



Universidad de las Ciencias Informáticas

Facultad 4

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Título:

Sistema de Monitorización y Control para Cargadores Eléctricos en vehículos eléctricos

Autor: Carlos Alberto León Vilches

Tutor: Ing. Andy Suarez Oña

Cotutor: Ing. Javier Orlando Villalobos Hernández

Ing. Fredy Xavier Betancourt Aguila

La Habana, diciembre 2022

“Año 64 de la Revolución”



“

El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando:[...]"

Che Guevara

Declaro ser el autor de la presente tesis que tiene por título: Sistema de Monitorización y Control para Cargadores Eléctricos en vehículos eléctricos y cedo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste, firmo la presente a los 8 días del mes de diciembre del año 2022 .



Carlos Alberto León Vilches

Firma del Autor



Ing. Andy Suarez Oña

Firma del Tutor



Ing. Javier Orlando Villalobos Hernández

Firma del Cotutor



Ing. Fredy Xavier Betancourt Aguila

Firma del Cotutor

Datos del Contacto

Tutor: Ing. Andy Suarez Oña graduado en la Universidad de las Ciencias Informáticas (UCI) en el año 2019.

Correo electrónico: aona@uci.cu

Cotutor: Ing. Javier Orlando Villalobos Hernández graduado en la Universidad de Ciencias Informáticas (UCI) en el año 2020.

Correo electrónico: jovillalobos@uci.cu

Cotutor: Ing. Fredy Xavier Betancourt Aguila graduado en la Universidad de Ciencias Informáticas (UCI) en el año 2020.

Correo electrónico: fxbetancourt@xetid.cu

DEDICATORIA

A mi madre que siempre estuvo, está y estará conmigo en todo momento, a mi padre que a pesar de no estar físicamente lo llevo siempre en mi corazón, este título es gracias a él y a su carácter, a mi tía toti por ser mi segunda madre, a mi hermano por ser mi estressa a seguir y paradigma a pesar de la distancia y a mi mujer que me ha dado el placer de ser lo más importante de este mundo, padre.

AGRADECIMIENTOS

Hoy es el día, si, hoy es el día en que por fin cumplo una de mis mayores metas, no es la 1era pero tampoco la última de ellas. No sabía cómo empezar esto ni por donde hacerlo, pero pregunté y me dijeron:

-Papi, ahí tú pones a todo el mundo, a todo el que quieras agradecerle ese triunfo, a todo aquel que te haya apoyado en algún momento, al que le das chuchos, al que te da a ti, al de las fiestas que terminan como se debe y como no deberían, al de las tomaderas hasta las 3 o las 4 de la mañana hablando de chismes, negocios, relaciones, novias, novios, disgustos, fracasos, cosas que solo se cuentan a la luz de la luna, ni que fuéramos hombres y mujeres lobos, a los que llegaron y por algunos motivos ya no están, a los que se quedaron a cumplir estos sueños junto a ti y a los que aspiran también a lograrlos, en fin. ¡A todo aquel que consideras tu FAMILIA!

En 1er lugar quiero agradecerles este título a mis padres que, sin ellos, estaría en Cienfuegos vendiendo cebolla en el mejor de los casos y en el peor ni quiero saberlo, a mi hermano, a mi mujer que me trae esa beba hermosa del mundo en su vientre, que seguro es como dicen por ahí (Ahora si las vas a pagar todas), mis tías Toti y Magaly, mi prima Neisi y mis padrinos. A esa familia de sangre y de sentimientos que no es mucha, pero es lo que tengo y de la que estoy orgulloso en todo momento.

-bueno ya, hasta aquí mis agradecimientos.

-y ya, más nadie, donde me pones a mí en eso.

-Vale, vale....

En 1.1 lugar, agradecer al claustro de profesores que se encuentra aquí presente que muchos, me dieron clases y otros los conozco desde que estaban estudiando, pero igual los valoro como magníficos profesionales y personas. Y para que no se

sientan fuera de esta familia, a todos ustedes, a todos los que me acompañan hoy, a mi piquetazo, que como dijo Dairon sin trabarse, (no la ponen fácil nunca). Mi Chama, que parece más mi papá que otra cosa, ¡Ay viejo ya no me regañes tanto ño! Te doy un consejo, no cantes dedícate a ser teacher de Ingles. Al Rolo que de verdad si no te llegas a graduar, de que te doy mi titulo te lo doy porque ya no cabía en ese cuarto. Al Pichi, Claudio y Yari espero que se acaben de poner para las cosas y se centren en lograr esto y algún día poder verlos aquí, donde estoy yo, aunque sea por videollamada. Paciente coño viejo espabílate y sale de los triángulos amorosos esos en que te metes, que no dan nada bueno. Al Fifi que espero que las mejores decisiones las tome con la cabeza fría y libre de lo que tú sabes. Damaris me dijiste que te pusiera de 1era pero coño vieja ya me basta con que voy a seguir aguantándote lo que me queda de vida por ser junto a Roy los padrinos de mi nena, con eso es más que suficiente. Tranca, gracias por estar presente y enseñarme a ser cuadrado a veces, como tú, pero te digo algo, a veces no se puede ser así, escuchar también es de sabios y las mujeres un poco claritas también son lindas, tampoco hay que exagerar.

Samira casi casi te hago mi novia y ya Dionis estaba más loco que porque llegara este día y le acabe de regalar la nutela que le debo. También agradecerle a mis prietos Julio y Clarke los mejores bailarines de toda la UCI, les deseo que nunca, pero nunca les falte ni el desodorante, ni el perfume. A mis cotutores Fredy y El Popo que muchas veces tuve que dejarlos sin dormir cayéndoles atrás, Andy y Adolfo (no me infles..., si..., y..., que más..., lee tu código papa..., estas embarcadoo..., jajajajaja...) que frustrantes son esas frases tuyas ñoooo, pero la experiencia obtenida fue genial en todos los sentidos. Guyana, Gretel, Jaidy, Carlos el escribano, Roberley, Cristian, Jesús el tanque programando, Jorgito, Arripapae y Yailé con su trío.

Félix, hoy cuando me pique el pelo de que te doy un poquito te doy, eso está más que claro. Frank vamos a ver si nos ponemos de acuerdo de los horarios que vas a estar con Yasiris porque esa Vicedecana es el alma de toda la facultad 4, Yordanki gracias por tus consejos como padre y como hombre. Agradecer a Yaili la 2D, Naila, Miguel Antúan, Alejandro, Wilber, Nelson, Nadieska, Aron, Adrián, Lesli, Joseyan, Póker, Gurri, Orlando, Javier, Quevin, Naomi y Ceilán mi gemela que pueden no estar aquí pero si formaron parte de grupo en estos años de estudio. A los dos mejores presidentes de la FEU que ha tenido la Fac4 Lester y Yadelis, en conclusión, a todos los que han llegado a mi vida a lo largo de estos años y han gravado su nombre en mí. ¡Muchas Gracias a todos!

Resumen:

Los vehículos eléctricos e híbridos enchufables se están convirtiendo día a día en una realidad cada vez más extendida. Así, la proliferación de dichos vehículos en todo el mundo va en aumento año tras año y cada vez más rápidamente, ya que se han reconocido como una tendencia irreversible. En Cuba la electrificación de este sector es una alternativa estratégica que contribuyó con la seguridad e independencia energética del país, al reducir su dependencia de los derivados del petróleo. En la presente investigación se trató de buscar una manera para darle solución a dicho problema, desarrollando el diseño *web* de un Sistema de Monitorización y Control de Cargadores Eléctricos para el uso de vehículos eléctricos e híbridos, así como la forma de desarrollar los servicios de electromovilidad para el ecosistema de carga público en nuestro país. El desarrollo del sistema *web* estuvo guiado por la metodología de desarrollo de *software* AUP en su versión (UCI) y para su implementación se utilizaron diversas tecnologías como el lenguaje de programación Java, el sistema gestor de base de datos PostgreSQL y como entorno de desarrollo IntelliJ IDEA. Se realizaron los flujos de modelado, implementación y prueba, indicados por la metodología escogida.

PALABRAS CLAVE: *eléctrico; híbrido; monitorización; vehículos; web.*

Abstract

Electric and plug-in hybrid vehicles are becoming an increasingly widespread reality day by day. Thus, the proliferation of such vehicles around the world is increasing year after year and more and more rapidly, since they have been recognized as an irreversible trend. In Cuba, the electrification of this sector is a strategic alternative that contributed to the security and energy independence of the country, by reducing its dependence on petroleum derivatives. In the present investigation, an attempt was made to find a way to solve this problem, developing the web design of a Monitoring and Control System for Electric Chargers for the use of electric and hybrid vehicles, as well as the way to develop electromobility services. for the public charging ecosystem in our country. The development of the web system was guided by the AUP software development methodology in its version (UCI) and various technologies were used for its implementation, such as the Java programming language, the PostgreSQL database management system and IntelliJ development environment. IDEA. The modeling, implementation and test flows were carried out, indicated by the chosen methodology.

Keywords: *electrical; hybrid; monitoring; vehicles; web.*

Índice de contenido

Introducción.....	16
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	21
1.1 Conceptos asociados a la investigación	21
1.2 Protocolo OCPP.....	25
1.3 Estudio de sistemas homólogos.....	26
1.3.1 Conclusión sobre el estudio de sistemas homólogos	30
1.4 Metodologías, herramientas y tecnologías.....	30
1.4.1 Metodología de desarrollo de <i>software</i>	30
1.4.2 Lenguaje de programación	31
1.4.3 Sistema de gestión de base de datos	32
1.4.4 Entorno de desarrollo.....	32
1.4.5 Herramienta para el modelado de procesos	33
Conclusiones del capítulo	34
CAPITULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA de SOLUCIÓN	35
2.1 Descripción de la propuesta de solución.....	35
2.2 Especificación de requisitos de <i>software</i>	35
2.2.1 Requisitos funcionales	36
2.2.2 Requisitos no funcionales	39
Requisitos no funcionales de usabilidad	39
2.3 Descripción de las historias de usuario	40
2.4 Patrón arquitectónico: Cliente-Servidor.....	41
2.5 Patrones GRASP	42
2.6 Patrones GOF	43

2.7 Diagrama de clases	44
Conclusiones del capítulo	44
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA	46
3.1 Estándar de codificación	46
3.2 Diagramas.....	48
3.3 Pruebas.....	50
3.3.1 Niveles de pruebas	51
3.3.2 Métodos de pruebas	52
3.4 Pruebas funcionales:.....	52
3.4.1 Método de caja negra mediante la técnica partición de equivalencia. ...	53
3.4.2 Pruebas unitarias	54
3.5 Resultado de las pruebas	56
Conclusiones del capítulo	58
CONCLUSIONES GENERALES.....	59
RECOMENDACIONES	60
REFERENCIAS BIBLIOGRÁFICAS	61
ANEXOS	69
Anexo 1: Historias de usuario	69
Anexo 2: Pruebas de caja negra mediante la técnica de partición de equivalencia.	79

Índice de tablas

Tabla 1: Comparación de sistemas homólogos.....	28
Tabla 2: Requisitos funcionales.....	36
Tabla 3: Historia de usuario " Registrar un punto de carga".....	40
Tabla 4: Historia de usuario "Editar un punto de carga".....	40
Tabla 5: Realización de la prueba de caja negra al requisito funcional "Modificar una reservación".....	53
Tabla 6: No conformidades.	56
Tabla 7: Historia de usuario" Eliminar un punto de carga".	69
Tabla 8: Historia de usuario" Listar puntos de cargas".....	69
Tabla 9: Historia de usuario" Registrar un conector del punto de carga".	70
Tabla 10: Historia de usuario" Editar un conector del punto de carga".....	70
Tabla 11: Historia de usuario" Eliminar un conector del punto de carga".	71
Tabla 12: Historia de usuario" Listar un conector del punto de carga".	71
Tabla 13: Historia de usuario" Registrar una sesión de carga".....	72
Tabla 14: Historia de usuario" Editar la sesión de carga".....	72
Tabla 15: Historia de usuario" Eliminar una sesión de carga".	73
Tabla 16: Historia de usuario" Listar una sesión de carga".	73
Tabla 17: Historia de usuario" Registrar una reservación de un conector en el punto de carga".	74
Tabla 18: Historia de usuario" Editar una reservación de un conector en el punto de carga".	74
Tabla 19: Historia de usuario" Eliminar una reservación de un conector en el punto de carga".	75
Tabla 20: Historia de usuario" Listar una reservación de un conector en el punto de carga".	75
Tabla 21: Historia de usuario" Procesar el metrado procedente de un punto de carga en el proceso de carga".....	76

Tabla 22: Historia de usuario” Iniciar una sesión de carga de forma remota”.	77
Tabla 23: Historia de usuario” Detener una sesión de carga de forma remota”. ...	77
Tabla 24: Historia de usuario” Mostrar un listado de los usuarios autorizados a cargar en el punto de carga”.	78
Tabla 25: Realización de la prueba de caja negra al requisito funcional “Eliminar un conector del punto de carga”.	79

Índice de figuras

Figura 1: Diagrama de clases.....	44
Figura 2: Ejemplo de código.....	46
Figura 3: Declaración de verbos.....	47
Figura 4: Declaración de variable.....	47
Figura 5: Diagrama de paquetes.....	49
Figura 6: Diagrama de despliegue.	50
Figura 9: Iteraciones de pruebas.....	57

INTRODUCCIÓN

Hoy en el mundo se está generando un cambio a favor del cuidado del medio ambiente en alerta de los irreversibles efectos del cambio climático, esto ha iniciado una búsqueda de mejoras en las tecnologías más sustentables a nivel global en las industrias con nuevos procesos y productos amigables con el medio ambiente. Así lo han entendido los países desarrollados, que hoy han tomado como alternativa en la industria automotriz la electromovilidad por su amigable desempeño con el medio ambiente (Cerón & Muñoz, 2017).

Los vehículos eléctricos e híbridos enchufables se están convirtiendo día a día en una realidad cada vez más extendida. Así, la proliferación de dichos vehículos en todo el mundo va en aumento año tras año y cada vez más rápidamente, ya que se reconoce como una tendencia irreversible. Las razones del avance de esta tecnología son técnicas, económicas y ecológicas. Varias marcas de automóviles han anunciado que todos o gran parte de sus modelos, dispondrán de motorización eléctrica en los próximos años. Del mismo modo, aunque un poco más lejos en el tiempo, algunos países se plantean prohibiciones de circulación o de venta de vehículos diésel o gasolina (ORBIS, 2020).

Francia ha anunciado que prohibirá la venta de autos de combustión en 2040, Dinamarca, Irlanda, Alemania y Holanda quieren lograrlo en 2030 y Noruega en 2025. El Reino Unido ya anunció que prohibirá la venta de vehículos diésel y a gasolina en 2040. El desarrollo de la electromovilidad en la industria automotriz a nivel global ha impulsado que el mercado de la venta de vehículos particulares en Chile tenga un nuevo escenario desde la llegada año 2012 (Cerón & Muñoz, 2017).

En los países desarrollados se ha iniciado el proceso de implementación en la infraestructura de las ciudades como primer paso, ubicando puntos de cargas en espacios públicos y privados. Facilitando a los usuarios disponibilidad de puntos de

cargas en varios lugares para tener una permanente autonomía en su traslado (Cerón & Muñoz, 2017).

En el 2017, el número de vehículos eléctricos a nivel mundial ha sobrepasado los 4 millones de unidades y se espera que crezca de forma paulatina hasta los 228 millones en 2030 (Bunsen et al., 2018). Los vehículos eléctricos, ya sean híbridos o puramente eléctricos, están provistos de una batería que proporciona la energía necesaria para su funcionamiento y que ha de ser cargada desde una fuente externa. La capacidad de la batería está directamente relacionada con la autonomía del vehículo. Para vehículos híbridos su valor oscila entre 5 y 15 kWh, mientras que para vehículos puramente eléctricos se sitúa en valores sensiblemente superiores entre 15 y 85 kWh (García-Villalobos et al., 2014). Por estos motivos, el despliegue del vehículo eléctrico afectará de forma significativa a la planificación y operación del sistema eléctrico, debido a la necesidad de establecer puntos de carga que proporcionen la energía necesaria en el tiempo requerido por el usuario final. (Jose Maria Masa Ortega, 2020)

En Cuba, el sector del transporte automotor consume anualmente 992 mil toneladas de combustible: 74 % diésel y el 26 % gasolina. La electrificación de este sector es una alternativa estratégica que contribuye con la seguridad e independencia energética del país, al reducir su dependencia de los derivados del petróleo. La adquisición de vehículos eléctricos en la Empresa Eléctrica de La Habana, Empresa de Telecomunicaciones de Cuba (ETECSA), Ómnibus Metropolitanos y Aguas de La Habana, con excelentes indicadores de ahorro de combustible, refiere que la inserción gradual de vehículos eléctricos en el país es un hecho.

Sin embargo, el nivel de explotación de los vehículos eléctricos en Cuba está limitado a la mitad de la autonomía diseñada por el fabricante, pues los únicos puntos de carga están situados en las empresas poseedoras de estas flotas. Así, debe reservarse carga suficiente para poder regresar, lo que limita la capacidad de gestión con estos vehículos para viajes largos. Además, el mecanismo de carga

utiliza un comportamiento por defecto *plug and play* local, definido por el fabricante, que no verifica si el cliente está autorizado a utilizar el cargador, ni limita la cantidad de energía a servir (Analizan uso de vehículos eléctricos, 2022).

Estas deficiencias atentan contra el propósito de establecer a lo largo y ancho del país, incluido autopistas, un número considerable de puntos de carga supervisados y controlados remotamente bajo el concepto de electrolineras. Se ha identificado la ausencia en Cuba de un actor reconocido dentro de los ecosistemas de carga público a nivel internacional: los proveedores de servicios de electromovilidad. Por las razones antes expuestas se le dio la tarea al centro de Tecnologías Interactivas (VERTEX) de la Universidad de Ciencias Informáticas (UCI) de realizar un sistema *web* el cual tendrá como tarea fundamental velar por el monitoreo y control de dichos puntos de cargas.

Dada la situación anteriormente expuesta se plantea el siguiente:

Problema investigativo: ¿Cómo contribuir al monitoreo y control de los servicios que provee los cargadores eléctricos en los vehículos eléctricos e híbridos?

Para dar solución al problema anterior se propone como **objetivo general:** desarrollar un sistema de monitorización y control de los servicios que proveen los cargadores eléctricos de vehículos eléctricos e híbridos en Cuba.

Se definió como **objeto de estudio:** los sistemas de monitoreo y control de los servicios.

Enmarcado en el **campo de acción** de: los sistemas de monitoreo y control de los servicios que provee los cargadores eléctricos de los vehículos eléctricos e híbridos.

Objetivos de la investigación que se deben cumplir:

- Elaborar el marco teórico conceptual relacionado con los aspectos teóricos que sustentan la investigación.
- Realizar el análisis y diseño de la propuesta de solución.

- Desarrollar la propuesta de solución como soporte al modelo.
- Realizar pruebas funcionales a la implementación.
- Validar la implementación a través de los métodos definidos en la investigación.

Para facilitar el desarrollo de la investigación se utilizaron diferentes métodos científicos:

1. **Histórico – lógico:** este método se utilizó para la realización del estudio de la historia del arte para investigar sistemas *web* que presenten soluciones similares.
2. **Análisis – síntesis:** para el estudio de las fuentes bibliográficas existentes referentes al tema, identificando los elementos más importantes y necesarios para dar solución al problema planteado.
3. **Modelación:** se utiliza con el objetivo de reproducir la interacción de los objetivos en la vida real mediante la creación de modelos.

Métodos empíricos:

1. **Estudio documental:** en la consulta de la literatura especializada en las temáticas a fines de la investigación.
2. **Observación:** sirvió para valorar las tecnologías y productos existentes, además de comparar con sistemas homólogos y así poder definir la propuesta, dando a conocer de manera minuciosa lo que se desea, lo que hace falta realizar y como se realizará.

Estructura de la Tesis

El trabajo se desarrolla mediante una estructura que se compone de tres capítulos, a continuación, se describe cada uno de ellos:

Capítulo1. Fundamentación Teórica, en este capítulo se centrará en la enunciación del marco teórico, así como la situación real del objeto de estudio. Teniendo en cuenta las características principales de la relación del sistema con el protocolo en uso, además de las herramientas a usar para darle solución.

Capítulo2. Análisis y Diseño, en este capítulo se explica la propuesta de resultado y se detallan las bases de la investigación que la sostiene. Haciendo enfoque en la propuesta de solución, compuesto por los requisitos funcionales y no funcionales.

Capítulo3. Implementación y pruebas del sistema, en este capítulo se describe la implementación y posterior validación realizada al producto obtenido como solución. Se describen las pruebas realizadas al sistema para así poder validar si funciona de manera correcta y si cumple con las especificaciones planteadas.

Al concluir estos tres capítulos, se da el resultado obtenido y las diferentes recomendaciones obtenidas basándose en los resultados de dicha investigación. También se da a conocer la bibliografía de la cual se pudieron ampliar los conocimientos para dicha investigación y para concluir mostramos algunos anexos que apuntalan los resultados mostrados de dicha investigación.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se abordan una serie de conceptos asociados de vital importancia para el desarrollo de la investigación. Se centrará en la enunciación del marco teórico, así como la situación real del objeto de estudio. Teniendo en cuenta las características principales de la relación del sistema con el protocolo en uso, además de las herramientas a usar para darle solución.

Se quiere desarrollar un sistema de monitorización que permita controlar, mostrar y modificar los datos de determinados puntos de cargas mediante un sistema de gestión *web*. Este va a estar vinculado a un sistema central utilizando la comunicación mediante el protocolo (OCPP) *Open Charge Point Protocol* en su versión 1.6. Para darle solución a esta problemática se ha decidido desarrollar el diseño *web* de un Sistema de Monitorización y Control de Cargadores Eléctricos para el uso de vehículos eléctricos e híbridos. Así como la forma de desarrollar los servicios de electromovilidad para el ecosistema de carga público en nuestro país. Es necesario darle solución a la problemática con un sistema *web* porque existen una serie de usuarios que van a estar registrados y estos necesitan tener un control de acceso global en todo momento. Sin un sistema de gestión esto no fuera posible ya que este va a estar destinado a proporcionar ayuda a los diferentes niveles de decisión en una empresa, para coordinar las acciones necesarias en función de conseguir los objetivos propuestos. Ayuda a documentar los procesos para optimizar, mantener la efectividad en las operaciones, lograr la satisfacción del cliente y crear un entorno de trabajo en mejora continua. Suponen importantes ahorros de dinero para las empresas y permite un acceso rápido de la búsqueda de la información la cual va a estar siempre actualizada.

1.1 Conceptos asociados a la investigación

Para el entendimiento de la presente investigación se es necesario conocer algunos conceptos fundamentales como:

Aplicaciones web

Son programas que funcionan en internet, los datos o los archivos con que trabajan son procesados y almacenados dentro de la *web*. Estas aplicaciones por lo general no necesitan ser instaladas en el computador, su concepto está relacionado con el almacenamiento en la nube. Toda la información se guarda en grandes servidores, dejando una copia temporal dentro de tu equipo (libre, 2022).

Sistema central

El sistema central o sistema de control de carga es la pieza central de una infraestructura de recarga inteligente y sostenible para la carga de vehículos eléctricos. Este posee varias características e interfaces en un *hardware* compacto y modular. Su *software* escalable optimiza el funcionamiento de los postes de carga aumentando su disponibilidad y permitiendo una facturación precisa de los procesos de carga. Gracias a su plataforma de Linux abierta están preparados para aplicaciones que utilicen el internet de las cosas, *Smart Services* e integración de sectores energéticos (Contact, 2022).

Sistema de monitorización

Un sistema de monitorización es el encargado de hacer un seguimiento de estado tanto en la infraestructura como de los subsistemas presentes. Su objetivo es asegurar que dicho sistema sea fiable y estable capaz de proporcionar los servicios para los que ha sido diseñado (Castaños, 2021).

Punto de carga

Los puntos de cargas permiten recargar de forma segura cualquier tipo de coches eléctricos o híbridos enchufables. Son cargadores que se instalan mediante anclajes. Estos se pueden diferenciar por varios tipos o características singulares como:

Punto de recarga portátil o portables

Estos son los más usados porque brinda la facilidad de recargar en conectores domésticos convencionales y también en los industriales (lugEnergy, 2022).

Puntos de recargas de pared o también llamados *Wallbox*

Recargar de forma segura cualquier tipo de coche eléctrico o híbrido enchufable que se instalan mediante anclajes en la pared del garaje ya sea privado o unifamiliar o garaje comunitario, tienen como ventaja una instalación sencilla, facilidad de uso y su coste económico (lugEnergy, 2022).

Punto de recarga con control dinámico de potencia

Este es el último avance en la recarga de coche eléctricos se usa una tecnología que balancea la carga entre el vehículo eléctrico y otros consumos del hogar, para que no sobrepase nunca la potencia contratada y así ayuda a ahorrar en el consumo de energía (lugEnergy, 2022).

Punto de recarga en poste

Permite recargar el vehículo de tal manera que en un periodo de media hora alcance el 80% de la carga total de la batería, estos tipos de recargas pertenecen a administraciones públicas o empresas. Son muy factibles a la hora de realizar largos trayectos o viajes que superen la autonomía de nuestro coche eléctrico (lugEnergy, 2022).

Electromovilidad y *Plug and play*

Electromovilidad o movilidad eléctrica, hace referencia a la utilización de vehículos eléctricos, siendo entendido como aquellos que emplean combustibles y/o energía alternativa impulsado por un motor eléctrico. En este amplio concepto se incluyen una serie de tipologías: aquellos con batería eléctrica o 100 por ciento eléctricos, los híbridos enchufables y con celdas de combustible que incluyen vehículos de dos y tres ruedas, automóviles, camionetas comerciales ligeras, autobuses y camiones. Podríamos inferir que engloba dos tipos de tecnología: automóviles con batería de

ion-litio y automóviles con celdas de combustible, donde uno genera la electricidad con hidrógeno. Además, se está avanzando en el desarrollo de baterías de grafeno.

La movilidad sostenible no sería posible sin la implicación del ámbito gubernamental. Por consiguiente, se necesitan legislaciones que la fomenten a todos los niveles la utilización de medios alternativos de transporte, como lo es el vehículo eléctrico. Aunque no estrictamente la electromovilidad se refiere a los automóviles, existen hoy en día distintas tendencias punta de lanza en la implementación de medios de transporte ecológicos.

Uno de ellos es justamente el *carsharing*, que permite que a través de un mismo automóvil se desplacen un determinado número de personas que van a un mismo destino, como lo puede ser el trabajo, y que es precisamente muy empleado en esos ambientes laborales para disminuir emisiones contaminantes, como para reducir incluso los niveles de tránsito en las ciudades. Otras tendencias son el *motosharing* y el *bikesharing*, que permite que las personas accedan a medios de transporte eléctricos en las ciudades de mayor tránsito, como la Ciudad de México, que cuenta con varias marcas que proporcionan estos servicios (Zacua, 2021).

Al momento, la movilidad eléctrica todavía lleva consigo mitos causados por el miedo al cambio o desconocimiento. Por tal motivo, los fabricantes de infraestructura de carga y empresas que deciden instalar cargadores para vehículos eléctricos hoy optan por el *plug and play*. Este permite al usuario del auto eléctrico tener la posibilidad de acercarse a un punto de carga, conectar y cargar sin la necesidad de una autenticación. Esto es para hacer la tecnología más accesible y mediante una tarjeta de Identificación por Radiofrecuencia (RFID), una aplicación o una llave no sea un impedimento en la transición.

En países como Argentina los únicos autorizados a vender energía para la recarga de vehículos eléctricos e híbridos enchufables son las distribuidoras, por ende, un paso previo a la carga podría ser una barrera casi sin sentido. Sin embargo, en

lugares más avanzados en la penetración de la movilidad eléctrica se encuentra un sistema muy similar que es el *plug and charge*. Con esa solución el conductor llega, conecta, carga y al finalizar automáticamente se le debita de la billetera virtual del servicio contratado el monto correspondiente (*Plug and play*, 2021).

1.2 Protocolo OCPP

El protocolo OCPP, cuyas siglas significan *Open Charge Point Protocol*, es un protocolo libre / abierto para puntos de recarga o dicho de otra forma, protocolo *open-source* para gestionar, a través de un *web manager* o sistema de gestión en la nube (CPMS, *Charge Point Management System*), estaciones de recarga de vehículos eléctricos, conocidos con el término *electrolineras*, cuya función es idéntica al de un surtidor convencional de combustible con la diferencia de que suministra energía eléctrica a vehículos que dispongan de sistema de baterías, ya sean híbridos (motor de combustión + motor eléctrico) o puramente eléctricos (Vazquez, 2022).

Es un protocolo de comunicación que facilita el intercambio de información entre un cargador y su sistema de gestión *backend* y fomenta su interoperabilidad. El Sistema de Gestión de Estaciones de Carga (CSMS) es el sistema *backend* que permite a los operadores de los puntos de recarga gestionar y supervisar las operaciones en la instalación de recarga, la generación de informes y la gestión remota de los cargadores. La comunicación entre la estación de carga y el CSMS es la clave para ejecutar las operaciones de cargas y es donde se pone de manifiesto el protocolo OCPP el cual se utiliza para canalizar la comunicación entre la estación de carga y CSMS para poder realizar operaciones como la autorización previa y permitir al usuario del vehículo el uso de este servicio, proporcionar los detalles de la sección de carga al sistema *backend* para calcular los cargos, permitir la funcionalidad de las aplicaciones de usuarios para ubicar las estaciones de cargas y hacer reservas. Los datos necesarios de la sección de carga se transfieren al sistema de gestión de carga para así poder ver el estado en tiempo real de todos

los equipos de cargas que estén disponibles, actualmente cargándose, o cuales no están funcionando. Este protocolo tiene como principales ventajas que puede conectar cualquier CSMS con cualquier estación de carga siempre y cuando ambos sean compatibles con la versión del OCPP. Brinda a los operadores de puntos de carga la flexibilidad para adquirir equipos de cargas de múltiples fabricantes que cumplan con el OCPP. Este protocolo es independiente de la técnica de carga. Se pueden agregar nuevas capacidades y mejoras a los cargadores con relativa facilidad a través de la apertura que este ofrece. Permite la transferencia de energía bidireccional para facilitar la comunicación de la red y la respuesta automatizada a la demanda (ALLIANCE, 2021).

1.3 Estudio de sistemas homólogos

Existen varios sistemas y aplicaciones que permiten monitorizar y controlar cargadores eléctricos. A continuación, se detallan una serie de ellas que han podido aportar ideas ya sea por su cercanía con el tema y funcionalidades de la aplicación o por el diseño innovador:

- *Phoenix Contact E-Mobility*: Ofrece una amplia gama de tecnología de carga para el creciente mercado de la electromovilidad, tanto para la interfaz de carga en el lado del vehículo como para la infraestructura de recarga. Crean soluciones globales en red para una movilidad sostenible y respetuosa con el medio ambiente. Es especialmente conocida por sus cables de carga CCS (*Combined Charging System*), así como por el desarrollo de la tecnología HPC (*High Power Charging*) refrigerada por líquido, que permite una carga ultrarrápida en pocos minutos (Critchley, Liam, 2021).
- *Uhora Solar Solvers*: Compañía de servicios energéticos solares creada para revolucionar el autoconsumo solar garantizando la mayor calidad en sus servicios, con el fin de lograr que se produzca el cambio hacia la energía verde a través de la instalación de paneles fotovoltaicos. Entre los servicios

que prestan se encuentran, paneles solares, baterías solares y puntos de carga.¹

- *Plug4car*: Es una solución completa para la recarga de vehículos eléctricos. Esta solución está compuesta por *hardware* (puntos físicos de carga), *software* (plataforma *web* y aplicación móvil) y todos los servicios complementarios necesario (Instalación, *Service Desk*, monitorización, gestión remota y mantenimiento) (Dimática Software, 2022).
- CIRCONTROL: Empresa con tecnología propia donde la innovación es un reto constante. Fundada para dar soluciones en el mundo de la movilidad y soluciones de recarga para vehículos eléctricos. Ya desde sus inicios viene dedicando una parte importante de sus recursos en I+D, fabricando productos de alta calidad, esto le ha permitido ser líder no solo en soluciones estándar sino también en soluciones específicas demandadas por el mercado. Su actividad principal se centra en el desarrollo de: *Mobility* (Soluciones de eficiencia para la movilidad en aparcamientos) y *eMobility* (Soluciones de recarga inteligente de vehículos eléctricos). Todos los productos cumplen con las normas de calidad más exigentes y son sometidos a rigurosos controles en el proceso de fabricación. Dispone de laboratorio propio de ensayos, calibración y verificación metrológica.²
- Sistema de monitorización telemetría de una estación de recarga de vehículos eléctricos usando estándar OCPP v1.6: sistema de monitorización remota basado en el estándar OCPP v1.6 donde se visualizan los parámetros principales como: tensión, intensidad, consumo kWh, energía activa y energía reactiva importada y exportada a un vehículo y potencia activa (Saraullo, 2022).

¹ <https://www.vidaeconomica.com/2020/10/ubora-solar-solvers-solarwatt-energia-solar-andalucia/>

² <https://circontrol.com/es/>

La tabla 1 muestra la comparación de los sistemas homólogos analizados.

TABLA 1: COMPARACIÓN DE SISTEMAS HOMÓLOGOS.

Sistemas	De Pago	Cualidad Laboral	Desarrollo	Enfoque
Sistema de Monitorización de Telemetría de una Estación de Recarga usando estándar OCPPv1.6	No	Fuerte	Posee un desarrollo abierto, impulsado por el mercado existente adaptándose al ámbito donde se va a desplegar.	Posee un enfoque pragmático, aprovechando el conocimiento y la experiencia de los expertos como aspecto fundamental.
<i>Phoenix Contact E-Mobility</i>	Si	Fuerte	Utiliza tecnología de carga potente e inteligente.	Posee un enfoque Sostenible y futurista, el cual va dirigido a las nuevas generaciones automovilísticas.
<i>Uhora Solar Solvers</i>	Si	Fuerte	Pretende optimizar los recursos ahorrando en la factura de la luz con responsabilidad y sin complicación. Brinda soluciones	Va dirigida a las personas que poseen un proyecto de vida claro. Posee un enfoque sostenible, futuristas y siempre va enfocada a mantener presente los conceptos de: <ul style="list-style-type: none"> • Honestidad

			energéticas innovadoras, limpias y accesibles.	<ul style="list-style-type: none"> • Innovación • Superación • Sostenibilidad • Confianza
<i>Plug4car</i>	Si	Potente (Se autodenomina)	Sus actualizaciones evolucionan de acuerdo con las necesidades del mercado.	Posee un enfoque personalizado y permite el acceso a este sistema desde cualquier navegador de internet.
CIRCONTROL	Si	Fuerte	Ofrece en detalles los aspectos que considera elementales para la carga de vehículos eléctricos en hogares.	Brinda un enfoque dirigido a mejorar las comodidades de sus clientes a la hora de realizar una carga segura y soluciones de parqueo eficientes. Posee un carácter comprometido con la sostenibilidad y liderazgo en el mercado del vehículo eléctrico y con profesionales de mucho talento, como elemento clave de la compañía.

1.3.1 Conclusión sobre el estudio de sistemas homólogos

Después de haber realizado un amplio estudio de los sistemas homólogos existentes en el mundo para realizar el monitoreo de cargadores eléctricos, queda demostrado que el uso del Sistema de Monitorización de Telemetría de una estación de recarga de vehículos eléctricos usando estándar OCPP v1.6, es el más apropiado para usar. Teniendo en cuenta que nos brinda fortaleza y seguridad a la hora de desarrollar el sistema. Posee un desarrollo abierto sin restricciones en el uso del estándar utilizado, el cual va impulsado por las necesidades del mercado, cumpliendo así los requisitos tecnológicos y comerciales existentes. Conserva un enfoque pragmático que aprovecha el conocimiento y la experiencia de los trabajadores en infraestructura de carga de vehículos eléctricos. Define el OCPP como estándar vital para su desarrollo ya que es un protocolo de código abierto y gratuito sin restricciones a un único proveedor.

1.4 Metodologías, herramientas y tecnologías

A continuación, se definen la metodología de desarrollo de *software*, las herramientas tecnológicas que se utilizarán para el desarrollo de la propuesta de solución.

1.4.1 Metodología de desarrollo de *software*

La metodología de desarrollo de *software* que se selecciona para el desarrollo de la propuesta de solución es la variación de Proceso Unificado Ágil (AUP, por sus siglas en inglés) para la UCI. Esta metodología define tres fases (Inicio, Ejecución, Cierre) y cuatro posibles escenarios para modelar el sistema a partir de los requisitos.

En la presente investigación se seleccionará el escenario 4, el cual permite a través de historias de usuarios un encapsulamiento de los requisitos del sistema. Este escenario se aplica a negocios bien definidos y que no sean muy extensos, además el cliente estará siempre acompañando al equipo de desarrollo para convertir los detalles de los requisitos y así poder implementarlos, validarlos y probarlos. Intenta

evitar los torcidos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados. Nos permite usar cualquier conjunto de herramienta que se desee, principalmente las que sean más adecuadas a la hora de realizar el trabajo que a menudo son herramientas simples o incluso de código abierto. Esta metodología también enfatiza que el *software* funcional es la primera medida del progreso (Benítez, 2017).

1.4.2 Lenguaje de programación

El lenguaje de programación Java fue desarrollado originalmente por James Gosling en 1983 y luego el 27 de enero del 2010 fue adquirida por la compañía Oracle. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son compiladas a *bytecode* (clase Java), que puede ejecutarse en cualquier máquina virtual Java sin importar la arquitectura de la computadora subyacente (Oracle, 2022).

Como *framework* para el desarrollo del *backend* se escogió *Java Spring Boot* el cual es una herramienta principal para el desarrollo *web*. Nace de la complejidad que se tenía que hacer a la hora de crear proyectos *Java Enterprise Edition*, *loa EJBs* y la forma de realizar el proyecto era muy engorroso. (Gonzales, 2021). Con *Spring Boot* dejamos de lado de tener que desplegar artefactos *Jar* o *War* de manera independiente en uno o muchos servidores *web* diferentes. Porque nos provee una serie de contenedores *web servlet* para que se despliegue nuestra aplicación automáticamente solo con un “*Run*”. Toda la arquitectura la integramos desde *Spring Boot* como foco principal, además podemos hacer uso de anotaciones tales como *Repository*, *Service*, *Component*, entre muchas otras que nos permiten desarrollar nuestra aplicación bajo la arquitectura que nos provee *Spring*, bien sea para interactuar con bases de datos, crear lógica de negocio o simplemente desarrollar un componente general para todas las capas de la aplicación (Garcia, 2021).

Como *framework* principal para el desarrollo del *Frontend* se escogió Angular el cual está escrito en *TypeScript*. Posee un desarrollo basado en componentes para crear una aplicación *web* escalable y ofrece una mejor estructura sintáctica al tiempo que hace que la base del código sea más fácil de entender y mantener. Angular posee una colección de biblioteca bien integradas, que cubren una amplia variedad de características que incluyen enrutamiento, administración de formularios y comunicación cliente-servidor. Permite desarrollar, compilar, probar y actualizar el código fuente de la aplicación desarrollada. Este *framework* aprovecha una plataforma que puede escalar desde un solo desarrollador hasta aplicaciones de nivel empresarial. *Google* se encarga de que su mantenimiento y su actualización sea lo más sencilla posible, para poder aprovechar los últimos desarrollos con un mínimo de esfuerzo. Consta con más de 1,7 millones de desarrolladores, actores de bibliotecas y creadores de contenidos. Angular posee una arquitectura modelo-vista-vista-modelo. Proporciona una guía clara sobre cómo se debe estructurar la aplicación y ofrece un flujo de datos bidireccional al tiempo que proporciona un dominio real (Gonçalves, 2021).

1.4.3 Sistema de gestión de base de datos

Como sistema de gestión de bases de datos se escogió *PostgreSQL* en su versión 11 el cual es un sistema de código abierto de administración de bases de datos. En este sistema, las consultas relacionales se basan en *Structured Query Language* (SQL), mientras que las no relacionales hacen uso de *JavaScript Object Notation* (JSON). Este posee tipos de datos avanzados y permite ejecutar optimizaciones de rendimiento avanzadas que son varias características que solo se pueden ver en sistemas de bases de datos comerciales (Borges, 2019).

1.4.4 Entorno de desarrollo

Para el desarrollo del *frontend* se seleccionó *Visual Studio Code* (VS Code) en su versión 1.73.1, el cual es un editor de código fuente desarrollado por *Microsoft*. Es un *software* libre y multiplataforma. Cuenta con un soporte para depuración de

código y dispone de un sinnúmero de extensiones. Estas extensiones dan la posibilidad al usuario de escribir y ejecutar códigos de cualquier tipo de lenguaje de programación. Es capaz de detectar errores de forma automática antes de ejecutar el código o la depuración. Algo importante y una ventaja con respecto a un IDE completo que incluye todos los componentes en un solo paquete, es que, con VS Code puedes instalar únicamente las herramientas de desarrollo requeridas, y personalizarlo de acuerdo con tus necesidades (Flores, 2022).

Como entorno de desarrollo para el *backend* se seleccionó *IntelliJ IDEA* en su versión 2.7.4 el cual es un IDE inteligente y sensible al contexto para trabajar con Java y otros lenguajes JVM como *Kotlin*, *Scala* y *Groovy* en todo tipo de aplicaciones. Además, *IntelliJ IDEA ultimate* puede ayudarte a desarrollar aplicaciones *web* de pila completa, gracias a sus potentes herramientas integradas, compatibilidad con *JavaScript* y tecnologías relacionadas, y compatibilidad avanzada con marcos de trabajo populares como *Spring* y *Spring Boot*. Este tiene como gran ventaja que le brinda al usuario una experiencia perfecta desde el primer momento. Proporciona un acceso rápido a todas las funcionalidades y herramientas integradas e importantes para su trabajo y como una amplia variedad de opciones de personalización para que sea compatible con su flujo de trabajo (Tatiana Grin, 2022).

1.4.5 Herramienta para el modelado de procesos

Para el modelado de procesos se utilizó el *Visual Paradigm* en su versión 8.0 esta permite ayudar a los equipos de desarrollo de *software* a capturar los requisitos correctos y transformarlos en diseños precisos. Ayuda a los desarrolladores a crear el *software* adecuado según los requisitos lo que genera un resultado con mayor calidad. Es una herramienta CASE (*Computer Aided Software Engineering*, Ingeniería de *Software* Asistida por Computadora) para el desarrollo que utilizan el lenguaje UML (Lenguaje Unificado de Modelado) ofreciendo confiabilidad y estabilidad en el proceso de desarrollo. Permite dibujar todos los tipos de diagramas

de clases, código inverso, generar códigos desde diagramas y la generación de documentación (Méndez, 2015).

Conclusiones del capítulo

Durante este capítulo se abordaron los elementos teóricos que sustentan la propuesta de solución del problema planteado, en este sentido se concluye:

- El análisis de los sistemas homólogos permitió identificar las principales funcionalidades que debe tener el sistema a desarrollar.
- El estudio de las tendencias en el desarrollo de este dominio de aplicaciones contribuyó a la selección de las herramientas, tecnologías y metodología a utilizar en la presente investigación.
- La selección de las herramientas y lenguaje de programación a utilizar aseguró el ambiente de desarrollo necesario para la implementación de la propuesta de solución; en el cual se definió como lenguaje de programación Java, como entorno de desarrollo para el *backend* se seleccionó *IntelliJ IDEA* en su v2.7.4, para el desarrollo del *frontend* *Visual Studio Code* en su versión 1.73.1, como herramienta para diseñar y modelar procesos el *Visual Paradigm* en su versión 8.0 y como gestor de base de datos *PostgreSQL* en su versión 11.

CAPITULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

El presente capítulo aborda los principales aspectos relacionados con las características de la propuesta de solución. Se encapsulan los principales requisitos funcionales y no funcionales con los que debe cumplir la solución propuesta, así como estilo arquitectónico y los patrones de diseño para lograr buenas prácticas en el diseño y posterior implementación del sistema.

2.1 Descripción de la propuesta de solución

Dada las necesidades planteadas en la situación problemática de la presente investigación la solución propuesta constituye un sistema para el monitoreo de cargadores eléctricos. El *software* de desarrollo sobre la base de una aplicación *web* brindándole a los usuarios los beneficios que estas aportan, el acceso al sistema por medio de un servidor *web*, así como las actualizaciones y el mantenimiento del *software* sin necesidad de distribuir e instalar.

Este sistema contribuye a gestionar, a través de un *web manager* o CSMS estaciones de recarga de vehículos eléctricos, conocidos con el término electrolineras, cuya función es idéntica al de un surtidor convencional de combustible con la diferencia de que suministra energía eléctrica a vehículos que dispongan de sistema de baterías, ya sean híbridos (motor de combustión + motor eléctrico) o puramente eléctricos (evblog, 2020).

2.2 Especificación de requisitos de *software*

Con el objetivo de que el sistema se desarrolle con las necesidades dictadas por el cliente a partir del uso de la tormenta de ideas y entrevista al cliente se obtuvieron los requisitos funcionales y no funcionales.

2.2.1 Requisitos funcionales

Los requisitos funcionales son el resultado del sistema y sus componentes cuando el usuario realiza una tarea o función descrita como un conjunto de entradas, comportamientos y salidas sobre los mismos. Fueron evaluados con el fin de establecer la prioridad y complejidad según el producto de trabajo de la evaluación de requisitos propuesto por la metodología AUP-UCI. Para determinar la prioridad se asumieron los criterios de urgencia he Importancia para el cliente.

En la siguiente tabla se muestra la descripción de los requisitos funcionales que se tuvieron en cuenta para la realizar el sistema:

TABLA 2: REQUISITOS FUNCIONALES.

Nombre	Descripción	Prioridad	Complejidad
RF1: Registrar un punto de carga.	El sistema debe permitir registrar un punto de carga.	Alta	Media
RF2: Editar un punto de carga.	El sistema debe permitir editar un punto de carga.	Alta	Alta
RF3: Eliminar un punto de carga.	El sistema debe permitir eliminar un punto de carga.	Alta	Media
RF4: Listar puntos de cargas.	El sistema debe permitir listar los puntos de cargas registrados en el sistema.	Alta	Alta

RF5: Registrar un conector del punto de carga.	El sistema debe permitir registrar un conector al punto de carga.	Alta	Media
RF6: Editar un conector del punto de carga.	El sistema debe permitir editar el conector del punto de carga.	Alta	Alta
RF7: Eliminar un conector del punto de carga.	El sistema debe permitir eliminar el conector del punto de carga.	Alta	Media
RF8: Listar un conector del punto de carga.	El sistema debe permitir listar los conectores de los puntos de carga.	Alta	Media
RF9: Registrar una sesión de carga.	El sistema debe permitir registrar una sesión de carga.	Alta	Alta
RF10: Editar una sesión de carga.	El sistema debe permitir editar la sesión de carga.	Alta	Alta
RF11: Eliminar una sesión de carga.	El sistema debe permitir eliminar una sesión de carga.	Alta	Alta
RF12: Listar una sesión de carga.	El sistema debe permitir listar una sesión de carga.	Alta	Alta
RF13: Registrar una reservación de un conector en el punto de carga.	El sistema debe permitir registrar una reservación de un conector en el punto de carga.	Alta	Alta

RF14: Editar una reservación de un conector en el punto de carga.	El sistema debe permitir editar una reservación de un conector en el punto de carga.	Alta	Alta
RF15: Eliminar una reservación de un conector en el punto de carga.	El sistema debe permitir eliminar una reservación de un conector en el punto de carga.	Alta	Alta
RF16: Listar una reservación de un conector en el punto de carga.	El sistema debe permitir listar una reservación de un conector en el punto de carga.	Alta	Alta
RF17: Procesar metrado.	El sistema debe permitir procesar el metrado procedente de un punto de carga en el proceso de carga.	Alto	Alto
RF18: Iniciar una sesión de carga de forma remota.	El sistema debe permitir que se inicie una sesión de carga de forma remota.	Alta	Alta
RF19: Detener una sesión de carga de forma remota.	El sistema debe permitir que se detenga una sesión de carga de forma remota.	Alta	Alta
RF20: Mostrar un listado de los usuarios autorizados a cargar en el punto de carga.	El sistema debe permitir que se muestre un listado de los usuarios autorizados a cargar en el punto de carga.	Alta	Alta

2.2.2 Requisitos no funcionales

Los requisitos no funcionales son atributos de calidad que especifican criterios, que pueden usarse para evaluar las operaciones de un sistema en cuanto a características del *software* y cualidades restricciones. Estos sirven como normas para realizar un correcto funcionamiento del sistema que se va a realizar.

Requisitos no funcionales de usabilidad

- RnF1: El sistema debe de ser comprensible para personas de 18 años en adelante.
- RnF2: El sistema debe mostrar toda la información necesaria para que el usuario pueda interactuar con este con facilidad.

Requisitos no funcionales de *hardware* y *software*

- RnF3: <Memoria RAM>. <El sistema debe ejecutarse con una memoria RAM mínima de 2GB >.
- RnF4: <Microprocesador>. <El sistema debe ejecutarse sobre un procesador *Intel Core i3 2370M* o superior >.
- RnF5: <Atributo de Calidad> <Debe garantizar el acceso a la información de manera confiable, cumpliendo los parámetros de uso y calidad>.
- RnF6:<Máquina virtual de *java JDK-11*>.
- RnF7: <Plataforma > <El sistema debe ejecutarse en *Linux* y *Windows*. >.

Requisitos no funcionales de seguridad

- RnF8: El usuario no puede realizar una sesión de carga si este no se encuentra registrado en el sistema.

Requisitos no funcionales de adecuación funcional

- RnF9: El sistema debe garantizar que cada botón tenga una funcionalidad.
- RnF10: El sistema debe garantizar que este tenga todas las funcionalidades desarrolladas y listas para su uso.

2.3 Descripción de las historias de usuario

Para la obtención de los requisitos de *software* tal como lo plantea la metodología AUP-UCI en su escenario 4, se generaron las historias de usuarios para cada uno de los requisitos funcionales adquiridos. A continuación, se muestran algunas historias de usuario, el resto se encuentra en el [Anexo 1](#).

TABLA 3: HISTORIA DE USUARIO " REGISTRAR UN PUNTO DE CARGA".

Historia de usuario	
Número: 1	Nombre: Registrar un punto de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir registrar un punto de carga para esto es necesario que se introduzca los siguientes atributos, el charge_point_id, central_system_path, charge_box_serial_number, charge_box_vendor, charge_point_serial_number, description, firmware_version, iccid, identity, imsi, last_heart_beat, meter_serial_number, meter_type, status_registration.	

TABLA 4: HISTORIA DE USUARIO "EDITAR UN PUNTO DE CARGA".

Historia de usuario	
Número:2	Nombre: Editar un punto de carga.
Usuario: Usuarios	
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración:1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir editar un punto de carga. Para poder realizar esta funcionalidad es necesario modificar alguno de los siguientes atributos, charge_point_id, identity, central_system_path, description, iccid, imsi, meter_type.	

2.4 Patrón arquitectónico: Cliente-Servidor.

Cliente-Servidor es uno de los estilos arquitectónicos comercializados más conocidos, el cual está compuesto por dos mecanismos, el distribuidor y el consumidor. El distribuidor es un servidor que brinda una serie de servicios o recursos los cuales son consumido por el cliente en este caso los puntos de cargas y los usuarios registrados en el servidor. Existe un servidor y múltiples clientes que se conectan al servidor para recuperar todos los recursos necesarios para funcionar. En este sentido, el cliente solo es una capa para representar los datos y se detonan acciones para modificar el estado del servidor, mientras que el servidor es el que hace todo el trabajo pesado, se encarga de monitorial todos los puntos de cargas conectados a él, y así mismo a todos los usuarios almacenados en este sistema.

Esta es una arquitectura en el que el servidor deberá exponer un mecanismo que permite a los clientes conectarse, que por lo general es TCP/IP, esta comunicación permitirá una comunicación continua y bidireccional, de tal forma que el cliente puede enviar y recibir datos del servidor y viceversa (Oscar Blancarte, 2020).

2.5 Patrones GRASP

En diseño orientado a objetos, GRASP son patrones generales de *software* para asignación de responsabilidades, es el acrónimo de "GRASP (*object-oriented design General Responsibility Assignment Software Patterns*)". Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de *software*. (Kord, 2011)

Principales patrones GRASP:

1. Experto

Es el responsable de asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Este patrón está presente en las clases *ChargePoint*, *ChargingSession*, *Connector* y *Reservation*.

2. Creador

Es el encargado de asignar la responsabilidad de la creación de una nueva instancia y explica que clase es la encargada de crear objetos. Este patrón está presente en las clases *ChargePointRepository*, *ChargingSessionRepository*, *ConnectorRepository* y *ReservationRepository*.

3. Alta Cohesión

Las clases están realizadas de forma tal que contengan las mínimas responsabilidades y colaboren con otras para llevar a cabo una tarea. Esto permitirá tener patrones más fáciles de mantener, entender y reutilizar. Este patrón está presente en las clases *ChargingSessionRepository*, *ChargePointRepository*, *Connector* y *Reservation*.

4. Bajo Acoplamiento

Es el estado ideal que siempre se intenta obtener para lograr una buena programación o un buen diseño. De tal forma que, en caso de producirse un cambio en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases. Este patrón está presente en las clases *Connector*, *Reservation*, *ChargePointRepository* y *ChargePoint*.

5. Controlador

Es un objeto responsable de la administración de los programas del sistema, que no pertenece a la interfaz del usuario, el controlador recibe la solicitud del servicio desde la capa guía y coordina su realización delegando a otros objetos (2011). Este patrón está presente en la clase *ChargePointController*, *ChargingSessionController*, *ConnectorController* y *ReservationController*.

2.6 Patrones GOF

Los patrones de diseño del grupo *Gang of Four* (GOF), se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. A continuación, se describirá el patrón GOF utilizado en la solución propuesta:

Decorator (Decorador): Es un patrón estructural que permite que las clases puedan expandirse dinámicamente mientras el *software* está funcionando, dicho más fácil permite añadir responsabilidades extras a objetos concretos de manera dinámica. Su objetivo es hacer que los componentes del *software* orientados a objetos sean más flexibles y fáciles de utilizar. Este patrón se evidencia en la clase controller utilizando el decorador *@RestController* el cual hace referencia que esta clase es una clase controladora (Carlos A. Guerrero, 2013).

2.7 Diagrama de clases

A continuación, se muestra un fragmento del diagrama de clases asociado a las reservaciones. Ver figura 1.

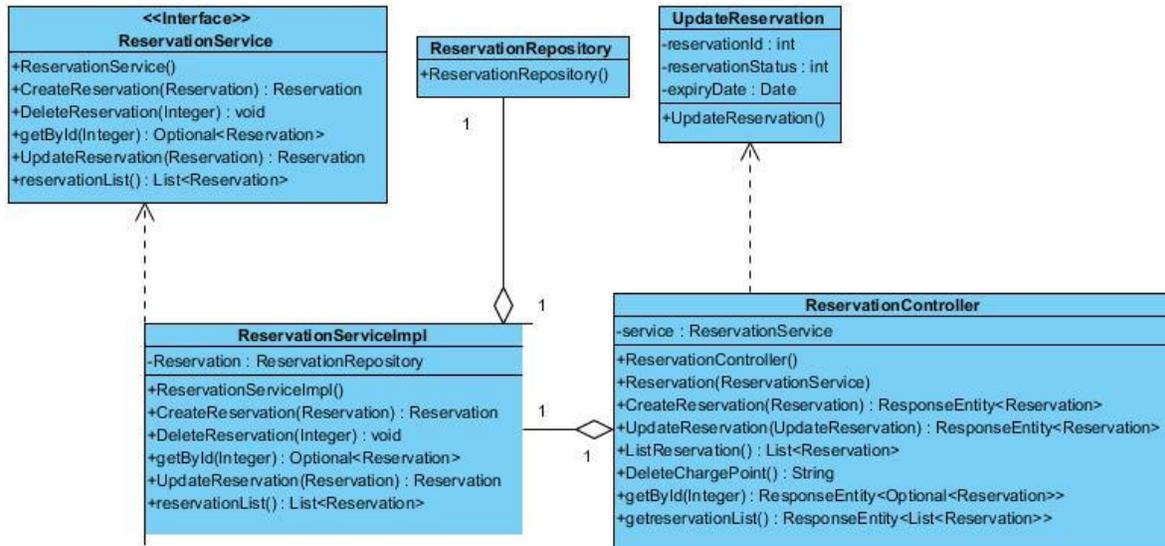


FIGURA 1: DIAGRAMA DE CLASES.

Conclusiones del capítulo

Durante este capítulo se abordaron los elementos para poder diseñar, monitorear y llevar el análisis de un sistema son procesos imprescindibles para el desarrollo de cualquier sistema que se quiera emprender.

- El uso de la metodología AUP-UCI en su escenario #4 como guía del proceso de desarrollo de la herramienta permitió realizar un trabajo organizado y estructurado, generando los artefactos correspondientes a cada una de sus etapas. Proporcionó ventajas a la hora de ver he investigar el comportamiento de sus componentes.

- Fueron definidos 20 requisitos funcionales, los cuales fueron descritos haciendo el uso de las historias de usuarios.
- Fueron definidos 10 requisitos no funcionales, los cuales describen las cualidades o características que tendrá el sistema.
- Se definió la arquitectura del sistema, seleccionando el modelo Cliente-Servidor como patrón arquitectónico y apoyándose en patrones de diseño GRASP y GOF, lo que permitió la correcta estructuración del sistema.

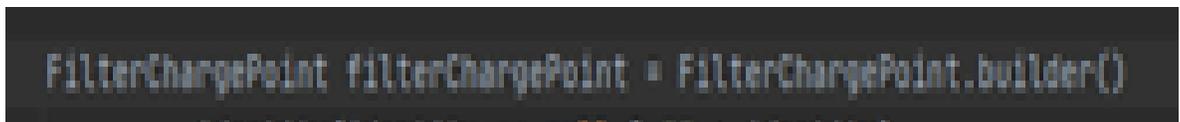
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA

El presente capítulo tiene como principal objetivo lograr la implementación del sistema apoyándonos en el diseño desarrollado en el capítulo anterior. Según la metodología con que se está trabajando AUP-UCI dicha implementación se debe realizar de forma iterativa y ascendente para poder brindar un producto final más próspero y completo para el cliente. Permitiendo que los implicados en este trabajo tengan una mayor experiencia con el desarrollo de este producto para agilizar y fortalecer sus conocimientos en futuros trabajos. También se describirán los estándares de codificación, el diagrama de paquetes, los estilos de métodos, el diagrama de despliegue y se expondrán los conjuntos de pruebas realizadas al sistema de cada versión.

3.1 Estándar de codificación

Entendemos como estándar de código a un conjunto de conversaciones establecidas de ante mano denominaciones. Si bien los programadores deben implementar un estándar de forma prudente y tender siempre al formato más práctico posible, aunque la forma usada va siempre dirigida a la facilidad de comprender el código y devolver ciertas partes realizadas por otros integrantes, así como la depuración de las mismas (ASPL, 2019).

- Se utiliza como estilo de escritura el *Lower Camel Case* esta práctica de escritura consiste en la unión de dos o más palabras sin espacios entre ellas, pero las diferencian una letra minúscula inicial, a partir de la segunda palabra se utiliza mayúscula, por ejemplo: Ver Figura 2.



```
FilterChargePoint filterChargePoint = FilterChargePoint.builder()
```

FIGURA 2: EJEMPLO DE CÓDIGO.

- El idioma que se utiliza es el inglés y el lenguaje de programación Java para los nombres de constantes, variables, estructuras, funciones, clases y métodos de clases, por ejemplo: Ver Figura 2.
- En la nomenclatura podemos ver que el idioma por defecto a la hora de dar sentido funcional al nombre de clases, variables, constantes, entre otros. Será una mezcla entre la nomenclatura tradicional en inglés y la nomenclatura funcional adoptada, por ejemplo: Ver Figura 2.
- Los métodos deberán ser verbos (en infinitivo), en mayúsculas y minúsculas con la primera letra del nombre en minúsculas, y con la primera letra de cada palabra interna en mayúsculas, por ejemplo: Ver Figura 3.

```

public ResponseEntity<Optional<Connector>> obtenerById(@RequestParam Integer id) {
    Optional<Connector> connector = service.getById(id);
    if (connector.isEmpty()) {
        return (ResponseEntity<Optional<Connector>>) ResponseEntity.noContent();
    }
    return ResponseEntity.ok(connector);
}
  
```

FIGURA 3: DECLARACIÓN DE VERBOS.

- Los nombres de las variables tanto de instancia como estáticas reciben el mismo tratamiento que para los métodos, y la longitud del nombre, por ejemplo: Ver Figura 4.

```

Connector connector = Connector.builder()
  
```

FIGURA 4: DECLARACIÓN DE VARIABLE.

3.2 Diagramas

Diagrama de paquetes

Los diagramas de paquetes son diagramas estructurales que se emplean para mostrar la organización y disposición de diversos elementos de un modelo en forma de paquete. Un paquete es una agrupación de elementos del Lenguaje Unificado Modelado (UML) relacionados como diagramas, documentos, clases o incluso otros paquetes. Cada elemento está anidado dentro de un paquete que se representa como una carpeta de archivos dentro del diagrama (Cillero, 2022). A continuación, se muestra el diagrama de paquetes de la propuesta de solución implementada donde se ven relacionadas las clases *controller* que se encarga de coordinar la comunicación entre las peticiones que hace el cliente y comunica a las vistas para que modele los datos peticionados por el cliente. La clase *service* es la encargada de resolver la lógica de negocio dentro de nuestra aplicación. La clase *entity* es la clase donde se definen las columnas de las tablas según los atributos que van a tener cada entidad. La clase *repository* es la encargada de encapsular el almacenamiento, búsqueda y recuperación de una colección de objetos, esta implementación busca que las tareas repetitivas de mantenimiento en las colecciones de datos se realicen de una forma más fácil. En la clase *model* es la encargada de crear cuantas vistas sea necesarias de un conjunto de tablas, se encarga de controlar el formato de los atributos para ajustarnos a un determinado requerimiento. Ver Figura 5.

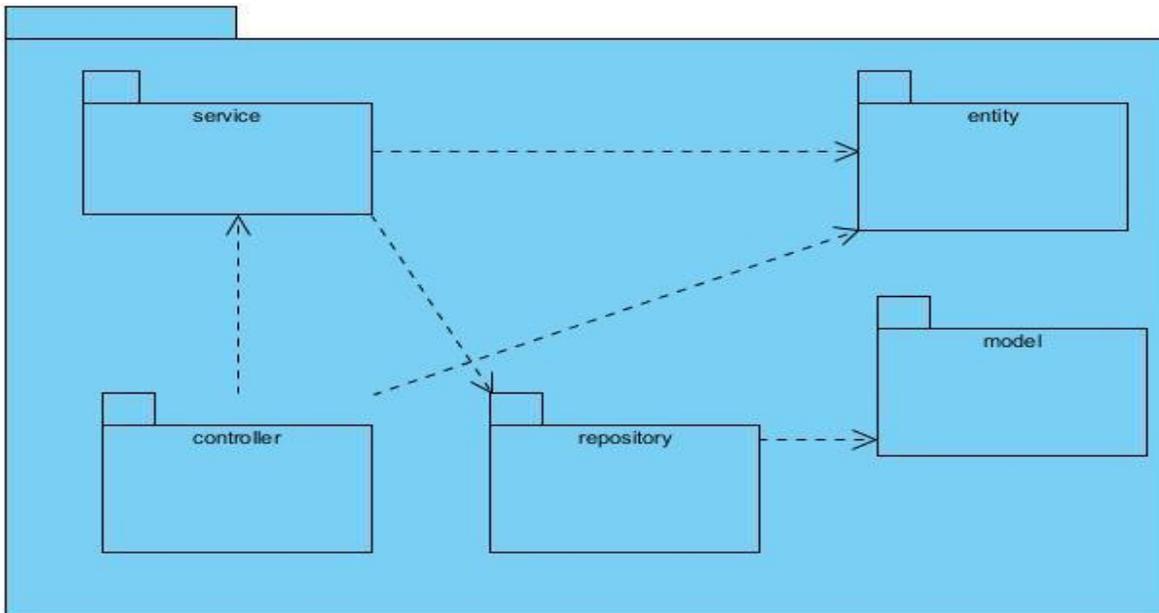


FIGURA 5: DIAGRAMA DE PAQUETES.

Diagrama de despliegue

Los diagramas de despliegues describen la topología del sistema, la estructura de los elementos del *hardware* y el *software* que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicaciones tales como enlaces de red. (Arciniega, 2021). Ver Figura 6.

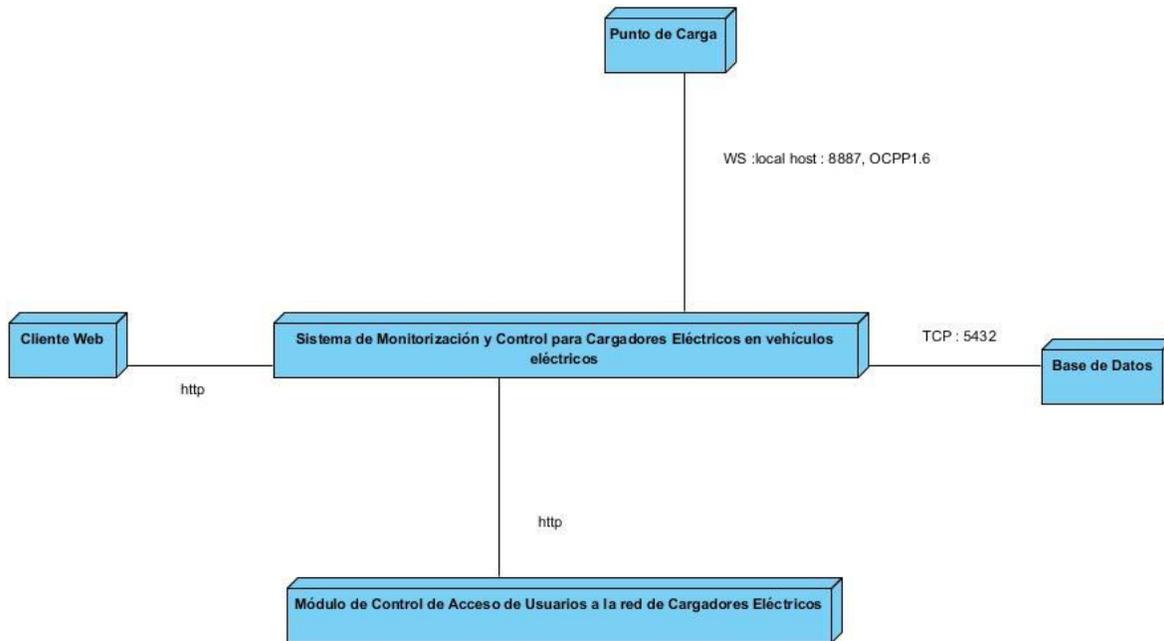


FIGURA 6: DIAGRAMA DE DESPLIEGUE.

3.3 Pruebas

Al desarrollar cualquier *software* es algo muy común que se cometan cualquier tipo de errores que conlleven a disimiles errores en las aplicaciones. Por tal motivo se necesita realizar varias pruebas de *software* para poder erradicar la mayoría de estos errores. Estas pruebas se pueden definir como el conjunto de procesos con los cuales pretende probar un sistema o aplicación con el propósito de comprobar su correcto funcionamiento. Se debe tener presente que estas se pueden realizar en diferentes momentos del desarrollo de la aplicación, desde su creación hasta su instante de despliegue. También se pueden ejecutar de manera automática determinando en cualquier momento si tenemos una aplicación estable para verificar si un cambio en cualquier parte de este sistema puede afectar en otra parte del sistema (Rivero, 2012).

3.3.1 Niveles de pruebas

Para realizar una prueba de *software* se debe tener en cuenta una serie de objetivos en diferentes escenarios y niveles de trabajo. Estas son agrupadas por niveles que se encuentran en diferentes etapas el proceso de desarrollo (Turrado, 2020).

1. Pruebas de aceptación: estas pruebas al igual que las de sistema, están enfocadas en el comportamiento de todo el sistema o producto. Tienen como propósito validar que el sistema está completo, que funcionará según se espera. Verificar que los comportamientos funcionales y no funcionales son los que se especificaron y además proveer información que permita decidir si el sistema está óptimo o no para salir a producción. En este tipo de pruebas el usuario juega un papel muy importante, participando en las mismas.
2. Pruebas de integración: estas pruebas están enfocadas en cómo interactúan los distintos componentes o sistemas entre sí. También buscan encontrar defectos, los cuales pueden estar en las interfaces, en los componentes o sistemas. Pueden ser realizadas tanto por los desarrolladores (generalmente cuando es una prueba de integración de componentes) o por los probadores (generalmente cuando es una prueba de integración entre sistemas).
3. Pruebas de componente o unitarias: estas pruebas están enfocadas en los componentes, unidades o módulos, o lo que es lo mismo, los elementos más pequeños del software. Son realizadas generalmente por el desarrollador y de forma automatizada. Las mismas se encargan de verificar que los comportamientos funcionales y no funcionales del componente son los diseñados y especificados. Además, buscan encontrar defectos y a su vez prevenir la propagación de los mismos en otros niveles de prueba.
4. Pruebas de sistema: estas pruebas se centran en el comportamiento y las capacidades de los sistemas o productos. Entre sus propósitos están: la reducción de riesgos, verificar que los requerimientos funcionales y no funcionales del sistema sean cumplidos y además validar que el sistema

funciona como se espera, generando así confianza en la calidad del sistema. Son realizadas por los probadores.

3.3.2 Métodos de pruebas

Para obtener un genuino diseño de caso de prueba es necesario con mayores probabilidades de descubrir los defectos de un sistema se *software* es necesario emplear métodos de pruebas como: (Rivero, 2012)

Método de la Caja Negra: Este método se basa en introducir datos de entradas y obtiene de este los datos de salidas que se esperan del sistema. Como a la mayoría de usuarios de programas extenso no les interesa los detalles de sus funcionamientos y lo único que les interesa es conseguir respuestas.

Método de Caja Blanca o Caja de Cristal: Este método va dirigido a las funciones internas y la evaluación del código. Utiliza la estructura de control del diseño para así poder derivar los casos de pruebas.

3.4 Pruebas funcionales:

Las pruebas funcionales están enfocadas principalmente en verificar que las funcionalidades especificadas en el sistema cumplan con las especificaciones previstas, por lo que estas están basadas en los requisitos funcionales (Iarrea, 2020).

Estos tipos de pruebas incluyen pruebas unitarias, pruebas de regresión, pruebas de aceptación del usuario, además de muchas otras. Las pruebas funcionales se pueden realizar además usando técnicas de caja negra, las cuales realizan pruebas sobre la interfaz del programa a probar, es decir, sobre las entradas y salidas de dicho programa. Para validar las funcionalidades de nuestro sistema se decidió realizar dos tipos de pruebas fundamentales: prueba de caja negra mediante la técnica de partición de equivalencia y las pruebas unitarias.

3.4.1 Método de caja negra mediante la técnica partición de equivalencia.

Para realizar estas pruebas se han diseñado casos de pruebas, estos miden la funcionalidad en un conjunto de acciones o condiciones para verificar el resultado esperado. Para realizar esto hacen falta un número de datos ya sean válidos o inválidos dependiendo del objetivo planteado ya sea un error o probar una funcionalidad para que ayuden a la ejecución de estos casos. Para el diseño de los mismos se utiliza la técnica de partición de equivalencia, esta técnica es un enfoque efectivo para las pruebas la cual ayuda a la explicación de los errores que se comete con frecuencia a la hora de programar. Esta técnica es aplicable a entradas de datos realizados por personas (Peño, 2015).

A continuación, se expone el caso de prueba realizado al requisito: Modificar una reservación, el resto se encuentra en el Anexo 2.

TABLA 5: REALIZACIÓN DE LA PRUEBA DE CAJA NEGRA AL REQUISITO FUNCIONAL “MODIFICAR UNA RESERVACIÓN”.

Id del escenario	Esce na-rio	Iden tific ador de la rese rvac ión	Esta do de la rese rvac ión	Respuesta del sistema	Resultado de la prueba
3.1	Elimin o	[V,I, N/A]	[V,I, N/A]	Muestra la reservación modificada.	Se modificó de forma satisfactoria la reservación.
	ok	V	V		

3.2	Campos vacíos	V	I	Muestra el mensaje: "Rellene los campos".	Se alertó de forma satisfactoria la presencia de campos vacíos.
		1			
		I	V	Muestra el mensaje: "Rellene los campos".	Se alertó de forma satisfactoria la presencia de campos vacíos.
			2		
3.3		I	I	Muestra el mensaje: "Rellene los campos".	Se alertó de forma satisfactoria la presencia de campos vacíos.
		NA	NA	Muestra el mensaje: Datos Incorrectos.	Se alertó de forma satisfactoria la presencia de campos vacíos.
		Auto	Metro		

3.4.2 Pruebas unitarias

Las pruebas unitarias o *unittesting* son una forma de comprobar que un fragmento de código funciona correctamente. Es uno de los procedimientos que se llevan a cabo dentro de las metodologías ágiles. Consiste en aislar una parte del código y comprobar que funciona a la perfección. Son *test* que validan el comportamiento de un objeto y su lógica (KeepCoding , 2022).

A continuación, se muestra la prueba unitaria asociada a la funcionalidad "Crear una sesión de carga" utilizando el *framework* por defecto que posee Spring Boot.

@SpringBootTest

```
public class ChargingSessionTest {
```

@Autowired

```
private ChargingSessionService chargingSessionService;
```

@Test

```
void testChargingSession() {
```

```
  int id = testCreate();
```

```
  testList();
```

```
  ChargingSession chargingSession = testGetById(id);
```

```
  testStatus(chargingSession);
```

```
  testUpdate(id);
```

```
  testDelete(id);
```

```
}
```

@Test

```
int testCreate() {
```

```
  ChargingSession chargingSession = new ChargingSession();
```

```
  chargingSession = chargingSessionService.CreateReservation(chargingSession);
```

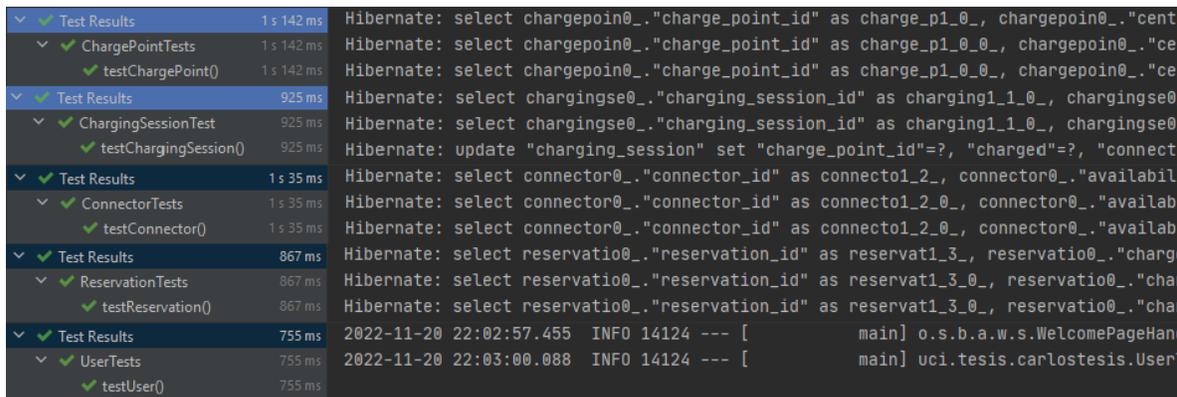
```
  assertThat(chargingSession.getChargingSessionId()).isNotNull();
```

```
  return chargingSession.getChargingSessionId();
```

```
}
```

```
}
```

A continuación, se muestra la Figura 7 donde se ve el resultado de las pruebas unitarias realizadas durante el proceso de implementación, en la que se evidencia la admisión de todas las verificaciones.



```

Test Results 1 s 142 ms Hibernate: select chargepoint0_."charge_point_id" as charge_p1_0_, chargepoint0_."cent
ChargePointTests 1 s 142 ms Hibernate: select chargepoint0_."charge_point_id" as charge_p1_0_0_, chargepoint0_."ce
testChargePoint() 1 s 142 ms Hibernate: select chargepoint0_."charge_point_id" as charge_p1_0_0_, chargepoint0_."ce
Test Results 925 ms Hibernate: select chargingse0_."charging_session_id" as charging1_1_0_, chargingse0
ChargingSessionTest 925 ms Hibernate: select chargingse0_."charging_session_id" as charging1_1_0_, chargingse0
testChargingSession() 925 ms Hibernate: update "charging_session" set "charge_point_id"=?, "charged"=?, "connect
Test Results 1 s 35 ms Hibernate: select connector0_."connector_id" as connecto1_2_, connector0_."availabil
ConnectorTests 1 s 35 ms Hibernate: select connector0_."connector_id" as connecto1_2_0_, connector0_."availab
testConnector() 1 s 35 ms Hibernate: select connector0_."connector_id" as connecto1_2_0_, connector0_."availab
Test Results 867 ms Hibernate: select reservatio0_."reservation_id" as reservat1_3_, reservatio0_."charg
ReservationTests 867 ms Hibernate: select reservatio0_."reservation_id" as reservat1_3_0_, reservatio0_."cha
testReservation() 867 ms Hibernate: select reservatio0_."reservation_id" as reservat1_3_0_, reservatio0_."cha
Test Results 755 ms 2022-11-20 22:02:57.455 INFO 14124 --- [ main] o.s.b.a.w.s.WelcomePageHan
UserTests 755 ms 2022-11-20 22:03:00.088 INFO 14124 --- [ main] uci.tesis.carlostesis.User
testUser() 755 ms
  
```

FIGURA 7: PRUEBAS UNITARIAS.

3.5 Resultado de las pruebas

Al realizar las pruebas se pudieron identificar una serie de no conformidades.

Estas se pueden observar en la siguiente tabla:

TABLA 6: NO CONFORMIDADES.

NO.NC	Requisito Funcional	Descripción	Complejidad	Estado
1	RF1	Error de validación del campo last_heart_beat ya que aceptaba cualquier hora definida.	Media	Resuelto
2	RF5	Error de validación del campo connector_id ya que aceptaba	Media	Resuelto

		que se insertara un identificador ya almacenado.		
3	RF19	No detenía de forma remota una sección de carga.	Baja	Resuelto

En la siguiente figura se muestran los datos correspondientes a cada iteración de prueba por las que transitó el módulo.

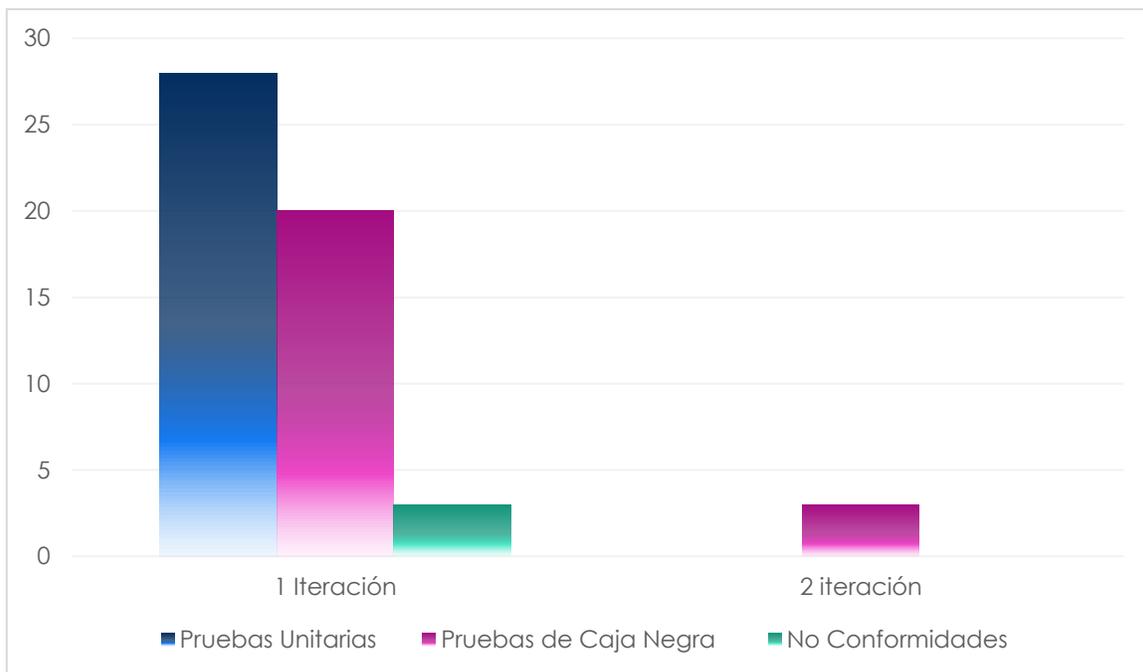


FIGURA 7: ITERACIONES DE PRUEBAS.

En resumen, de las iteraciones de pruebas realizadas se obtuvieron un total de 3 no conformidades (NC) significativas: 2 con complejidad media y una con complejidad baja. Después de concluidas las iteraciones se resolvieron las 3 NC arrojadas quedando el ciclo totalmente cerrado.

Conclusiones del capítulo

Una vez concluido el capítulo se puede llegar a la decisión de que:

- Con la realización de pruebas funcionales se detectaron una serie de no conformidades que fueron corregidas, asegurando así la calidad del sistema.
- Con las pruebas unitarias se comprobó que el código de la aplicación funciona correctamente.

CONCLUSIONES GENERALES

El desarrollo de la presente investigación permitió arribar a las siguientes conclusiones generales:

- Se desarrolló un sistema de monitorización y control de cargadores eléctricos para garantizar el control de todos los puntos de cargas existentes en Cuba.
- Se obtuvo un documento como memoria escrita que detalla y describe la solución implementada.
- Las pruebas realizadas al sistema garantizaron un correcto funcionamiento del sistema.

RECOMENDACIONES

Luego de concluir el trabajo se puede arribar a las siguientes recomendaciones:

- Usar *GitHub* o *GitLab* para el control de versiones.

REFERENCIAS BIBLIOGRÁFICAS

1. Alcalá-Rodríguez, J., Gudiño-Lau, J., Charre-Ibarra, S., Ruíz-Vanoye, J. A., & Fuentes-Penna, A. F.-P. (2018). Sistema de Monit

ALLIANCE, OPEN CHARGE. 2021. Open Charge Point Protocol 1.6. *Open Charge Point Protocol 1.6*. [Online] OPEN CHARGE ALLIANCE, 2021. [Cited: septiembre 22, 2022.] <https://www.openchargealliance.org/protocols/ocpp-16/>.

Analizan uso de vehículos eléctricos. Noticias, Agencia Italiana de. 2022. La Habana : ANSA Latina, 2022.

Arciniega, Fernando. 2021. Tipos de diagramas para representar sitios web. [Online] 2021. [Cited: noviembre 24, 2022.] <https://fernandoarciniega.com/tipos-de-diagramas-para-representar-sitios-web/>.

ASPL. 2019. Estandar de Codificación . *Library*. [Online] Library, 2019. [Cited: 11 4, 2022.] <https://1library.co/article/est%C3%A1ndares-codificaci%C3%B3n-t%C3%ADtulo-sistema-diagn%C3%B3stico-estrat%C3%A9gico-desarrollo-d.yngk3jr0>.

Benítez, Drymon Alfonso. 2017. *Herramienta para generar productos de trabajos de la Habana* : s.n., 2017.

2019. blog.infranetworking. *blog.infranetworking*. [Online] 11 19, 2019. <https://blog.infranetworking.com/servidor-postgresql/>.

Borges, santiago. 2019. Servidor PostgreSQL. [Online] infranetworking, noviembre 19, 2019. [Cited: agosto 5, 2022.] <https://blog.infranetworking.com/servidor-postgresql/>.

Buyto. 2022. BUYTO. *www.buyto.es*. [Online] 2022. [Cited: septiembre 19, 2022.]

Carlos A. Guerrero, Johanna M. Suárez y Luz E. Gutiérrez. 2013. *Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web*. Ciudad Real de Minas : s.n., 2013.

Castaños, Iker. 2021. Semantic Systems . [Online] 2021. [Cited: junio 12, 2022.] <https://www.semantic-systems.com/semantic-noticias/articulos-tecnologicos/en-que-consiste-la-monitorizacion-de-sistemas/>.

Cillero, Manuel. 2009-2022. Manuel.cillero mi circunstancia digital . [Online] 2009-2022. [Cited: septiembre 3, 2022.] <https://manuel.cillero.es/>.

—. **2022.** mi circunstancia digital. [Online] 2022. [Cited: septiembre 4, 2022.] <https://manuel.cillero.es/>.

Contact, Phoenix. 2022. Sistemas de control de carga AC para postes de carga y cargadores de pared. *Sistemas de control de carga AC para postes de carga y cargadores de pared.* [Online] Phoenix Contact, 2022. [Cited: septiembre 22, 2022.] <https://www.phoenixcontact.com/es-pc/productos/tecnologia-de-carga-para-la-electromovilidad/sistemas-de-control-de-carga-ac>.

Critchley, Liam. 2021. Bench Talk for Desing Engineers. [Online] Febrero 4, 2021. [Cited: Septiembre 5, 2022.] <https://material-electrico.cdecomunicacion.es/noticias/proveedores/42839/phoenix-contact-e-mobility-inicia-la-produccion-en-sus-nuevas-instalaciones-de-15-000-metros-cuadrados>.

Desarrollo de una aplicacion para el control remoto de cargadores de coches eléctricos a través de Servicio Web. **Urgel, Daniel Salvador. 2016.** Zaragoza : Univercidad de Zaragoza, Escuela de Ingenieria y Adquitectura , 2016.

Dimática Software. 2022. plug4car.com. *plug4car.com.* [Online] plug4car, 2022. [Cited: septiembre 19, 2022.] <https://plug4car.com/>.

Diseño y desarrollo de un sistema de monitorización remoto de estación de recarga eléctrica. **Sanchez, Francisco Sanchez. 2018.** Cartagena : s.n., 2018.

evblog. 2020. evblog. *blog.placetoplug.com.* [Online] evblog, julio 27, 2020. [Cited: mayo 17, 2022.]

Flores, Frankier. 2022. Qué es Visual Studio Code y qué ventajas ofrece. [Online] julio 22, 2022. [Cited: septiembre 3, 2022.] <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>.

Garcia, Maikol. 2021. *Que es y para qu sirve Java Spring Boot.*

Gonçalves, Manuel José. 2021. ¿Qué es Angular y para qué sirve? ¿Qué es Angular y para qué sirve? [Online] hiberus blog, octubre 13, 2021. [Cited: octubre 4, 2022.] <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/>.

Gonzales, Marcos Vazques. 2021. ¿Qué es Spring Boot?

iberasync. [Online] <https://iberasync.es/arquitectura-cliente-servidor-modelo-de-3-capas/>.

2011. Ingeniería del Software II. *Ingeniería del Software II*. 2011.

Ingeteam. 2022. www.no.linkedin.com. *www.no.linkedin.com*. [Online] Ingeteam, 2022. [Cited: mayo 2022, 19.]

irekisoft. 2022. irekisoft, desarrollo web bajo demanda. *irekisoft*. [Online] 2022. [Cited: septiembre 21, 2022.] <https://www.irekisoft.net/servicios/paginas-web-dinamicas-y-estaticas/>.

irekisoft, Servicios Informaticos para empresa. 2022. irekisoft. *Paginas web Dinamicas y estaticas*. [Online] irekisoft, 2022. [Cited: septiembre 21, 2022.] <https://www.irekisoft.net/servicios/paginas-web-dinamicas-y-estaticas/>.

2018. *Java (lenguaje de programación)*.

JAvier. 2018. *El arte de las cosas*. Habana : s.n., 2018.

Jose Maria Masa Ortega, Antonio Gomez Exposito. 2020. Sistema de Recarga de Vehiculos Electricos: Revisión Tecnológica he Impacto en el Sistema Eléctrico . 2020.

KeepCoding . 2022. ¿Qué son las pruebas unitarias de software? ¿Qué son las pruebas unitarias de software? [Online] agosto 1, 2022. [Cited: noviembre 2022, 21.] <https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/>.

Kord, Maia. 2011. Patrones GRASP. Patrones GoF. Diferencia entre GRASP y GoF. *TuxNots*. [Online] diciembre 3, 2011. [Cited: septiembre 4, 2022.] <https://sites.google.com/site/tuxnotes/materias-de-la-facu/metodologia-de-sistemas/patronesgrasppatronesgofdiferenciaentregraspygof>.

larrea. 2020. Tipos de pruebas de software: diferencias y ejemplos. *Tipos de pruebas de software: diferencias y ejemplos*. [Online] loadview, octubre 16, 2020. [Cited: noviembre 8,

2022.] <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>.

libre, GCFaprende. 2022. Que son las aplicaciones web? . *GCFaprende libre*. [Online] GCFGlobal, 2022. [Cited: septiembre 21, 2022.] <https://edu.gcfglobal.org/es/informatica-basica/que-son-las-aplicaciones-web/1/>.

lugEnergy. 2022. lugEnergy . *Tipos de puntos de recarga para coche eléctrico*. [Online] lugEnergy , 2022. [Cited: septiembre 22, 2022.] <https://www.lugenergy.com/puntos-de-recarga/>.

—. **2022.** lugEnergy . *Tipos de puntos de recarga para coche eléctrico*. [Online] lugEnergy , 2022. [Cited: septiembre 22, 2022.] <https://www.lugenergy.com/puntos-de-recarga/>.

—. **2022.** lugEnergy . *Tipos de puntos de recarga para coche eléctrico*. [Online] lugEnergy , 2022. [Cited: septiembre 22, 2022.] <https://www.lugenergy.com/puntos-de-recarga/>.

lugenergy. 2022. Tipos de puntos de recarga para coche eléctrico. *lugenergy* . [Online] lugenergy , 2022. [Cited: septiembre 22, 2022.] <https://www.lugenergy.com/puntos-de-recarga/>.

lugEnergy. 2022. Tipos de puntos de recarga para coche eléctrico. *lugEnergy*. [Online] lugEnergy, 2022. [Cited: septiembre 22, 2022.] <https://www.lugenergy.com/puntos-de-recarga/>.

Martorell, Sergio Delgado. 2020. 10 ventajas de las aplicaciones web para empresas. [Online] ateinco, junio 18, 2020. [Cited: noviembre 25, 2022.] <https://ateinco.com/10-ventajas-de-las-aplicaciones-web-para-empresas/>.

Méndez, Elizabeth Verdecia Martínez y Rayco Fonseca. 2015. *Extensiones de Visual Paradigm para la generación de producto de trabajo de apoyo a la Especificación de requisitos de software*. Habana : Universidad de las Ciencias Informáticas , 2015.

Novoseltseva, Ekaterina. 2020. *Los 5 Principales Patrones De Arquitectura De Software*. 2019. oasis. [Online] agosto 11, 2019. <https://www.oasis-open.org/committees/download.php/58916/OCPP-1.6.1-DRAFT.pdf>.

Oracle. 2022. Oracle Java. *Oracle*. [Online] Oracle, 2022. [Cited: 5 21, 2022.] <https://www.oracle.com/java/>.

ORBIS. 2020. ORBIS energía inteligente . *ORBIS energía inteligente* . [Online] ORBIS, 6 28, 2020. [Cited: 6 13, 2022.] <https://www.orbis.es/inicio>.

Oscar Blancarte. 2020. *Introduccion a la Adquitectura de Software*. 2020.

Peño, José Manuel Sánchez. 2015. *Pruebas de Software. Fundamentos y Técnicas*. Madrid : s.n., 2015.

Phoenix. 2020. www.tikal.com.uy. *www.tikal.com.uy*. [Online] Phoenix Contact, octubre 23, 2020. [Cited: septiembre 2022, 19.]

2017. pmoinformatica.com. [Online] febrero 6, 2017. <https://pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html?m=1>.

Power, Hive. 2021. OCPP vs IEC 63110 - Protocolo de comunicación abierto para V2G. *Hive Power*. [Online] Hive Power, diciembre 20, 2021. [Cited: mayo 17, 2022.] <https://www.hivepower.tech/es/blog/ocpp-vs-iec-63110-protocolo-abierto-de-comunicaci%C3%B3n-para-v2g>.

Presta, Mati. 2018. *7 razones para utilizar React*.

Rivero, Liudmila Betancourt. 2012. Proceso de prueba de un software. *TINO*. [Online] septiembre 11, 2012. [Cited: noviembre 8, 2022.] <https://revista.jovenclub.cu/proceso-de-prueba-de-un-software/>.

—. 2012. Proceso de prueba de un software. *TINO*. [Online] septiembre 11, 2012. [Cited: noviembre 8, 2022.] <https://revista.jovenclub.cu/proceso-de-prueba-de-un-software/>.

SA, CIRCONTROL. 2022. www.construnario.com. *www.construnario.com*. [Online] CIRCONTROL SA, septiembre 1, 2022. [Cited: septiembre 2022, 19.]

Salvador Urgel, Daniel. 2016. Desarrollo de una aplicacion para el desarrollo de coches electricos a travez de servicios web. *Zaguan*. [Online] Universidad de Zaragoza, 2016. [Cited: septiembre 5, 2022.] <https://zaguan.unizar.es/record/60808?ln=es>.

Sarullo, Federico Alejandro Vazquez. 2022. Protocolo Abierto para Puntos de Carga (OCPP): conexión de las estaciones de carga de vehículos eléctricos a sistemas centrales. *life is on*. [Online] *life is on*, enero 31, 2022. [Cited: marzo 2022, 2022.] <https://blogspanol.se.com/vehiculo-electrico/2022/01/31/protocolo-abierto-para-puntos-de-carga-ocpp-conexion-de-las-estaciones-de-carga-de-vehiculos-electricos-a-sistemas-centrales/>.

Tatiana Grin. 2022. *¿Qué es IntelliJ IDEA?* JetBrains Distributions s.r.o., Czech Republic : 2022.

Turrado, Jorge. 2020. Qué son las pruebas de software. [Online] *campusmvp.es*, marzo 10, 2020. [Cited: noviembre 10, 2022.] <https://www.campusmvp.es/recursos/post/que-son-las-pruebas-de-software.aspx>.

Vazquez, federico Alejandro. 2022. *Life is On. Life is On*. [Online] enero 31, 2022. [Cited: septiembre 16, 2022.] <https://blogspanol.se.com/vehiculo-electrico/2022/01/31/protocolo-abierto-para-puntos-de-carga-ocpp-conexion-de-las-estaciones-de-carga-de-vehiculos-electricos-a-sistemas-centrales/>.

Zacua. 2021. <https://zacua.com>. *https://zacua.com*. [Online] Zacua, enero 27, 2021. [Cited: 9 2022, 19.] <https://zacua.com/que-es-la-electromovilidad/>.

2. oreo para un cargador de baterías. XIKUA Boletín Científico de la Escuela Superior de Tlahuelilpan, 6(11), Article 11. <https://doi.org/10.29057/xikua.v6i11.2773>
3. Bunsen, T., Cazzola, P., Gerner, M., Paoli, L., Scheffer, S., Schuitmaker, R., Tattini, J., & Teter, J. (2018). Global EV Outlook 2018: Towards cross-modal electrification. En *Global EV Outlook 2018: Towards cross-modal electrification* [Report]. International Energy Agency.
4. Cargadores de vehículos eléctricos. (2021, julio 6). Ubor Solar Solvers. <https://uborasolar.es/cargadores-de-coches-electricos/>
5. Casanova Vasco, C. (2019). Desarrollo de un software de monitorización y control para el cargador de baterías de ion litio B6AC V2. <https://academica-e.unavarra.es/xmlui/handle/2454/35889>

6. Cerón, C., & Muñoz, B. (2017). Cargadores para vehículos eléctricos en estacionamientos: Cesbo. <https://repositorio.uchile.cl/handle/2250/176288>
7. Desarrollo de una aplicación para el control remoto de cargadores de coches eléctricos a través de Servicios Web—Repositorio Institucional de Documentos. (s. f.). Recuperado 8 de marzo de 2022, de <https://zagan.unizar.es/record/60808?ln=es>
8. Flor Julián, N. de la. (2015). Diseño de una aplicación para la monitorización del perfil de movilidad de los vehículos [Info:eu-repo/semantics/bachelorThesis]. E.T.S.I. Telecomunicación (UPM). <https://oa.upm.es/37310/>
9. García-Villalobos, J., Zamora, I., San Martín, J. I., Asensio, F. J., & Aperribay, V. (2014). Plug-in electric vehicles in electric distribution networks: A review of smart charging approaches. *Renewable and Sustainable Energy Reviews*, 38, 717-731. <https://doi.org/10.1016/j.rser.2014.07.040>
10. Infraestructura de recarga para electromovilidad | PHOENIX CONTACT. (s. f.). Recuperado 8 de marzo de 2022, de <https://www.phoenixcontact.com/es-ar/industrias/infraestructura-recarga#ex-k4jrp-2>
11. Ingeteam. (s. f.). Recuperado 8 de marzo de 2022, de <https://www.ingeteam.com/es-es/quienessomos/ingeteam.aspx>
12. Inicio—Orbis. (s. f.). Recuperado 8 de marzo de 2022, de <https://www.orbis.es/inicio>
13. OASIS Technical Committees by Name | OASIS. (s. f.). Recuperado 8 de marzo de 2022, de <https://www.oasis-open.org/committees/>
14. Ortega, J. M. M., & Expósito, A. G. (s. f.). Sistema de recarga de vehículos eléctricos: revisión tecnológica e impacto en el sistema eléctrico. 10.
15. Plug and play: Cómo funciona la tendencia en vehículos eléctricos que mejora la experiencia del usuario en la recarga - Portal Movilidad. (2021, mayo 31). <https://portalmovilidad.com/plug-and-play-como-funciona-la-tendencia-en-vehiculos-electricos-que-mejora-la-experiencia-del-usuario-en-la-recarga/>

16. Plug4car. (s. f.). Recuperado 8 de marzo de 2022, de <https://plug4car.com/xmlrpc.php>
17. Recarga de VE y Parking Eficiente—CIRCONTROL. (s. f.). Recuperado 8 de marzo de 2022, de <https://circontrol.com/es/>
18. Sánchez, F. S., & Sala, A. S. M. (s. f.). Sistema de monitorización telemetría de una estación de recarga de vehículos eléctricos usando estándar OCPP 1.6. 93.
19. Zacua. (s. f.). Zacua | Acelera el futuro, auto eléctrico mexicano. Zacua | Acelera el futuro, auto eléctrico mexicano. Recuperado 8 de marzo de 2022, de <https://www.zacua.com/>

ANEXOS

ANEXO 1: HISTORIAS DE USUARIO

TABLA 7: HISTORIA DE USUARIO” ELIMINAR UN PUNTO DE CARGA”.

Historia de usuario	
Número: 3	Nombre: Eliminar un punto de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir eliminar un punto de carga. Para realizar esta funcionalidad solo es necesario introducir el charge_point_id.	

TABLA 8: HISTORIA DE USUARIO” LISTAR PUNTOS DE CARGAS”.

Historia de usuario	
Número: 4	Nombre: Listar puntos de cargas.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir listar los puntos de cargas registrados. Además, debe permitir filtrar estos puntos de cargas y para esto es necesario introducir alguno de los siguientes	

datos, identity, central_system_path, charge_box_serial_number, charge_point_model, charge_point_serial_number, meter_serial_number, meter_type.

TABLA 9: HISTORIA DE USUARIO” REGISTRAR UN CONECTOR DEL PUNTO DE CARGA”.

Historia de usuario	
Número: 5	Nombre: Registrar un conector del punto de carga.
Usuario: Usuario	
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir registrar un conector al punto de carga. Para esto se debe introducir los campos de conector_id, charge_poit_id, availability_tipe_id, charge_point_status, charge_point_error_code_id, charge_status, info, vendor_id, vendor_error_code.	

TABLA 10: HISTORIA DE USUARIO” EDITAR UN CONECTOR DEL PUNTO DE CARGA”.

Historia de usuario	
Número: 6	Nombre: Editar un conector del punto de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	

Descripción: El sistema debe permitir editar el conector del punto de carga. Para poder realizar esta funcionalidad es necesario modificar alguno de los siguientes atributos, conector_id, charge_point_status, info, vendor_error_code.

TABLA 11: HISTORIA DE USUARIO” ELIMINAR UN CONECTOR DEL PUNTO DE CARGA”.

Historia de usuario	
Número: 7	Nombre: Eliminar un conector del punto de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir eliminar el conector del punto de carga. Para realizar esta funcionalidad solo es necesario introducir el connector_id del conector que se desea eliminar.	

TABLA 12: HISTORIA DE USUARIO” LISTAR UN CONECTOR DEL PUNTO DE CARGA”.

Historia de usuario	
Número: 8	Nombre: Listar un conector del punto de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	

Descripción: El sistema debe permitir listar todos los conectores registrados. Además, debe permitir filtrar estos conectores y para esto es necesario introducir alguno de los siguientes datos info, vendor_id, vendor_error_code.

TABLA 13: HISTORIA DE USUARIO” REGISTRAR UNA SESIÓN DE CARGA”.

Historia de usuario	
Número: 9	Nombre: Registrar una sesión de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir registrar una sesión de Carga. Para esto se debe introducir los campos de charging_session_id, transaction_id, user_id, charge_point_id, connector_id, reasons_id, time_stamp_start, meter_start, time_stamp_stop, duration, charged, reservation_id.	

TABLA 14: HISTORIA DE USUARIO” EDITAR LA SESIÓN DE CARGA”.

Historia de usuario	
Número: 10	Nombre: Editar la sesión de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1

Programador a cargo: Carlos Alberto León Vilches
Descripción: El sistema debe permitir editar la sesión de carga. Para poder realizar esta funcionalidad es necesario modificar alguno de los siguientes atributos charging_Session_id, time_stamp_start, meter_start, time_stamp_stop, meter_stop, duration, charged.

TABLA 15: HISTORIA DE USUARIO” ELIMINAR UNA SESIÓN DE CARGA”.

Historia de usuario	
Número: 11	Nombre: Eliminar una sección de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir eliminar una sesión de carga. Para realizar esta funcionalidad solo es necesario introducir el charging_Session_id de la sección de carga que se desea eliminar.	

TABLA 16: HISTORIA DE USUARIO” LISTAR UNA SESIÓN DE CARGA”.

Historia de usuario	
Número: 12	Nombre: Listar una sesión de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	

Descripción: El sistema debe permitir listar las sesiones de cargas registradas. Además, debe permitir filtrar estas secciones de cargas y para esto es necesario introducir alguno de los siguientes datos charging_session_id, transaction_id, user_id, charge_point_id, connector_id, charged, reservation_id.

TABLA 17: HISTORIA DE USUARIO” REGISTRAR UNA RESERVACIÓN DE UN CONECTOR EN EL PUNTO DE CARGA”.

Historia de usuario	
Número: 13	Nombre: Registrar una reservación de un conector en el punto de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir registrar una reservación de un conector en el punto de carga. Para esto se debe introducir los campos de reservation_id, charge_point_id, connector_id, user_id, reservation_status, expiry_date.	

TABLA 18: HISTORIA DE USUARIO” EDITAR UNA RESERVACIÓN DE UN CONECTOR EN EL PUNTO DE CARGA”.

Historia de usuario	
Número: 14	Nombre: Editar una reservación de un conector en el punto de carga.
Usuario: Sistema	

Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir editar una reservación de un conector en el punto de carga. Para poder realizar esta funcionalidad es necesario modificar alguno de los siguientes atributos reservation_id, reservation_status, expiry_date.	

TABLA 19: HISTORIA DE USUARIO” ELIMINAR UNA RESERVACIÓN DE UN CONECTOR EN EL PUNTO DE CARGA”.

Historia de usuario	
Número: 15	Nombre: Eliminar una reservación de un conector en el punto de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir eliminar una reservación de un conector en el punto de carga. Para realizar esta funcionalidad solo es necesario introducir el atributo reservation_id de la reservación que se desea eliminar.	

TABLA 20: HISTORIA DE USUARIO” LISTAR UNA RESERVACIÓN DE UN CONECTOR EN EL PUNTO DE CARGA”.

Historia de usuario

Número: 16	Nombre: Listar una reservación de un conector en el punto de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir listar las reservaciones de cargas registradas. Además, debe permitir filtrar estas reservaciones de cargas y para esto es necesario introducir alguno de los siguientes datos reservation_id, charge_point_id, connector_id, user_id, reservation_status, expiry_date.	

TABLA 21: HISTORIA DE USUARIO” PROCESAR EL METRADO PROCEDENTE DE UN PUNTO DE CARGA EN EL PROCESO DE CARGA”.

Historia de usuario	
Número: 17	Nombre: Procesar metrado.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir procesar el metrado procedente de un punto de carga en el proceso de carga.	

TABLA 22: HISTORIA DE USUARIO” INICIAR UNA SESIÓN DE CARGA DE FORMA REMOTA”.

Historia de usuario	
Número: 18	Nombre: Iniciar una sesión de carga de forma remota.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir que se inicie una sesión de carga de forma remota. Se va a insertar un botón para que el usuario de forma remota después de haber realizado la autenticación pueda iniciar el proceso de carga de forma remota.	

TABLA 23: HISTORIA DE USUARIO” DETENER UNA SESIÓN DE CARGA DE FORMA REMOTA”.

Historia de usuario	
Número: 19	Nombre: Detener una sesión de carga de forma remota.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir que se detener una sesión de carga de forma remota. Se va a insertar un botón para que el usuario de forma remota después de haber realizado el proceso de carga pueda detener este proceso de forma remota.	

TABLA 24: HISTORIA DE USUARIO” MOSTRAR UN LISTADO DE LOS USUARIOS AUTORIZADOS A CARGAR EN EL PUNTO DE CARGA”.

Historia de usuario	
Número: 20	Nombre: Mostrar un listado de los usuarios autorizados a cargar en el punto de carga.
Usuario: Sistema	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración: 1
Programador a cargo: Carlos Alberto León Vilches	
Descripción: El sistema debe permitir mostrar un listado de los usuarios autorizados a cargar en el punto de carga. El sistema de monitorización mediante una API-Rest le pedirá al Sistema de Recarga de Cargadores Eléctricos una lista de todos los usuarios que estarán conectados de forma remota en un punto de carga determinado.	

ANEXO 2: PRUEBAS DE CAJA NEGRA MEDIANTE LA TÉCNICA DE PARTICIÓN DE EQUIVALENCIA.

TABLA 25: REALIZACIÓN DE LA PRUEBA DE CAJA NEGRA AL REQUISITO FUNCIONAL “ELIMINAR UN CONECTOR DEL PUNTO DE CARGA”.

	Escena- rio	Identificador del conector	Respuesta del sistema	Resultado de la prueba
2.1	Registro ok	[V,I,N/A]	Muestra un mensaje con el conector eliminado.	Se eliminó de forma satisfactoria el conector.
		V		
		I		
2.2	Campos vacíos	I	Muestra el mensaje: “Rellene los campos”.	Se alertó de forma satisfactoria la presencia de campos vacíos.