 27 horas

Porcentaje alcanzable: 100% en 25 horas con 1 minutos.

Costo estimado de la carga: CLP\$ 20.624

Tiempo de estacionamiento: 91 minutos

Costo de estacionamiento: CLP\$ 2730

Costo Total: CLP\$ 23.354

Confirmar reserva

Limpiar selección

Detalles del cargador

Nombre: Cargador 955429

Tipo: Tesla

Potencia: 2.176059143022068 kW

Costo energía: CLP\$ 372 / kWh

Costo estacionamiento: CLP\$ 30 / min

Ubicación: -, -

Figura 4.22: Calendario de reserva por estación.

4.4. Algoritmo de recomendación

La opción de reserva rápida hace que el servidor busque las mejores opciones de cargador y horario para el usuario. Todo esto con un sistema de preferencias la cual le permite escoger el tipo de recomendación que quiere y que aspectos de la carga le interesan más (lejanía del cargador, tiempo de espera para la carga, tiempo de demora de la carga o costo de la energía). Esta opción busca una brecha en el tiempo de reserva de cada estación que pueda satisfacer las necesidades del usuario, después de encontrar una brecha le realiza una evaluación y le entrega al usuario un ranking de propuestas, donde puede aceptar la que más le acomode.

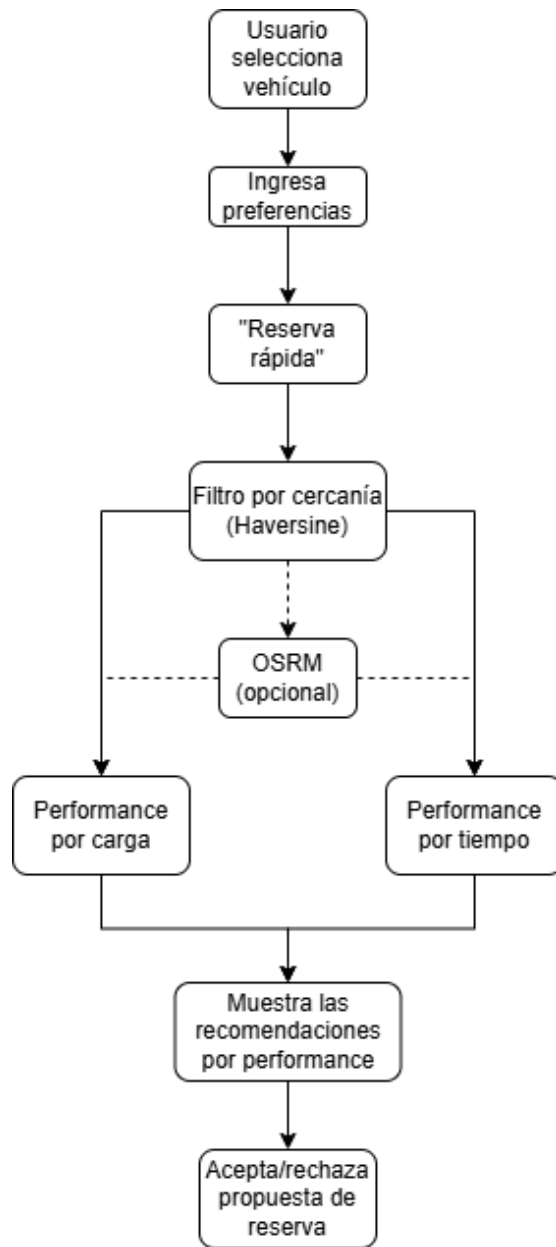


Figura 4.23: Diagrama de flujo para la recomendación de reserva.

4.4.1. Open Source Routing Machine (OSRM)

Para no sobrecargar el servidor con solicitudes y cálculos innecesarios, no se consideran todas las estaciones de carga para realizar las recomendaciones. Lo primero es que se filtran por cercanía con la fórmula Haversine (distancia de dos puntos en una es-

fera) y así truncan las posibles estaciones. Este método nos permite dos cosas: obtener la distancia entre la estación y el usuario, y estimar cuanto es el tiempo (con velocidad media) que le tomaría al usuario llegar a la estación de carga. Esto último es útil para saber desde cuando debe empezar la reserva del usuario en la estación. Aunque si bien la fórmula de Harvesine nos ayuda a obtener una estimación de los datos indicados, no representan la distancia en la trayectoria (calles o carretera) que cruza el automóvil para llegar a la estación. Para esto se utilizan distintas herramientas de navegación como Waze, Google Maps, etc. Que le dan indicaciones al conductor por donde irse y cuanto tiempo le tomará. Las herramientas gratuitas que otorgan estos servicios son para utilizar sus aplicaciones y no para integrar el cálculo de las rutas en una aplicación de terceros. Para sortear esto, se decidió utilizar OSRM, el cual es un servicio gratuito que tiene disponible su API a libre disposición y uso, el problema es que tiene un límite de consultas que se pueden generar en ciertos instantes de tiempo y funciona de manera dinámica, por lo que es imposible realizar un cálculo o algoritmo que facilite u optimice las consultas; esto sin contar que no es una solución escalable.

Por otro lado, los desarrolladores también dan la opción de uno levantar su propio servidor OSRM con Docker. Siguiendo los pasos que están en su página, se descargó el mapa de Chile y se levantó un contenedor OSRM local para realizar los cálculos y dejar la plataforma lista para ocupar este servicio. Lamentablemente ocupa una cantidad considerable de RAM para procesar la información de los mapas y realizar el ruteo, por lo que no se pudo levantar en la nube (más de 10GB de memoria). El motor de ruteo OSRM puede desplegarse en infraestructura cloud tipo IaaS (AWS EC2 o Azure VM), requiriendo aproximadamente 32–64 GB de memoria RAM para procesamiento regional. El costo operacional estimado varía entre USD 500 y USD 2000 mensuales dependiendo del proveedor y capacidad asignada. Finalmente, con respecto al cálculo de la distancia y el tiempo de llegada, la plataforma se dejó configurada para intentar conectarse al servidor local de OSRM, y en caso de no tener respuesta, utilizar la fórmula de Harvesine.

4.4.2. Performance

Para poder recomendar una estación de carga, son importantes varios aspectos. En [16], se propone la ecuación de *Performance* (2.1), donde se toma en cuenta la distancia entre la estación y el usuario, el tiempo de espera que tiene el usuario para utilizar la estación de carga, el costo de la energía en \$/kWh y lo que se demora en realizar la carga la estación (relacionada a la potencia de la misma). El algoritmo, evaluado bajo los criterios dados en [32], es un emparejamiento implícito usuario-objeto básico. Esto significa que es un emparejamiento en donde el usuario no coloca una lista de preferencias sino que el sistema busca por él cuales se adecúan, además de que sigue un sistema de *score* que se asigna mediante una fórmula. Aunque para aplicar este algoritmo es necesario aplicar filtros de datos duros, como la lejanía de la estación (para reducir el costo de cómputo) y el tipo de cargador, que se vuelve fundamental para que la estación sea útil para el usuario. Si bien Chile busca que las estaciones de carga puedan ser utilizadas por cualquier tipo de vehículo, aún no se implementa del todo y no es el caso de las estaciones de carga privada; las cuales no tienen que seguir las estandarizaciones que propone e impulsa el gobierno. Para emparejar se hicieron dos casos de uso:

1. **Carga objetivo:** En este caso, el usuario quiere cargar su batería hasta cierto porcentaje de carga. Se sigue utilizando la fórmula 2.1 con las restricciones/filtros mencionados:

$$P_i = \left(\frac{w_1}{\bar{w}} \frac{d_i}{\max_{i \in \{1, \dots, N\}} d_i} + \frac{w_2}{\bar{w}} \frac{t_i}{\max_{i \in \{1, \dots, N\}} t_i} + \frac{w_3}{\bar{w}} \frac{s_i}{\max_{i \in \{1, \dots, N\}} s_i} + \frac{w_4}{\bar{w}} \frac{c_i}{\max_{i \in \{1, \dots, N\}} c_i} \right)$$

w_x Peso de la variable correspondiente (distancia, duración de la carga, tiempo de espera y costo de la energía respectivamente)

\bar{w} Suma de todos los pesos

d_i Distancia de la estación i al usuario

t_i Tiempo el que se demora en realizarse la carga en la estación i

s_i Tiempo de espera para utilizar la estación i

c_i Costo de la energía en $\frac{CLP\$}{kWh}$

N Número total de estaciones que se están evaluando

La variable c_i es un dato duro y característica de cada estación, por lo que no depende del usuario. Mientras que d_i se calcula con la fórmula de Haresine o con OSRM si está disponible; s_i depende de la brecha que encuentre para hacer la reserva (incluye el tiempo de traslado hacia la estación). Por otra parte t_i se calcula con una fórmula simple:

$$t_i = \frac{CP * (PO - PA)}{PC} \quad (4.1)$$

CP Capacidad de la batería del vehículo en kWh

PO Porcentaje de batería deseado (objetivo)

PA Porcentaje de batería actual

PC Potencia de carga en kW

Para facilitar este cálculo se asume que la estación de carga trabaja con la potencia indicada en sus características por todo el proceso de carga y desprecia la interacción del BMS del EV y de la CS sobre este proceso. La estación que consiga el puntaje menor es la que es más adecuada según las preferencias, ya que todas las variables pueden considerarse como “no deseadas”, por lo tanto es mejor mientras más bajo sea el valor de cada una de ellas. Además, tienen valores entre 0 y 1, y se ocupa una normalización con respecto a los pesos, por lo que el peor puntaje posible termina siendo 1.

2. **Tiempo disponible:** En este caso, el usuario no busca cargar su vehículo hasta cierto punto, sino que ingresa cuanto tiempo dispone para cargar su vehículo. Este tiempo disponible pone una marca de hasta donde puede llegar la reserva. Por

lo tanto ahora el enfoque es otro, es aprovechar al máximo los minutos disponibles, tomando importancia la carga realizada en ese tiempo, y la utilización de la estación:

$$P_i = \left(\frac{w_1}{\bar{w}} \frac{d_i}{\max_{i \in \{1, \dots, N\}} d_i} + \frac{w_2}{\bar{w}} \frac{t_i}{\max_{i \in \{1, \dots, N\}} t_i} + \frac{w_3}{\bar{w}} \frac{s_i}{\max_{i \in \{1, \dots, N\}} s_i} + \frac{w_4}{\bar{w}} \frac{c_i}{\max_{i \in \{1, \dots, N\}} c_i} - \frac{w_5}{\bar{w}} \frac{e_i}{\max_{i \in \{1, \dots, N\}} e_i} \right) \quad (4.2)$$

e_i Energía dada por la estación i

En la ecuación 4.2 la energía entregada por la estación se resta, ya que esta se considera una “variable positiva” dentro de la función objetivo. A diferencia del otro caso de uso, t_i se calcula según 4.3; cabe destacar que en este caso, la ventana de tiempo está limitada por la capacidad de almacenaje de energía de la batería y el tiempo disponible. Además de que se ocupa la nueva variable e_i , que representa la energía capaz de entregar la estación durante la reserva, la cual se calcula con 4.4. Es importante que el peso de e_i es fijo e igual a 1, experimentalmente al colocarlo también como una preferencia (variable), no entregaba un ranking adecuado (no siempre recomendaba el que más energía daba).

$$tn_i = \frac{CP * (1 - PA)}{PC}$$

$$t_i = \min\{tn_i, \min\{FR, TD\} - IR\} \quad (4.3)$$

tn_i Tiempo necesario para que se realice una carga al 100 %

FR Hora del fin de la reserva

TD Suma de la hora de inicio de la solicitud más el tiempo disponible

IR Hora de inicio de la reserva

$$e_i = t_i * PC \quad (4.4)$$

Capítulo 5

Resultados

5.1. Despliegue de la plataforma

Se decide utilizar la plataforma de webhosting de “Render” para desplegar la plataforma dada su facilidad de uso y por sus planes gratuitos sin costos, dado que cuenta con comandos automatizados para compilar la aplicación en su plataforma facilita el mantenimiento, escalabilidad y prueba constante de la aplicación. Con el uso de éste webhoster sumado a la base de datos en línea de Mongo DB Atlas se encuentra a completa disposición una aplicación Web con capas de Backend y Frontend.

5.1.1. Frontend

El frontend se encuentra desplegado en un servicio de página web estática de “Render”, configurado para permitir la visualización de la interfaz gráfica de la aplicación tras compilar el código de React. La plataforma se encuentra vinculada directamente con el repositorio en GitHub lo que permite que tras cada cambio la aplicación se vuelva a desplegar automáticamente asegurando que el frontend esté siempre actualizado.

- **URL del Frontend:** <https://repositorio-memorias.onrender.com/>

Adicionalmente con el uso de la biblioteca “Capacitor” se realiza un port de la aplicación Web a una aplicación móvil para el sistema operativo Android

- **URL de la APK instalable:** <https://github.com/gabrielariasv/Repositorio-Memorias/releases/download/APK/app-release.apk>

5.1.2. Backend

El backend se encuentra desplegado en otro servicio estático de “Render”, no cuenta con ningún tipo de interfaz visible, se trata de una API web realizada con Javascript utilizando el framework de ExpressJS para todo tipo de peticiones a la API. su propósito es comunicar la base de datos generada en MongoDB Atlas que contiene toda la información relevante para la aplicación con el frontend. La plataforma se encuentra vinculada directamente con el repositorio en GitHub lo que permite que tras cada cambio la aplicación se vuelva a desplegar automáticamente asegurando que el frontend esté siempre actualizado.

- **URL del Backend (Sin vista web pero en funcionamiento):** <https://repositorio-memorias.onrender.com>

Capítulo 6

Conclusiones

Recapitulando, según los resultados, el proyecto ha logrado cumplir con las expectativas del desafío. Se implementa la plataforma web/móvil que considera tanto el monitoreo de estaciones de carga como la reserva de estas; logrando así dar una oportunidad a dueños de estaciones de carga a darle un uso comercial a sus cargadores y suplir la poca oferta que hay en este momento.

Los ocupantes de vehículos eléctricos tienen una plataforma fácil de acceder y ocupar, con mucha información a la mano y dos posibilidades de reserva:

Bibliografía

- [1] Lira, G. Modelos y precios de autos 100% eléctricos en Chile en 2025, 2025. <https://www.autofact.cl/blog/noticias/autofact/autos-electricos>.
- [2] Laborde, A. Chile pisa el acelerador en la compra de automóviles eléctricos: sus ventas aumentaron en 183% en un año, 2025. <https://elpais.com/chile/2025-01-08/chile-pisa-el-acelerador-en-la-compra-de-automoviles-electricos-s.html>.
- [3] Gerlach, N. Infraestructura en Chile: Descubre Cuántos Cargadores Hay Para Autos Eléctricos, 2025. <https://www.rutamotor.com/infraestructura-en-chile-descubre-cuantos-cargadores-hay-para-aut>
- [4] Energica.city. Instalación de cargadores para autos eléctricos, n.d. https://energica.city/instalacion_cargadores Recuperado el 18 de mayo de 2025.
- [5] Ministerio de Energía, Gobierno de Chile. Plataforma de Electromovilidad - Instalación de cargadores para vehículos eléctricos, n.d. <https://energia.gob.cl/electromovilidad/sistemas-de-carga/instalacion-de-puntos-de-carga> Recuperado el 18 de mayo de 2025.

- [6] La Tercera. Maestro chasquilla: guía básica para entender el circuito eléctrico en casa, 2023. <https://www.latercera.com/practico/noticia/maestro-chasquilla-guia-basica-para-entender-el-circuito-electrico-en-BBVXNUKM4ZERDD3ROZCRFNBVWM/>.
- [7] Superintendencia de Electricidad y Combustibles (S.E.C.). INSTALACIONES DE CONSUMO EN BAJA TENSIÓN, octubre 2003. https://www.sec.cl/sitioweb/electricidad_norma4/norma4_completa.pdf Recuperado el 19 de mayo de 2025.
- [8] Tienda Copec Voltex. Productos, n.d. <https://copecvoltex.cl/collections/all> Recuperado el 19 de mayo de 2025.
- [9] BESEN INTERNATIONAL GROUP CO., LTD. EV charging station manufacturer, 2024. <https://www.besen-group.com/evse-manufacturer/> Recuperado el 2025-12-12.
- [10] XCharge North America. XCharge north America - C7, n.d. <https://www.xcharge.us/product/c7> Recuperado el 19 de mayo de 2025.
- [11] Gallardo, L. Copec abre su primera estación de carga 100 % eléctrica en Chile, 2025. <https://www.latercera.com/mtonline/noticia/copec-abre-su-primera-estacion-de-carga-100-electrica-en-chile/FAZJZAXUFZFWDJXPKDYQ3MO2GE/>.
- [12] Arnechino, F. Tesla presenta Superchargers y anuncia puntos de carga rápida eléctrica en Chile: Parque Arauco, 2024. <https://redgol.cl/tendencias/tesla-presenta-superchargers-y-anuncia-puntos-de-carga-rapida-electrica>
- [13] N Gerlach. Infraestructura en Chile: Descubre cuántos cargadores hay para autos eléctricos, 2025. <https://www.rutamotor.com/>

infraestructura-en-chile-descubre-cuantos-cargadores-hay-para-autos
Recuperado el 26 de enero, 2026.

- [14] Ministerio de Energía, Gobierno de Chile. Ecocarga - Plataforma de Electromovilidad, n.d. <https://energia.gob.cl/electromovilidad/ecocarga> Recuperado el 26 de enero de 2026.
- [15] Voltbras. Gestión de carga de vehículos eléctricos - Inicio, n.d. <https://voltbras.com/inicio/> Recuperado el 26 de enero de 2026.
- [16] Md. U. Hashmi, M. Meraj Alam, O. L. V. Ramarozatovo, and M. Shadab Alam. Wecharge: democratizing ev charging infrastructure. In *CIREC Porto Workshop 2022: E-mobility and power distribution systems*, volume 2022, pages 1034–1038, 2022.
- [17] Yuntao Wang, Zhou Su, Jiliang Li, Ning Zhang, Kuan Zhang, Kim-Kwang Raymond Choo, and Yiliang Liu. Blockchain-based secure and cooperative private charging pile sharing services for vehicular networks. *IEEE Transactions on Vehicular Technology*, 71(2):1857–1874, 2022.
- [18] A. F. M. Suaib Akhter, Tawsif Zaman Arnob, Ekra Binta Noor, Selman Hizal, and Al-Sakib Khan Pathan. An edge-supported blockchain-based secure authentication method and a cryptocurrency-based billing system for p2p charging of electric vehicles. *Entropy*, 24(11), 2022.
- [19] L. H. S. Lube, S. Lube, F. Ahrens, and V. Lohweg. Enhancing EV Charging Stations Through IoT Platforms and Service Applications: An Analysis of the E-Mobility App Landscape. In *2024 IEEE International Conference on Engineering, Technology, and Innovation (ICE/ITMC)*, pages 1–10, 2024.
- [20] G. Sathish, S. S. Bharadwaj, S. K. K. Pavan, and V. S. K. Reddy. Smart Wireless EV Charging Station Using Raspberry Pi and Web-Based Monitoring. In *2025*

International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI), 2025.

- [21] E. Lobato, L. Prazeres, I. Medeiros, F. Araújo, D. Rosário, E. Cerqueira, M. Tostes, U. Bezerra, W. Fonseca, and A. Antloga. A Monitoring System for Electric Vehicle Charging Stations: A Prototype in the Amazon. *Energies*, 16(1):152, 2023.
- [22] R. Somasundaram, S. S. G. P. Prasad, K. K. K. Kumar, and M. G. G. Prasanth. Smart Systems Employing IoT Devices for Monitoring and Control of Electric Vehicle Residential Charging. In *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, 2024.
- [23] S. Guillemain, R. Choulet, G. Guyot, and S. Hing. Electrical Vehicle Smart Charging Using the Open Charge Point Interface (OCPI) Protocol. *Energies*, 17(12):2873, 2024.
- [24] A. Prasetyo, B. Sudiarto, and I. Garniwa. A Design of Smart Charging Architecture for Battery Electric Vehicles. In *2023 IEEE PES Innovative Smart Grid Technologies Europe (ISGT EUROPE)*, 2023.
- [25] Idowu Adetona Ayoade and Omowunmi Mary Longe. A comprehensive review on smart electromobility charging infrastructure. *World Electric Vehicle Journal*, 15(7), 2024.
- [26] S. Iqbal, N. F. Alshammari, M. Shouran, and J. Massoud. Smart and Sustainable Wireless Electric Vehicle Charging Strategy with Renewable Energy and Internet of Things Integration. *Sustainability*, 16(6):2487, 2024.
- [27] Alberto Varone, Zeno Heilmann, Guido Porruvecchio, and Alessandro Romanino. Solar parking lot management: An iot platform for smart charging ev fleets, using real-time data and production forecasts. *Renewable and Sustainable Energy Reviews*, 189:113845, 2024.

- [28] T. Fleck, S. Gohlke, and Z. Nochta. A System for the Efficient Charging of EV Fleets. *World Electric Vehicle Journal*, 14(12):335, 2023.
- [29] M. Dorokhova, J. Vianin, J.-M. Alder, C. Ballif, N. Wyrsh, and D. Wannier. A Blockchain-Supported Framework for Charging Management of Electric Vehicles. *Energies*, 14(21):7144, 2021.
- [30] Fazliddin Makhmudov, Dusmurod Kilichev, Ulugbek Giyosov, and Farkhod Akhmedov. Online machine learning for intrusion detection in electric vehicle charging systems. *Mathematics*, 13(5), 2025.
- [31] H. Tanyıldız, C. Şahin, Ö. Dinler, H. Migdady, K. Saleem, A. Smerat, A. Gandomi, and L. Abualigah. Detection of cyber attacks in electric vehicle charging systems using a remaining useful life generative adversarial network. *Scientific Reports*, (10092), 2025.
- [32] J. Ren, F. Xia, X. Chen, J. Liu, M. Hou, A. Shehzad, N. Sultanova, and X. Kong. Matching algorithms: Fundamentals, applications and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(3):332–350, 2021.
- [33] Harvard Dataverse. Dataset: Harvard Dataverse - persistentId=doi:10.7910/DVN/NFPQLW, n.d. <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/NFPQLW> Recuperado el 16 de diciembre de 2025.