

Universidad Católica del Uruguay

Facultad de Ingeniería y Tecnologías



**Prototipo de un Sistema de
Alimentación de Vehículo Eléctrico
(SAVE)**

Autores:

Victoria Borges

Diego González

Diego Irazoqui

Orientador:

Dr. Alfredo Arnaud

**Memoria de Grado presentada como requisito parcial para la
obtención del grado de Ingeniero en Telecomunicaciones e
Ingeniero en Sistemas Eléctricos de Potencia**

Montevideo, marzo de 2020

Resumen

Este trabajo de grado se enmarca un proyecto de colaboración propuesto entre UTE y el centro ICT4V (*Information and Communication Technology for Verticals* [1]). Se plantea el montaje de un prototipo de Sistema de Alimentación de Vehículos Eléctricos (en adelante SAVE) sobre la base del kit open source desarrollado por *OpenEVSE* [2] para el uso privado, adaptando el mismo para que forme parte de una red de puntos de carga de acceso público. Además, se plantea la integración del prototipo con los sistemas de identificación de clientes, de gestión de carga y de incentivos basados en la tecnología blockchain con los cuales UTE trabaja de forma experimental. Se busca identificar y desarrollar las oportunidades que presenta la aplicación de la tecnología blockchain a las redes eléctricas inteligentes.

El trabajo se realiza con el apoyo de SimpleTECH [3] como auspiciante, brindando apoyo técnico y un equipo de tipo *Smart Vending Machine*, que cuenta con un Raspberry Pi integrada a métodos de pago electrónico y con un módulo que permite la comunicación por RFID. Se plantea la integración de este equipo con el kit *OpenEVSE* de forma de agregar las funcionalidades de identificación, control y forma de pago al punto de carga.

El objetivo principal del proyecto es el desarrollo de un prototipo de SAVE público a partir de uno destinado al uso privado en una vivienda. El proyecto permitirá adquirir experiencia sobre el funcionamiento de dichos dispositivos para evaluar su posterior fabricación y montaje en Uruguay, y por otro lado tener mayor control sobre las funcionalidades y la evolución del sistema de recarga.

De forma resumida, se lograron los objetivos logrando interactuar con las herramientas de UTE y la tecnología blockchain. Se probaron varios casos de uso y la carga de un vehículo de 7,4 kW.

En el primer capítulo se realizará una introducción al problema, su justificación, se resumen los resultados esperados en función de los objetivos planteados, y se realiza una presentación del estado de la tecnología tanto en Uruguay como en el mundo.

El segundo capítulo presenta el contexto del proyecto en cuanto a protocolos, tecnologías a integrar y la normativa de Uruguay para puntos de carga. En el siguiente capítulo, se aborda el kit desarrollado por *OpenEVSE*. Se listan las componentes y registran las pruebas de funcionamiento realizadas sobre el mismo. A partir de ello, se identifican las funcionalidades que se deben agregar para convertirlo en un punto de carga de una red pública.

En el capítulo cuatro, como resultado del estudio anterior, se define la arquitectura propuesta para componentes y comunicación del nuevo punto de carga. Luego se describen las funcionalidades desarrolladas por el equipo de proyecto y las pruebas de funcionamiento del nuevo diseño.

Por último, se plantea un estudio de costos y modelo de negocio potencial para el prototipo.

Abstract

This paper is part of a collaboration project between UTE and the ICT4V centre (Information and Communication Technology for Verticals [1].) We propose the assembly of a prototype of an Electric Vehicle Power System (hereinafter, SAVE) based on the open source kit developed by OpenEVSE [2] for private use, by adapting it to become a part of a network of public access chargers. In addition, we propose that this prototype be incorporated into the customer identification, cargo management and incentive systems based on blockchain technology with which UTE is working on experimental basis. The aim is to identify and develop the opportunities offered by the application of blockchain technology to smart grids.

The work is carried out with the support of SimpleTECH [3] (sponsor), providing technical support and a Smart Vending Machine equipment, which has Raspberry Pi integrated into electronic payment methods and a module that allows RFID communication. We propose the incorporation of this equipment to the OpenEVSE kit to add the identification, control and payment functionalities to the loading point.

The main purpose of the project is the development of a public SAVE prototype based on one intended for private use at home. The project will allow us to acquire experience on the operation of these devices to evaluate their subsequent manufacture and assembly in Uruguay. On the other hand, it will also allow us to have greater control over the functionalities and evolution of the recharging system.

In summary, the objectives were achieved by interacting with UTE tools and blockchain technology. We tested several cases and the load of a 7.4 kW vehicle.

In the first chapter, we introduce the problem and its justification. The expected results are summarized based on the proposed objectives, and we present the state of technology in Uruguay and in the world.

In the second chapter, we provided context of the project in terms of protocols, technologies to integrate and the Uruguayan regulations for loading points. In the next chapter, we discuss the kit developed by OpenEVSE. The components are listed and the performance tests carried out on it are recorded. Based on this, we identify the functionalities that must be added to make it a load point of a public network.

In chapter four, as a result of the previous study, we define the proposed architecture for components and communication of the new charging point. Then, we describe the functionalities developed by the project team and the functional tests of the new design.

Finally, we propose a study of costs and potential business model for the prototype.

Glosario

Backend

Parte del desarrollo que se encarga de la lógica para que el sistema funcione.

Baudrate

Unidad de medida utilizada en telecomunicaciones, que representa el número de símbolos por segundo en un medio de transmisión digital.

CargaME

Sistema propio de UTE para monitoreo y control de la red de cargadores de vehículos eléctricos.

Chaincodes

Programa en el que se implementa la lógica de aplicaciones que se ejecutan sobre una blockchain Hyperledger Fabric.

Contactor

Equipo eléctrico utilizado para comandar la alimentación de una carga eléctrica. El mismo cuenta con contactos principales y una bobina energizada externamente. Los contactos principales se conectan al circuito que se quiere alimentar. Cuando la bobina del contactor queda excitada por la circulación de la corriente, esta mueve el núcleo en su interior y arrastra los contactos principales y auxiliares, estableciendo a través de los polos, el circuito entre la red y el receptor.

Control Pilot

Pin encargado de comunicar estados del vehículo y corriente disponible para la carga, de acuerdo según el protocolo IEC 62169.

Duty Cycle

Es la relación entre la porción de tiempo en que una señal periódica está en su valor alto y el período de la señal.

JSON

Formato de texto para el intercambio de datos.

Mangón / Conector

Accesorio colocado en la punta del cable para conectar al vehículo.

Merkle Tree

Estructura de datos en forma de árbol, donde cada dato está etiquetado con un hash de la concatenación de la etiqueta de sus hijos.

Modulación por ancho de pulso (PWM)

Técnica en la que se modifica el ciclo de trabajo de una señal periódica.

OCPP

Protocolo de comunicación establecido por la Open Charge Alliance entre un punto de carga y un sistema central de gestión de carga.

OpenEVSE

Punto de carga doméstico, de arquitectura y código abierto, fabricado y vendido por la marca de mismo nombre.

Proximity Pilot

Pin encargado de comunicar que se encuentra conectado un vehículo eléctrico, de acuerdo al protocolo IEC 62169.

Punto de carga

Es el equipamiento eléctrico que permite conectar un vehículo y realizar la carga del mismo.

Raspberry Pi

Ordenador de placa única desarrollado por Raspberry Pi Foundation.

RFID

Sistema de autenticación mediante una tarjeta magnética excitada por radio frecuencia.

SAE J1772

Protocolo de comunicación entre un punto de carga y un vehículo eléctrico, desarrollado por la Society of American Engineers (SAE).

SDK

Conjunto de herramientas de software que permite la creación de una aplicación informática para un sistema en particular.

Sistema de Alimentación de Vehículos Eléctricos (SAVE)

Se les llama a todos los elementos que componen la arquitectura de un sistema destinado a la carga de vehículos eléctricos. En Uruguay se le llama comúnmente SAVE al punto de carga.

Smart charging

Funcionalidad que busca optimizar la carga de un vehículo, tomando en cuenta la disponibilidad de energía de la red, la energía del vehículo, la ubicación del punto de carga y otros factores.

Smart grid

Término utilizado para referirse a una red de energía eléctrica que en base a la recolección de datos (consumos, energía generada, clima) gracias al despliegue de Tecnologías de Información y Comunicación despacha la energía de forma inteligente y eficiente.

Timestamp

Es una indicación de fecha/hora correspondiente a un evento. Son la cantidad de milisegundos que transcurrieron desde el 1 de enero del 1970.

Índice de contenido

Glosario	4
Índice de contenido	6
Índice de ilustraciones	8
1. Introducción	10
1.1. Definición del problema	10
1.2. Objetivo	10
1.3. Motivación y justificación del proyecto	11
1.4. Estado del Arte	12
1.4.1. Movilidad eléctrica	12
1.4.2. Blockchain	14
2. Validación inicial.....	16
2.1. Contexto	16
2.1.1. RFID	16
2.1.2. OCPP	16
2.1.3. SAE J1772 y IEC62169.....	17
2.1.4. Flujo de carga típico	20
2.1.5. Normativa en Uruguay	21
2.1.6. Blockchain	22
2.2. Cargador OpenEVSE	23
2.2.1. Componentes	23
2.2.2. Prueba de funcionamiento kit OpenEVSE	25
2.3. Identificación de mejoras a implementar	28
3. Arquitectura propuesta.....	29
3.1. Identificación RFID	29
3.2. Display	29
3.3. Conexión con el vehículo.....	30
3.4. Blockchain	31
3.5. OCPP 1.6.....	34
3.6. Comunicación 3G	34
3.7. Forma de pago.....	34
4. Funcionalidades desarrolladas.....	35
4.1. Scripts desarrollados	38
4.1.1. rfid_reading_loop.py.....	38
4.1.2. evse_functions.py.....	38
4.1.3. ocpp_chargepoint_class.py.....	39
4.1.4. charge_point.py	40
4.2. Blockchain	41

5.	Pruebas y simulaciones del prototipo.....	42
5.1.	Proceso de carga	42
5.1.1.	Etapa 0: Energización del punto de carga	42
5.1.2.	Etapa 1: Identificación de usuario.....	43
5.1.3.	Etapa 2: Inicio de transacción	44
5.1.4.	Etapa 3: Carga del vehículo	44
5.1.5.	Etapa 4: Cierre de sesión de carga	45
5.2.	Pruebas con simulador de UTE.....	45
5.3.	Prueba de carga con vehículo eléctrico	46
5.4.	Pruebas remotas	47
5.5.	Resultados	48
5.6.	Pruebas blockchain	49
5.6.1.	Red local con Hyperledger Fabric.....	49
5.6.2.	Light client en Raspberry PI.....	50
5.6.3.	Red en IBM Cloud.....	50
6.	Viabilidad de implementación del prototipo en el mercado	51
6.1.	Costos.....	51
6.2.	Modelo de negocio para el cargador	54
6.2.1.	Quién	54
6.2.2.	Qué - Game Changing.....	54
6.2.3.	Cómo	55
6.2.4.	Cuánto	55
7.	Plan de Trabajo y Cronograma.....	56
8.	Conclusiones.....	59
9.	Referencias.....	61
10.	Anexos.....	63
10.1.	Registro pruebas IEC 62169	63
10.1.1.	Medidas para 10A	63
10.1.1.	Medidas para 30A	64
10.1.1.	Medidas para 60A	65
10.2.	Plano punto de carga trifásico 22 kW	67
10.3.	Comandos RAPI OpenEVSE	68
10.4.	Log de carga remota.....	74

Índice de ilustraciones

Ilustración 1 – Arquitectura de solución propuesta	11
Ilustración 2 - Ruta eléctrica de UTE actual [4]	14
Ilustración 3 – Circuito PP en SAE J1772 [14].....	19
Ilustración 4 – Esquema conector tipo 2 [15]	20
Ilustración 5 - Esquema de carga en OCPP 1.6	21
Ilustración 6 – Componentes del OpenEVSE (tapa)	23
Ilustración 7 – Componentes del OpenEVSE ensamblado (interior)	23
Ilustración 8 – Circuito para simulación comunicación Control Pilot	25
Ilustración 9 – Señal vista en CP desconectado	26
Ilustración 10 – Señal en CP simulando EV conectado	27
Ilustración 11 – Señal en CP simulando carga.....	27
Ilustración 12 – Señal en CP simulando Ventilación requerida	27
Ilustración 13 – Señal en CP simulando una falla	28
Ilustración 14 – Arquitectura propuesta	29
Ilustración 15 – Arquitectura maestro esclavo	30
Ilustración 16 – Estructura red blockchain.....	33
Ilustración 17 - Diagrama para ejecutar la carga de un vehículo.....	35
Ilustración 18 – Flujo de una sesión de carga	36
Ilustración 19 – Flujo para resolución de errores durante sesión de carga.....	37
Ilustración 20 – Display de los estados y errores en el OpenEVSE	41
Ilustración 21 – Circuito auxiliar con simulador de UTE	46
Ilustración 22 – Carga de vehículo eléctrico en proceso	47
Ilustración 23 – Cronograma inicial	57
Ilustración 24 – Cronograma real vs línea base	58
Ilustración 25 – 10A EV conectado	63
Ilustración 26 – 10A EV cargando	63
Ilustración 27 – 10A ventilación requerida	64
Ilustración 28 – 30A EV conectado	64
Ilustración 29 – 30A EV cargando	64
Ilustración 30 – 30A no conectado	65
Ilustración 31 – 30A error	65
Ilustración 32 – 60A EV conectado	65
Ilustración 33 – 60A EV cargando	66
Ilustración 34 – 60A no conectado	66
Ilustración 35 – 60A ventilación requerida	66

Índice de tablas

Tabla 1 – Objetivos específicos del proyecto.....	10
Tabla 2 – Modos de carga SAE J1772.....	17
Tabla 3 – Niveles de carga.....	17
Tabla 4 – Niveles de voltaje y señales en Control Pilot, la amplitud.....	18
Tabla 5 – Corriente en función de Duty Cycle en Control Pilot	18
Tabla 6 – Estados indicados por PP.....	19
Tabla 7 – Pruebas de funcionamiento IEC 62169	26
Tabla 8 – Verificaciones etapa 0	43
Tabla 9 – Verificaciones etapa 1	43
Tabla 10 – Verificaciones etapa 2	44
Tabla 11 – Verificaciones etapa 3	44
Tabla 12 – Verificaciones etapa 4	45
Tabla 13 – Resultados de las pruebas	49
Tabla 14 – Rubrado de costos de producción de un cargador	53

1. Introducción

1.1. Definición del problema

Actualmente, UTE se encuentra en proceso de fomento e instalación de un sistema de movilidad eléctrica. Para ello, trabaja con sistemas de recarga diseñados y construidos en el extranjero los cuales suministran tanto los cargadores como la plataforma para gestionar la carga. El problema que surge es que se trata de plataformas propietarias cerradas, con información limitada sobre su composición y con muchas limitaciones para adaptar a los requerimientos de UTE. Estos no permiten acceder al firmware lo que imposibilita tanto el mantenimiento de los puntos de carga como el desarrollo de nuevas funcionalidades y su adaptación a nuevos modelos de negocio.

El presente proyecto se desarrolla en dos áreas de trabajo. Por un lado, desde el punto de vista eléctrico, se realiza el ensamblado y conexión del cargador a partir de las componentes estándar, validación de cumplimiento de normativa uruguaya y verificación de prestaciones desde el punto de vista de la entrega y limitación de potencia a la hora de realizar una carga.

Por otro lado, desde el punto de vista de las comunicaciones se abordará el problema en su conjunto. Desde la interfaz usuario-cargador buscando integrar al prototipo algún sistema sencillo e intuitivo que posibilite a cualquier persona en la vía pública realizar una carga de forma autónoma. Por otro lado, también se aborda la comunicación entre el cargador y el sistema de gestión de carga de UTE. Esto implica establecer un intercambio de información con la red de UTE a través de OCPP 1.6. En esta etapa, se evaluará la viabilidad de que el punto de carga integre una red blockchain que permita el registro de una transacción. Se aclara que UTE está interesada en la tecnología blockchain para el manejo de las transacciones públicas, pero aún no se ha implementado nada en su plataforma.

1.2. Objetivo

El objetivo general del proyecto es desarrollar un prototipo de Sistema de Alimentación de Vehículos Eléctricos (SAVE) que pueda formar parte de una red de recarga a partir de un punto de carga destinado a uso doméstico. Para ello, se proponen los siguientes objetivos específicos:

Objetivo específico	Resultado esperado
Controlar SAVE mediante sistema central de UTE	Inicio y detención de carga puede ser realizado desde sistema central
Identificar usuarios mediante RFID y autorización de forma de pago	Sesión de carga se habilita únicamente cuando un usuario se encuentra registrado en el sistema central y con un medio de pago habilitado
Probar viabilidad que el SAVE integre una red blockchain	Punto de carga cumpla requisitos para poder establecer comunicación dentro de una red blockchain
Realizar una demo de una carga	Carga de un vehículo eléctrico exitosa

Tabla 1 – Objetivos específicos del proyecto

Para lograr los objetivos se partirá de componentes de cargador para vehículos eléctricos del kit OpenEVSE [2]. Se buscará transformar dicho kit de uso doméstico en un punto de carga que pueda formar parte de una red de recarga. Se realizará una validación de forma que se pueda identificar las componentes faltantes para cumplir los objetivos trazados. Además, se evalúa la integración de una placa desarrollada por SimpleTech, la cual cuenta con un sistema de identificación y pago, conformada por una Raspberry Pi y un módulo para la comunicación

RFID. Dentro de la validación, también se verificará el cumplimiento de los protocolos de comunicación entre vehículo y punto de carga (SAE J1772 y IEC 62169) y con el sistema central de carga (OCPP 1.6). Los protocolos se presentan en los capítulos 2.1.2 y 2.1.3.

Luego se estudiará y evaluará la viabilidad de realizar transacciones basadas en blockchain. Esto va alineado con la experimentación en smart grid de UTE previsto para los próximos años [4] [5].

Finalmente, se evaluará la posibilidad de la fabricación y/o implementación en Uruguay, así como comparar ventajas y desventajas contra otros dispositivos ya existentes en el mercado.

La siguiente imagen representa un diagrama de componentes del prototipo final. La arquitectura cuenta con una Raspberry Pi como núcleo de la solución, siendo la encargada de establecer la comunicación entre todas las partes involucradas.

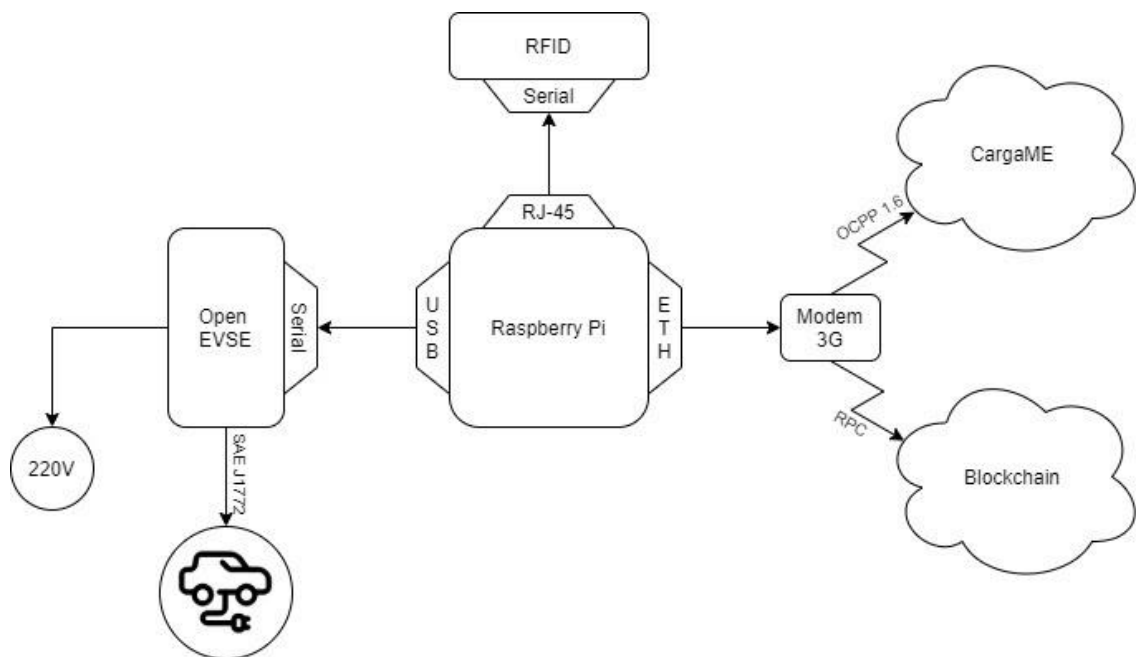


Ilustración 1 – Arquitectura de solución propuesta

1.3. Motivación y justificación del proyecto

Se identifican dos principales focos para llevar adelante el proyecto. Por un lado, la necesidad de facilitar el desarrollo de una red de puntos de carga en Uruguay. Por otro, la búsqueda de innovar en cuanto a las capacidades ofrecidas por los SAVE existentes en el mercado, puntualmente en la modalidad para llevar adelante las transacciones y capacidad de adaptación a nuevos modelos de negocio, que permitan, en particular, gestionar de manera inteligente el proceso de carga para adaptarlo a las necesidades del sistema eléctrico.

Uruguay, a través de la UTE como su gestor de red eléctrica, está promoviendo la utilización de la movilidad eléctrica, lo cual implica un cambio de infraestructura e inversión no solamente en los puntos de carga (hardware) sino de un sistema (software) que pueda gestionar dichos puntos. Dado que a nivel mundial la movilidad eléctrica es una alternativa relativamente nueva, aún no hay estándares definitivos sobre los cuales basarse. Los mismos se encuentran en proceso de creación, impulsados por una asociación de proveedores (Open Charge Alliance [6]) de puntos de carga que han fijado ciertos criterios comunes para permitir la interoperabilidad de distintos SAVE que pertenecen a una misma red eléctrica. Al momento, el gestor de la red eléctrica se ve obligado a obtener tanto el punto de carga como el sistema para gestionar la

misma de un único proveedor. Esto genera una relación de dependencia y rigidez del servicio ofrecido con dicho proveedor, lo cual entorpece y encarece el desarrollo de una red pública de carga para el sistema de movilidad eléctrica.

En ese contexto, se identifica como un punto de optimización el desarrollo de un SAVE propio, que permita flexibilidad para adaptarse a los cambios de la tecnología y modelos de negocio. Las principales ventajas de la optimización estarían en los siguientes rubros:

- **Costos:** se evita la dependencia de los proveedores y también la obsolescencia temprana del equipo incorporado, teniendo la posibilidad de mantenerlo y actualizarlo según las necesidades que surjan. Además, disminuirían los costos de mantenimiento ante fallas o roturas de los componentes.
- **Compatibilidad:** hay garantía de que el equipo funciona, a pesar de que dentro de la red de puntos de carga existan SAVE de distintos proveedores.
- **Flexibilidad de la demanda:** Se busca que los SAVE que formen partes de redes de carga desplegada por privados puedan ofrecer servicios de flexibilidad de la demanda de energía eléctrica, para por ejemplo adaptar la carga a la disponibilidad de energía eólica.
- **Potencial producto o servicio a ofrecer:** se generaría un valor nuevo en la empresa, con la posibilidad de ofrecer tanto en el mercado interno como en lo regional un producto propio, generando una nueva fuente de ingreso.

A futuro existe un gran potencial de desarrollo de la participación de los vehículos eléctricos dentro de una red eléctrica. Para adaptarse a las variaciones en la generación y demanda de energía eléctrica, sería deseable que los vehículos inyecten energía a la red en lo que se conoce como Vehicle-To-Grid (V2G). Esto refuerza el interés en tener control sobre el SAVE, de forma de adaptarlo a los cambios requeridos.

Un punto de innovación respecto al SAVE que se quiere desarrollar es incorporar la tecnología blockchain para obtener trazabilidad y transparencia sobre las transacciones, siendo esta una forma de auditoría.

En Uruguay, la Agencia de Gobierno Electrónico y Sociedad de la Información y del Conocimiento (AGESIC) [7] es la agencia que tiene como cometido procurar la mejora de los servicios al ciudadano, utilizando las posibilidades que brindan las tecnologías de la información y las comunicaciones (TIC). Asimismo, impulsará el desarrollo de la Sociedad de la Información en el Uruguay con énfasis en la inclusión de la práctica digital de sus habitantes y el fortalecimiento de las habilidades de la sociedad en la utilización de las tecnologías.

En su Plan de Gobierno Digital para el año 2020 [8] uno de los objetivos planteados es el de Gobierno Eficiente, siendo un gobierno eficiente aquel que busca mejorar los servicios de cada organismo mediante el desarrollo de sistemas de gestión que simplifiquen y unifiquen los procesos transversales entre ellos. Una de las patas de sostén de este gobierno digital es el generar avances en la digitalización de la gestión de los registros públicos facilitando instrumentos normativos, institucionales y técnicos que faciliten el uso y desarrollo de nuevas tecnologías para la validez de los registros como lo es blockchain.

De esta manera, es de interés validar la viabilidad de incorporar blockchain en el SAVE, convirtiéndolo en un potencial diferencial dado que de momento no hay en el mercado nacional un cargador con esta característica.

1.4. Estado del Arte

1.4.1. Movilidad eléctrica

En el mundo actual, el medio ambiente ha tomado un rol prioritario dado el cambio climático acelerado que se está viviendo que deriva en eventos extremos e inesperados. Se han

iniciado innumerables acciones y acuerdos con el fin de contrarrestar ese avance. En ese marco, el cambio de la matriz energética en los países (Uruguay es buen ejemplo) es uno de los pasos realizados, buscando ir rumbo a la generación basada en fuentes renovables. Particularmente en Uruguay, hubo un rápido desarrollo de los parques eólicos los cuales logran aportar hasta el 40% de la energía demandada por el país. Esto genera que haya excedentes de energía renovable debido a la variabilidad de la disponibilidad del recurso. En ese contexto, el vehículo eléctrico emerge como un recurso ideal ya que su consumo de energía está relacionado habitualmente a la noche, momento en que hay excedente de energía, aportando flexibilidad en el consumo para adaptarse a las variaciones antes mencionadas. De esto resulta que sea estratégico para Uruguay el desarrollo de la movilidad eléctrica.

A la par de esto, el paradigma de la movilidad también se está reconfigurado para perseguir el mismo fin: sustituir los hidrocarburos por energía eléctrica que disminuya la contaminación y torne más eficiente la circulación por las ciudades. Hay una tendencia mundial cada vez más intensa para sustituir los vehículos a combustión tradicionales por vehículos híbridos o totalmente eléctricos. La principal limitación que está sufriendo el avance de la tecnología está relacionada a la autonomía de las baterías para vehículos eléctricos y el tiempo de recarga [9]. Mientras lo común es recorrer cerca de 500 km sin realizar recargas, los vehículos eléctricos de mayor autonomía recién están alcanzando los 250-300 km aproximadamente, aunque los autos más económicos y vendidos rondan en torno a los 150-200 km de autonomía [10]. Sumado a esto, llenar un tanque de combustible insuere unos 2 o 3 minutos, mientras que una carga completa de batería puede tomar horas.

En cuanto a los puntos de carga, existe una gran oferta para ámbitos privados. Esto refiere a equipos pensados para instalación en una vivienda y uso personal. Es decir, se asemeja a un nuevo electrodoméstico cuyo consumo se refleja en la cuenta de energía mensual a la que se está habituado. En cambio, para lugares públicos (rutas) o semipúblicos (empresas o estacionamientos), donde la persona que consume el servicio varía constantemente, no hay soluciones que cubran óptimamente todo el ciclo. Cada fabricante propone una solución completa pero basado en plataformas desarrolladas por sí mismo, lo cual implica más costo de mantenimiento, encarece el precio del producto y da menos flexibilidad para adaptar al uso que se requiera. La suma de dichos factores deriva en que la interoperabilidad de una red de cargadores de diferentes fabricantes sea una tarea muy compleja.

En este contexto, la propuesta de UTE es generar una red eléctrica nacional, con un cargador cada 60km aproximadamente. Teniendo en cuenta esto, es de interés para UTE estudiar la viabilidad del desarrollo en el mercado de un producto SAVE que permita crear una red de recarga sobre la base de puntos de carga instalados por privados que puedan ser compartidos para su uso público. Además, el desarrollo local del software y parte del hardware facilita la adaptación del producto a las necesidades futuras. Esto implica agregar a las funciones tradicionales de un punto de carga (identificación y habilitación de carga) la posibilidad de aceptar pagos e integrar una red blockchain para aportar seguridad, trazabilidad, auditabilidad y el desarrollo de nuevos modelos de negocio. De forma resumida, la red de nodos blockchain podría estar integrada por varios actores involucrados en el servicio de recarga, entre los cuales UTE jugaría un rol importante en la gobernanza, teniendo acceso al control remoto de los procesos de recarga para adaptarlos a la disponibilidad de energía renovable y capacidad de la red eléctrica instante a instante.



Ilustración 2 - Ruta eléctrica de UTE actual [4]

1.4.2. Blockchain

El 31 de octubre de 2008 se publica “Bitcoin: A Peer-to-Peer Electronic Cash System” de Satoshi Nakamoto [11] siendo este el puntapié inicial para el nacimiento de la tecnología blockchain. En este paper, Satoshi propone una versión puramente peer-to-peer de dinero electrónico permitiendo que los pagos en línea se envíen directamente de una parte a otra sin pasar por una institución financiera.

En los siguientes años, blockchain evolucionó no únicamente como una tecnología de intercambio de criptomonedas, sino que también es aplicable en una enorme cantidad de contextos como el sector de la salud, la trazabilidad de mercaderías, el almacenamiento de información, la emisión de certificados, la verificación de identidad, entre otros.

Blockchain [12] se puede resumir como un gran libro de contabilidad distribuido, el cual permite a los participantes escribir y actualizar registros. La criptografía asegura que los registros almacenados permanezcan iguales una vez añadidos al libro contable. Estos registros se guardan en forma de transacciones, las cuales son procesadas y guardadas en bloques. A su vez, cada bloque está vinculado criptográficamente al siguiente bloque. Esta estructura proporciona un método eficiente para verificar cualquier tipo de transacción sin pasar por todos los datos almacenados. Esto tiene como ventajas la descentralización, el manejo de datos en series de tiempo, mantenimiento colectivo, capacidad de programación y la seguridad. Estas características despiertan particular interés para la creación de sistemas monetarios programables y sistemas financieros.

La tecnología blockchain hace posible eliminar el tercero de confianza que se beneficia como intermediario entre las dos partes que requieren de realizar la transacción, siendo esta transacción segura y transparente. Esto lleva a la aparición de blockchains permissionadas, donde la participación pasa a ser solo con permiso para ingresar. Hyperledger.org es un esfuerzo de colaboración de código abierto creado para avanzar en las tecnologías de cadena de bloques de

la industria. Es una colaboración global, organizada por la Fundación Linux, que incluye líderes en finanzas, banca, IoT, cadena de suministro, fabricación y tecnología.

A pesar de la importante inversión en términos de tiempo, dinero y esfuerzo que precisaría, la adopción de la tecnología blockchain en el sector energético podría ampliar la visibilidad, aumentar la eficiencia operativa y agilizar el cumplimiento de las obligaciones de presentación de información

UTE se encuentra trabajando transversalmente en lo referido a ciudades inteligentes y movilidad eléctrica, por lo tanto, el uso de electrodomésticos gestionables (IoT) y la incorporación de vehículos eléctricos serán una realidad en los próximos años, completando la posibilidad de gestionar prácticamente el cien por ciento de la demanda. El mayor desafío y oportunidades de desarrollo de tecnología está asociado con el manejo de datos en gran escala (big data), con la seguridad de los datos personales y con la operación del sistema.

2. Validación inicial

2.1.Contexto

2.1.1. RFID

RFID es una tecnología de identificación por radiofrecuencia, en la cual un lector envía una señal continua dentro de su radio de alcance (determinado por la frecuencia con la que transmite). Cuando una etiqueta o tarjeta (TAG) se encuentra dentro de este radio envía un código de identificación (único) al lector.

2.1.2. OCPP

Originalmente, cada proveedor de puntos de carga establecía una forma de comunicación con el gestor de carga en función de sus necesidades. Con el avance de la tecnología y surgimiento de múltiples proveedores, dicha comunicación se volvió un desafío para los gestores de las redes eléctricas pues se generaban muchos problemas de interoperabilidad al incorporar SAVE de diferentes proveedores. Por este motivo, surge la “Open Charge Alliance” [2] que busca definir criterios comunes entre los fabricantes de puntos de carga y los operadores de las redes eléctricas. Hoy en día, dicha alianza cuenta con más de 120 afiliados y ha creado el protocolo OCPP [13].

El objetivo de OCPP es establecer el protocolo de comunicación entre un punto de carga y el sistema central, desde que se agrega un punto de carga a la red, hasta la autorización y sesión de carga llevada adelante por un usuario. Desde su origen en 2010 hasta el momento, existen múltiples versiones del protocolo. Las más desarrolladas son la 1.5 y 1.6, pero actualmente se encuentra en proceso de implementación la versión 2.0. Es importante tener en cuenta que no todas las versiones del protocolo son compatibles. Es decir, un sistema central implementado en OCPP 1.5 no va a poder comunicarse con un punto de carga basado en OCPP 1.6 y viceversa.

A grandes rasgos, el protocolo establece las funciones que deben ser llevadas adelante en la comunicación. Esto es:

- Listado de funciones.
- Flujo de los diálogos que se establecen entre el sistema central y el punto de carga.
- Formato de la información que se intercambia en cada función.

Adicionalmente, clasifica las funciones en 6 grupos, de los cuales se destacan las funciones “Core” y funciones “Firmware Management”. Ambos grupos de funciones son los básicos que debe poder llevar adelante un punto de carga para cumplir OCPP. El resto de los grupos de funciones apuntan hacia mejoras operativas para un gestor de red eléctrica.

Actualmente, las principales modificaciones que se están llevando adelante apuntan a permitir funciones relacionadas a la gestión inteligente de la carga y al flujo de potencia desde las baterías de los vehículos hacia a la red (en desarrollo en la versión 2.0).

En Uruguay, el sistema central de UTE está basado en OCPP 1.6, por lo que todo punto de carga que se quiera instalar deberá tener dicha versión del protocolo.

2.1.3. SAE J1772 y IEC62169

Previo a indicar las particularidades de las normas, hay que indicar que habitualmente se nombran cuatro modos de carga en función de la infraestructura utilizada:

Modo	Descripción instalación	Capacidad
1	Punto de conexión doméstico	Monofásico o trifásico Hasta 11kW (16A)
2	Punto de conexión doméstico con protección dedicada	Monofásico o trifásico Hasta 22kW (32A)
3	Punto de conexión en AC específico para vehículo eléctrico	Monofásico o trifásico Hasta 43kW (63A)
4	Punto de conexión en DC específico para vehículo eléctrico	De 38kW a 350kW

Tabla 2 – Modos de carga SAE J1772

En este documento, siempre se hará referencia al modo 3 dado que se trata de un punto de carga específico para vehículos eléctricos, en alterna. Aún no se ha desarrollado en Uruguay la carga en continua, modo en el que Estados Unidos lidera mundialmente, principalmente por el desarrollo de los cargadores rápidos de Tesla.

Para la comunicación entre el punto de carga y el vehículo, existen dos normativas que establecen los tipos de conectores, los niveles de carga aceptados y cómo se realiza la comunicación entre el vehículo y el punto de carga.

En Estados Unidos, la SAE (Society of Automotive Engineers) estableció el estándar J1772, en el cual se establecen los siguientes niveles de carga:

Nivel de carga	Alimentación	Rango de potencias
1	AC o DC	0 a 10 kW
2	AC o DC	10 a 50kW
3	DC	Mayor a 50kW (hasta 350kW al momento)

Tabla 3 – Niveles de carga

El nivel 1 está asociado a niveles de tensión hasta 120Vac, utilizado en Estados Unidos. En el caso de Uruguay, aplica la carga con nivel 2 (230Vac monofásico o 400Vac trifásico). En nuestro país, las estaciones de carga más habituales son de 22kW (32A máximo) y últimamente se están instalando totem de 43kW (63A máximo).

Por otro lado, se normalizan por región los tipos de conectores utilizados en cada nivel. La particularidad que comparten todos los conectores es la de poseer dos cables para comunicación de información entre el vehículo y el punto de carga. Los mismos son conocidos como Control Pilot (CP) y Proximity Pilot (PP).

El primero, es el responsable de informar el estado del vehículo eléctrico al punto de carga. Esta información es enviada mediante una señal de onda cuadrada, de 12 V de amplitud y frecuencia 1 kHz. En función de su estado, el vehículo coloca un cierto valor de resistencia entre dicho pin y tierra. En la Tabla 4 se muestran qué estado representa cada valor de resistencia.

Estado	Estado de la carga	Amplitud señal ($\pm 1V$)	Resistencia en CP
Estado A	No conectado	+12Vdc	N/A
Estado B	EV conectado	+9 Vdc	2.74k Ω
Estado C	EV cargando	+6 Vdc	882 Ω
Estado D	Ventilación requerida	+3 Vdc	246 Ω
Estado E	Apagado	0V	N/A
Estado F	Error	-12 Vdc	N/A

Tabla 4 – Niveles de voltaje y señales en Control Pilot, la amplitud

Además, mediante el PWM (Pulse Width Modulation) de la señal, el punto de carga le indica al vehículo la corriente disponible para la carga. A continuación, se muestran los niveles típicos de corrientes:

Duty Cycle	Máxima corriente
10%	6 A
20%	12 A
30%	18 A
40%	24 A
50%	30 A
66%	40 A
80%	48 A
90%	65 A
94%	75 A
96%	80 A

Tabla 5 – Corriente en función de Duty Cycle en Control Pilot

De forma genérica:

- Para rango de corrientes de 6A a 51A:
 - $Duty\ cycle = \frac{Corriente\ (A)}{0.6}$
- Para rango de corrientes de 51A a 80A
 - $Duty\ cycle = \frac{Corriente\ (A)}{2.5} + 64$

El segundo pin (proximity pilot) es donde surge la principal diferencia entre los estándares. De forma genérica, mediante la medición de voltaje entre PP y tierra (en adelante, PE) se informa al punto de carga el estado de la conexión. El voltaje varía en función de las resistencias que se colocan entre PP y PE. En este punto, la normativa americana contempla que los conectores de

los puntos de carga tengan un botón, el cual se presiona cuando se quiere conectar un vehículo. En el caso de Europa, esto no es obligatorio, lo que implica un nivel de voltaje menos. En la Ilustración 3 se muestra el circuito esquemático, contemplando dicho botón en paralelo a una resistencia.

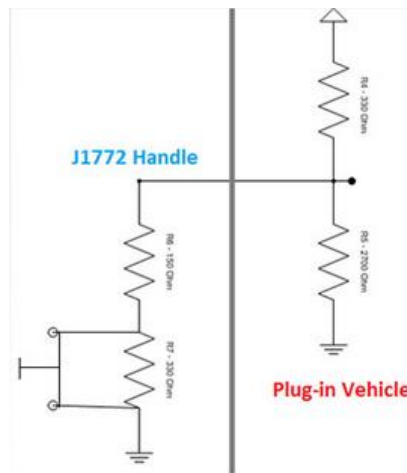


Ilustración 3 – Circuito PP en SAE J1772 [14]

Estado	Voltaje entre PP-PE
No conectado	4.5 V
Botón presionado	3.0 V
Conectado	1.5 V

Tabla 6 – Estados indicados por PP

Otra función importante del pin es comunicar al vehículo la corriente admisible del cable. En el caso de los conectores que ya vienen con cable incorporado, el proveedor se encarga de colocar una resistencia entre PP y PE. En el caso de que el cable sea armado (como el caso del proyecto) se debe insertar dicha resistencia, en función de la sección de cable utilizada. En ese caso, $330\ \Omega$ indican 32A y es lineal para la corriente que se desee.

En cuanto al tipo de conector a utilizar para la conexión con el vehículo, en Uruguay está establecido por normativa la utilización del conector Tipo 2 (también conocido como Mennekes), el cual cuenta con 7 pines:

- Tres fases y neutro (L1, L2, L3, N)
- Tierra (PE)
- Control Pilot (CP)
- Proximity Pilot (PP)

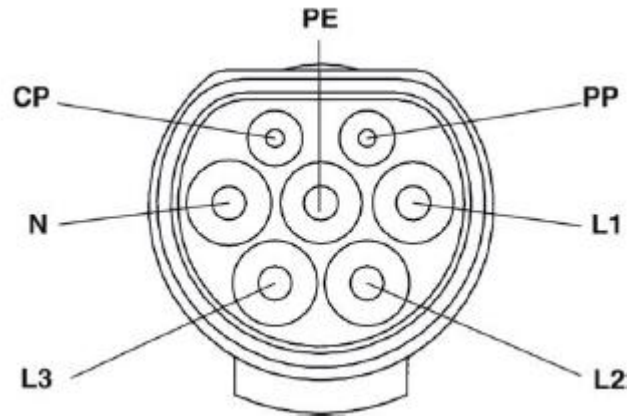


Ilustración 4 – Esquema conector tipo 2 [15]

2.1.4. Flujo de carga típico

Dentro de la carga de un vehículo eléctrico podemos distinguir tres actores:

- Sistema central de carga
 - Puede ser gestionado por el Operador de la red de carga o el Operador de la red eléctrica.
 - Debe ser el que autorice la carga de un vehículo en función de la identificación del usuario y de la corriente disponible.
- Punto de carga
 - Es el punto de interconexión específico para el vehículo eléctrico.
 - Su cometido principal es dar seguridad eléctrica al proceso de carga.
 - Canaliza la comunicación entre vehículo y sistema central.
- Vehículo eléctrico
 - Dentro del mismo, se encuentra el cargador de las baterías.

A grandes rasgos, el cargador que está dentro del vehículo es el que regula eléctricamente todo el proceso de carga para optimizar el flujo de energía y alargar la vida útil de las baterías. La curva de carga de las baterías presenta habitualmente una zona de corriente constante donde se da la carga rápida de la batería hasta un cierto nivel, luego del cual se continuará a voltaje constante hasta llegar al 100% de la carga.

En este proceso, ni el punto de carga ni el sistema central tienen participación más allá de chequear que se está consumiendo dentro de los parámetros preestablecidos. Es decir, la regulación eléctrica para optimizar la carga de las baterías la realiza puramente el cargador a bordo del vehículo. El punto de carga y sistema central, solo se encargará de validar usuarios y cerrar la transacción asociada a la sesión de carga.

Al conectar un vehículo, a través del PWM de la señal del CP el vehículo indicará la potencia de carga (corriente) solicitada. El sistema central, mediante el punto de carga, aprobará dicho consumo o indicará el valor de corriente que dispone, para que el vehículo vuelva a negociar con el valor de corriente especificado. Mediante esta negociación, se asegura que el vehículo jamás solicite más potencia de la disponible en dicho punto de carga. En este proceso, el vehículo también chequea la corriente admisible del cable indicada mediante la resistencia entre PP y PE.

Luego de definida la corriente de carga, el usuario deberá identificarse en el punto de carga y, en caso de estar registrado, comenzará la carga. Este proceso forma parte del protocolo OCPP. La sesión de carga continuará hasta que el mismo usuario vuelva a identificarse para solicitar el corte de la alimentación o llegue una orden externa desde el sistema central de UTE.

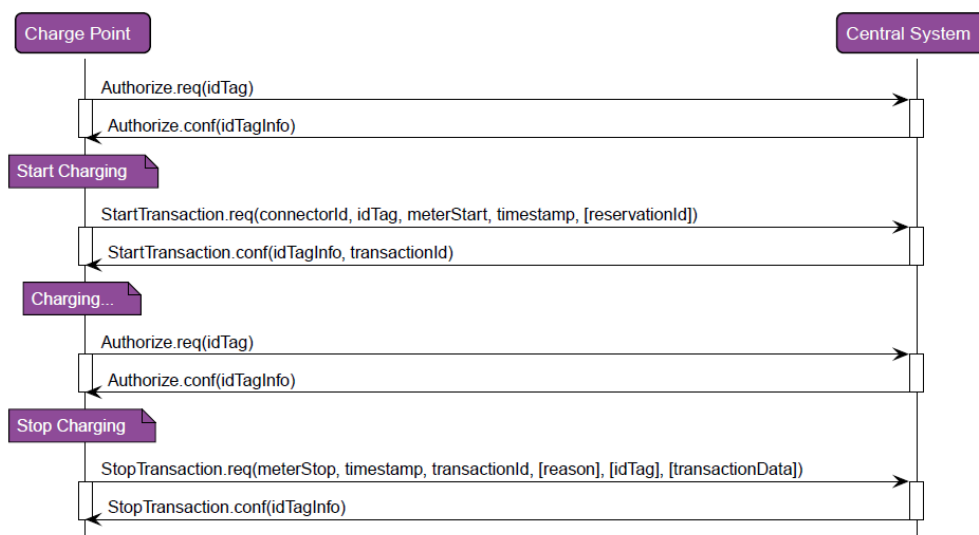


Ilustración 5 - Esquema de carga en OCPP 1.6

En este esquema, no se contempla las nuevas funcionalidades que se comenzaron a implementar en OCPP 1.6 y que continúan en desarrollo en OCPP 2.0 referidas al smart charging. En las nuevas versiones del protocolo, no solamente habría una instancia de identificación, sino que la velocidad de carga estaría dada por el estado de la red en ese momento, según un perfil de carga previamente establecido por el sistema central. La forma de facturar se explica en el capítulo siguiente.

2.1.5. Normativa en Uruguay

En Uruguay, UTE no establece restricciones en cuanto al modo de carga de los vehículos eléctricos. Permite los modos 1, 2 y 3 indistintamente, aunque se encuentra en proceso de desarrollo de su red de carga, instalando puntos en la vía pública. La única restricción está indicada para el conector en el vehículo, el cual debe ser Tipo 2. Tampoco hay una reglamentación específica para indicar si el punto de carga debe poseer cable para conexión con el vehículo o puede simplemente tener un conector, implicando en este caso que para la conexión del vehículo se deba contar con el cable. En ese sentido, también se está tendido a incluir puntos de carga con cable, aunque por ahora se encuentran indistintamente de los dos tipos dentro los puntos de carga ya desplegados por el país.

Para llevar adelante una carga en un lugar público, primero hay que registrarse como usuario ante la UTE de forma de obtener la tarjeta de identificación. Dicha tarjeta opera en la modalidad pospago, asociando la misma a una cuenta de UTE residencial o comercial de forma de recibir mensualmente el monto asociado a la carga de vehículos eléctricos. La integración con el equipo desarrollado por SimpleTech permitiría incluir la modalidad pre-pago ya que se encuentra asociado a métodos de pago electrónicos. Está establecido en el pliego tarifario los costos relacionados a movilidad eléctrica, diferenciándolos en tres franjas (valle, llano y punta - ver [16]). En ese sentido, el servicio ofrecido en la actualidad no tiene ninguna flexibilidad y es similar a cualquier consumo: dada la energía consumida y horario, el monto a pagar.

La potencia máxima disponible está dada por cada punto de carga (en el mapa de UTE se puede ver cuál es la potencia disponible en cada lugar). Más allá de la gestión interna que realice el auto, UTE solamente puede autorizar o cortar la carga, pero cuenta con la infraestructura necesaria para realizar una gestión inteligente de la misma. El pago se realiza en función de la energía consumida (Wh).

2.1.6. Blockchain

Blockchain [17] se puede definir como un sistema de registro y procesamiento de transacciones basado en la descentralización, la transparencia y la seguridad de los datos. Estos registros son replicados entre muchas partes diferentes, se pueden almacenar y rastrear, pero luego de introducido no podrá ser modificado o eliminado. De esta manera se asegura que una vez que se realiza una transacción, esta quede registrada generando copias fieles y ordenadas para cada actor de la red.

Si bien existen diferentes arquitecturas de red blockchain, en términos generales se destacan cuatro actores principales:

- nodos: equipos que conforman el backbone de una red blockchain que valida transacciones y almacena y distribuye la información en tiempo real entre los distintos participantes. Estas funciones suelen subdividirse en diferentes tipos de nodos.
- transacciones: Llamada a procedimiento remoto firmado que autoriza a realizar alguna acción dentro de la red.
- bloques: conjunto de transacciones y registros que junto con información criptográfica y metadata representan la información que es almacenada en una blockchain. Cada bloque guarda un vínculo criptográfico con el anterior conformando la cadena de bloques.
- Contratos inteligentes: Scripts o registros que contienen la lógica de aplicación y que determinan las acciones que se ejecutan cuando se procesa una transacción.

Dependiendo del uso que se le dé a una red blockchain, se puede categorizar de diferentes formas. Un tipo de categorización refiere a la relación que hay entre los participantes, se puede dividir entre una blockchain pública y una privada. Una blockchain pública se encuentra abierta y cualquiera puede crear un nodo y procesar transacciones, así como también cualquiera puede participar enviando transacciones para que sean procesadas. Mientras que una blockchain privada pertenece a una organización donde todos los participantes están identificados y deben ser autorizados por la organización para enviar transacciones y para participar en el procesamiento de estas.

Otro tipo de categorización es la de blockchain permissionada o no permissionada. Estas refieren a los permisos de escritura, lectura y validación de los datos. Una blockchain permissionada es gobernada generalmente por un consorcio de entidades o empresas que definen las reglas de participación en la red y de interacción con la misma, otorgando permisos de lectura, escritura y verificación de transacciones a los participantes. Sin embargo, la validez de las transacciones se realiza mediante mecanismos de consenso y en caso de ser necesario, se puede ocultar información sensible mediante algoritmos de cifrado. Una blockchain no permissionada permite a cualquier participante a realizar acciones sin restricciones.

De manera general, la blockchain de consorcio permissionada es la que se adapta mejor a la mayoría de los casos de uso blockchain para empresas, ya que los participantes que procesan transacciones están debidamente identificados y se desea tener control sobre las aplicaciones que se despliegan en la plataforma.

2.2.Cargador OpenEVSE

2.2.1. Componentes

El análisis parte de un kit fabricado por OpenEVSE, el cual tiene la particularidad de tener código abierto, tanto para el hardware como el software, y por este motivo, ofrece gran flexibilidad para desarrollar funcionalidades sobre él. La primera tarea realizada fue identificar sus componentes.

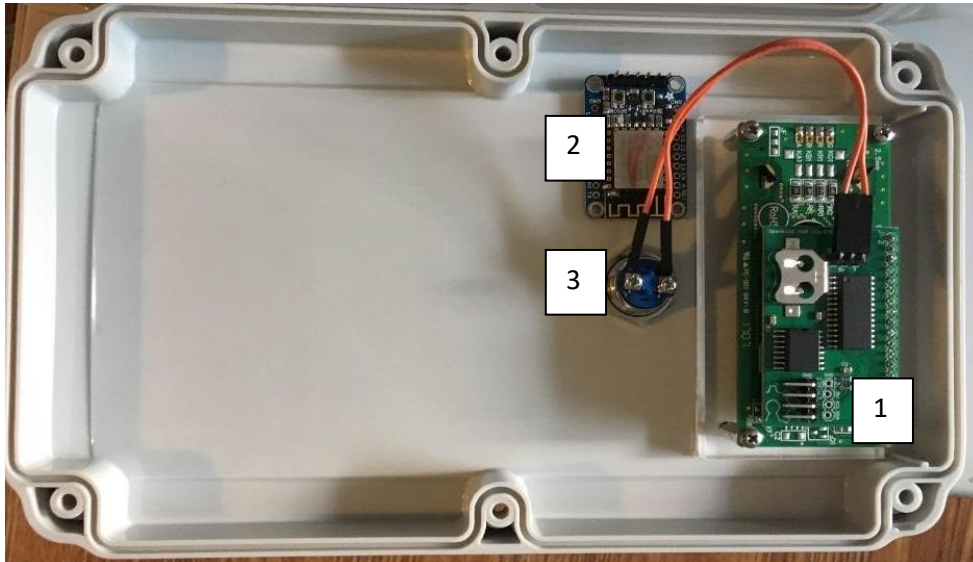


Ilustración 6 – Componentes del OpenEVSE (tapa)

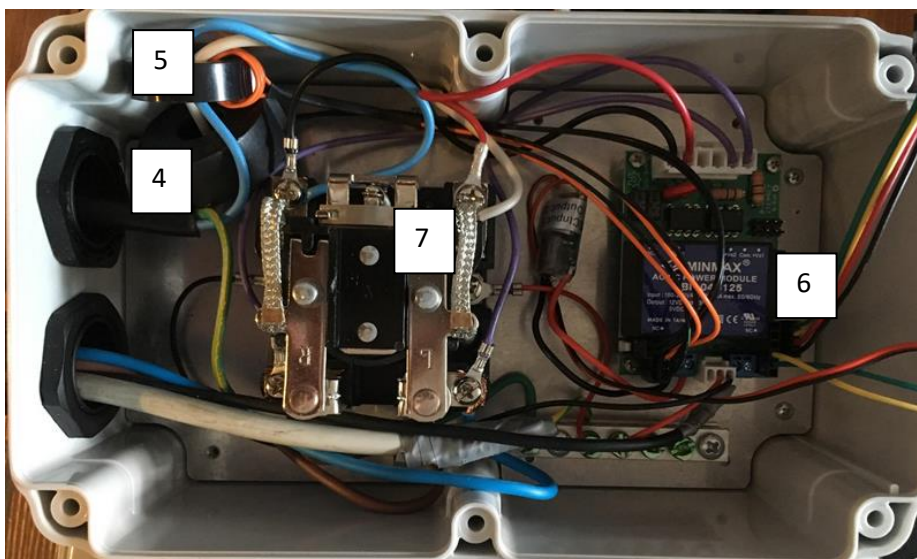


Ilustración 7 – Componentes del OpenEVSE ensamblado (interior)

A grandes rasgos, el kit está compuesto por:

1. Display

Display con capacidad para dos líneas alfanuméricas de hasta 16 caracteres cada una.

2. Módulo WiFi ESP 8266

El mismo incluye un microcontrolador, que permite la operación del OpenEVSE utilizando WiFi.

3. Botonera

El OpenEVSE incluye un botón mediante el cual se pueden setear y configurar opciones de carga.

4. Transformador para medición corriente de fase (CT)

Transformador para medición de corriente por las fases del cargador, capacidad hasta 100A. Dado que el equipo es monofásico, es suficiente con un transformador.

5. Transformador de corriente para fugas a tierras (GFCI)

Transformador de corriente para medición de las fugas a tierra. Se enhebran todos los conductores a través de este, teniendo una sensibilidad de 20mA.

6. Controlador OpenEVSE plus V5

Placa desarrollada por OpenEVSE, para control del proceso de carga e implementación de IEC 62169 para comunicación entre vehículo y punto de carga. Incluye una fuente de 12Vdc.

7. Contactor monofásico

Corriente nominal de 50A y bobina de 12Vdc. La tensión en bornes de la bobina está comandada por el controlador OpenEVSE.

Adicionalmente al kit OpenEVSE, se realizó la compra de un conector Tipo 2, para la conexión a vehículos. Al mismo, se le agregó una resistencia entre PP y PE de 200Ω. De acuerdo con lo indicado en el capítulo 2.1.3, con dicha resistencia se estará indicando el vehículo eléctrico que la corriente admisible del cable es de 19.4A.

$$I_{ADMISIBLE} = \frac{200\Omega}{330\Omega} \times 32A = 19.4 A$$

De esta forma, el vehículo no solicitará más corriente que la disponible en el punto de carga (indicada a través del CP) o la admisible por el cable (indicada a través del PP). Así, queda protegido el cable de 2.5mm² utilizado en nuestro conector. Esto fue probado con un vehículo, según se resume en el capítulo 5. De acuerdo con la normativa de UTE (tabla XI del capítulo 2 del Reglamento Baja Tensión), la corriente admisible para esa sección es 28A.

El armado del equipo se realizó siguiendo el diagrama de conexionado proporcionado por OpenEVSE [18].

Por otro lado, para que sea viable habilitar la energía de carga a la salida del contactor, el controlador OpenEVSE debe contar con información en el pin CP. En caso de dejarlo libre estaremos en el nivel 12V, valor usado para indicar que no hay ningún vehículo conectado (ver Tabla 4). Para obtener un valor diferente de 12V a la hora de realizar las pruebas, se utiliza un simulador propiedad de UTE, el cual no es más que una configuración de resistencias y llaves ON/OFF utilizadas para modificar la resistencia vista a tierra, de forma de simular los valores previstos en el estándar IEC 62169. El esquemático del circuito es el siguiente:

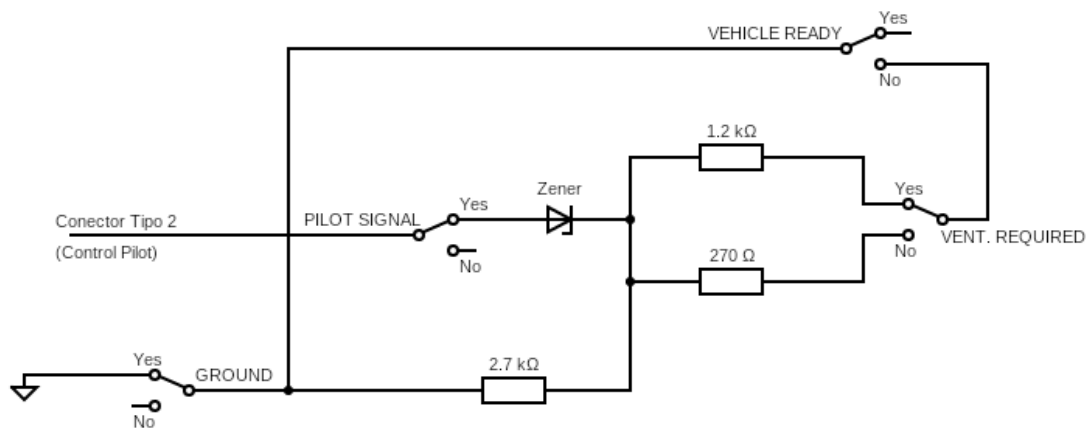


Ilustración 8 – Circuito para simulación comunicación Control Pilot

2.2.2. Prueba de funcionamiento kit OpenEVSE

El primer punto para desarrollar con la llegada del equipo fue el cableado tanto para la alimentación desde la red eléctrica como el cable para el conector Tipo 2. Se utilizó conductores de 2.5mm² tanto para la alimentación a la red eléctrica como para el conector, dado que era la máxima sección que se podía conectar a los componentes. Eléctricamente, sería necesario la utilización de una sección mayor, según se describe posteriormente.

Se utilizan cargas domésticas conectadas al simulador de UTE para las pruebas de consumo. Esto facilita la conexión a la carga, ya que no se dispone de un vehículo con el terminal tipo 2 macho. Con este fin, se conecta a la salida del contactor una toma Schucko.

Prueba de carga sin simulador de Control Pilot

Se prueba energizar el OpenEVSE, configurar desde la botonera y alimentar un ventilador de 1600W. Dado que el controlador OpenEVSE espera información del auto a la entrada del CP y este no se encuentra conectado, jamás da la orden de cerrar el contactor para alimentar la carga a la salida. Esto está acorde a lo especificado en el IEC 62169, ya que el valor 12 V indica que no hay vehículo conectado (ver Tabla 4) y, por lo tanto, no se envía la señal para cerrar el contactor.

Prueba de carga con simulador de Control Pilot

Se repitió la prueba anterior, pero esta vez se conecta el simulador al pin CP del OpenEVSE. Se registran los siguientes estados mostrados en el display, en función de la posición de las llaves del simulador

Vehicle Ready	Ventilation Required	Control Pilot signal	GND	Display OpenEVSE	Estado de la carga
0	0	0	0	Connected	EV conectado
0	0	0	1	Ready	Error
0	0	1	0	Ready	Error
1	0	0	0	Charging	EV cargando
1	0	0	1	Ready*	Error
1	0	1	0	Ready*	Error
1	1	0	0	Vent. Req.**	Ventilación requerida

Tabla 7 – Pruebas de funcionamiento IEC 62169

Para corroborar estos datos se realizaron medidas con un osciloscopio simulando los estados de la Tabla 7 y observando el voltaje de salida en el CP buscando verificar que el controlador OpenEVSE efectivamente cumple con las especificaciones IEC 62169. Las imágenes que se muestran a continuación ejemplifican el caso de un duty cycle de 40%, lo que está asociado a una corriente máxima de 24 Ampere.

En primer lugar, se mide el nivel de la señal simulando que no hubiera una carga conectada (CP desconectada). Efectivamente se obtiene el valor esperado que es de 12V para describir el *Estado A - No conectado*.

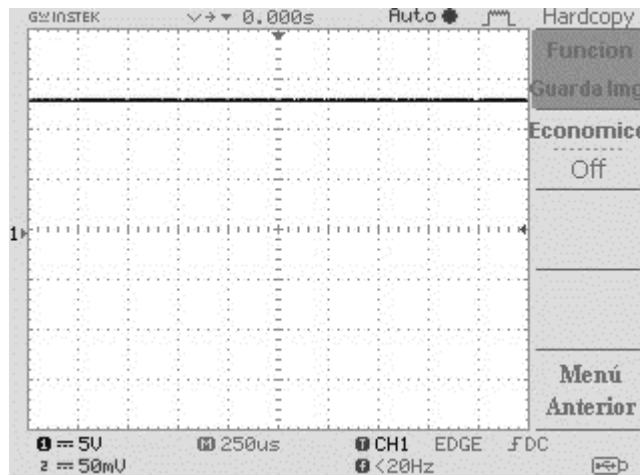


Ilustración 9 – Señal vista en CP desconectado

Luego, se conecta la CP, al medir la señal se observa el valor de la señal en 9V con un *duty cycle* del 40%. Estos valores corresponden al *Estado B - EV conectado*.

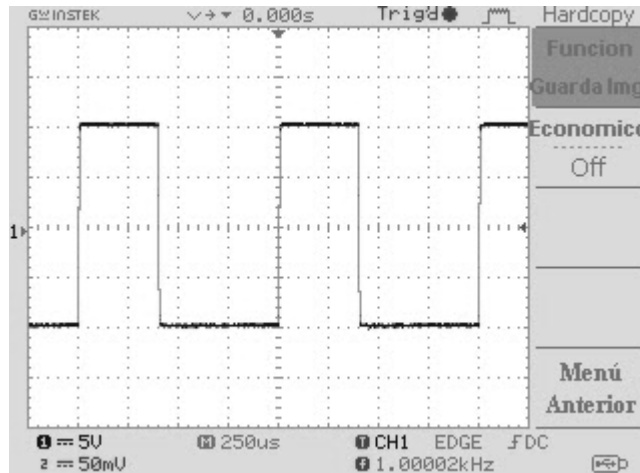


Ilustración 10 – Señal en CP simulando EV conectado

Con la CP conectada, se activa la llave en el simulador de *Vehicle Ready* y al medir la señal se obtiene el valor de 6V, este valor representa al *Estado C - EV cargando*.

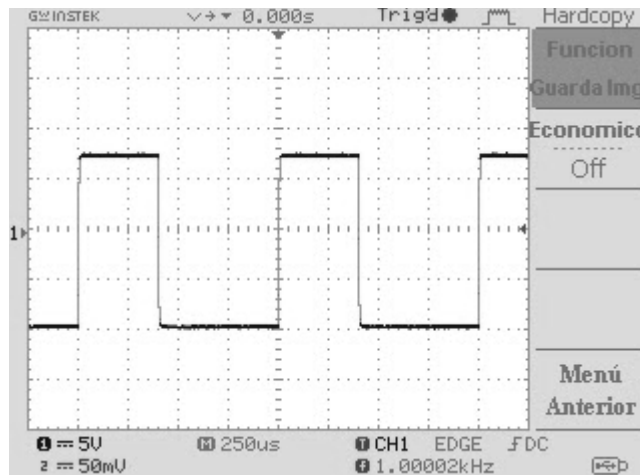


Ilustración 11 – Señal en CP simulando carga

Sobre el escenario anterior se activa la llave que simula la falla de *Ventilation Required* y se obtiene el valor de 3V, representa al *Estado D - Ventilación requerida*.

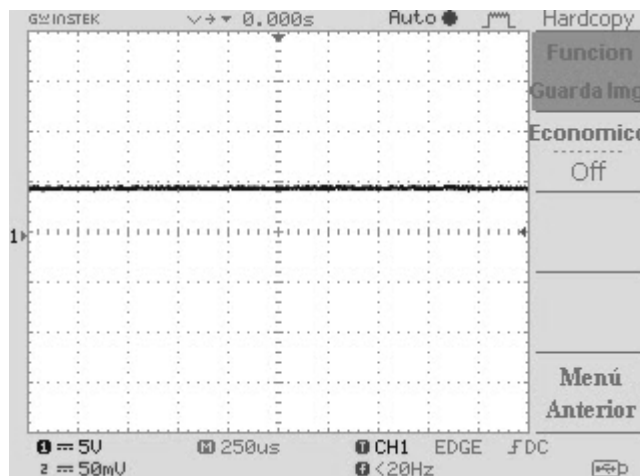


Ilustración 12 – Señal en CP simulando Ventilación requerida

Se vuelve el cargador estado C y se activa alguna de las otras dos fallas existentes en el simulador, pudiendo ser falla en la CP o falla en el chequeo de tierra. Al activar estas fallas, la medida en el osciloscopio muestra la medida de -12V, siendo este el *Estado F - Error*.

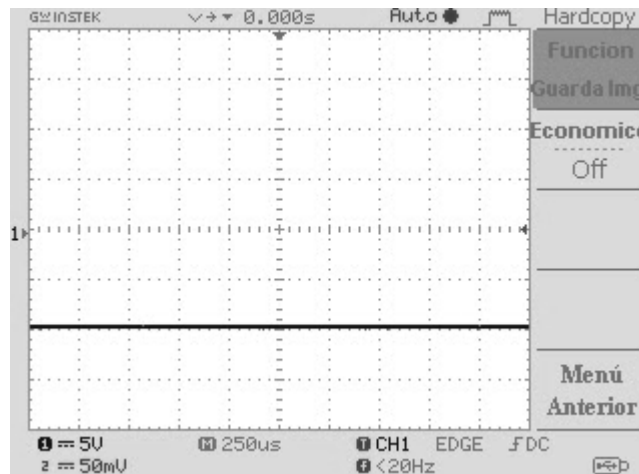


Ilustración 13 – Señal en CP simulando una falla

Se realizaron medidas para diferentes valores de corriente de alimentación a la carga (ver Anexo 10.1) verificando que el *duty cycle* y forma de onda son correctas. De esta forma se da por válido la implementación del protocolo IEC 62169 en el controlador OpenEVSE, por lo cual no es necesario realizar ningún desarrollo adicional para la comunicación vehículo-punto de carga

2.3. Identificación de mejoras a implementar

Para el desarrollo del proyecto, se identificaron una serie de mejoras para cumplir con los requerimientos de UTE y el alcance planteado para el trabajo.

- Para lograr el control sobre el cargador de forma de poder gestionar la carga con la plataforma CargaME de UTE mediante el protocolo OCPP se requiere de mayor capacidad de cómputo que la que tiene la placa OpenEVSE. Adicionalmente, el controlador del OpenEVSE cuenta con un firmware custom, lo cual dificulta el desarrollo y la integración de paquetes y librerías de software necesarias. Se decide entonces utilizar la Raspberry Pi para que desde esta se establezca el control del SAVE. Dentro del estudio de la viabilidad de poder realizar transacciones blockchain, también se concluye que el OpenEVSE no cuenta con capacidad de cómputo suficiente. Por lo que se buscará la implementación de blockchain a través de la Raspberry Pi.
- Es requerimiento en los pliegos para puntos de carga de vehículos eléctricos de UTE que la identificación se realice mediante RFID. Es por esto que se decide incorporar la placa de identificación RFID como periférico a la Raspberry Pi para lograr la identificación del usuario final y para eventualmente facilitar modalidades de pre-pago.
- Tanto para blockchain como para establecer la comunicación con la plataforma CargaME se requiere de conectividad a internet. Este punto se resuelve con la incorporación de un módem 3G. UTE provee de una tarjeta SIM con IP privada dentro de su red corporativa para hacer posible la conexión con la plataforma CargaME de desarrollo.
- Por otro lado, tanto para transformar el punto de carga en trifásico 22kW como para lograr la protección contra contactos indirectos y protección del cable, es necesario incorporar los siguientes elementos:
 - Termomagnética, 4 polos, corriente nominal 32A, curva C.
 - Interruptor diferencial, 4 polos, sensibilidad 30mA, corriente nominal 32A.

- Contactor trifásico, 4 polos, corriente nominal 32A y bobina 12Vdc.
- Dos transformadores de intensidad 1000:1 (adicional al ya existente en el equipo OpenEVSE).

Con estas modificaciones se cumpliría con las normativas de UTE para instalaciones de baja tensión.

Para el cable de alimentación, se supone una instalación enterrada, con cable tetrapolar asilado en polietileno. Según la normativa vigente de UTE (tabla XV del anexo al Capítulo 2 del reglamento de baja tensión de UTE [19]), realizando dimensionamiento por corriente admisible se debería utilizar 6mm².

El plano de cableado del punto de carga propuesto se encuentra en anexos (ver Anexo 10.2).

3. Arquitectura propuesta

Observando las necesidades de control, identificación, conexión por internet para el gestor OCPP y para las transacciones blockchain es que se desarrolla la siguiente arquitectura para el cargador.

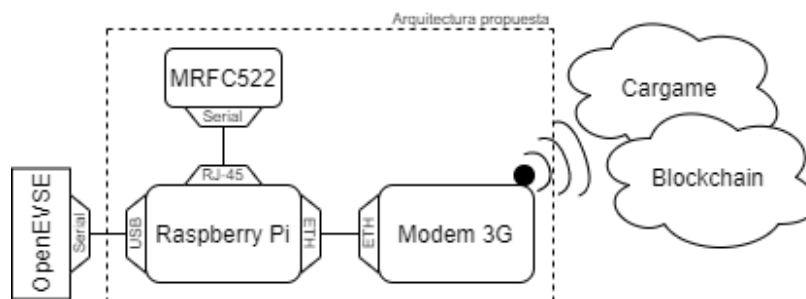


Ilustración 14 – Arquitectura propuesta

El diseño final se centra en colocar la Raspberry Pi como cerebro del cargador, para que oficie de intermediario entre las distintas componentes manteniendo varios hilos de comunicación activos simultáneamente. En el desarrollo del proyecto se utilizó una Raspberry Pi que fue brindada por SimpleTECH, como parte de su equipo utilizado como *smart vending machine*.

A continuación, se describen las conexiones creadas, sus funcionalidades y posibles variaciones analizadas.

3.1. Identificación RFID

Para la identificación del usuario mediante RFID se utiliza una solución provista por SimpleTECH que consta de una placa RFID MRFC522 [20] conectada mediante serial a un puerto RJ-45 agregado a la Raspberry Pi. Esta placa RFID trabaja a una frecuencia de 13,56MHz, la que se denomina de alta frecuencia (HF) y se encuentra dentro del espectro que no necesita licencia. El usuario deberá contar con una tarjeta de identificación en la frecuencia antes mencionada que, al pasar por el lector, refleje el número de identificación que será interpretado como texto plano.

Se optó por esta solución dado que es la utilizada por UTE en este momento. UTE cedió una tarjeta RFID autorizada para poder realizar las pruebas de autenticación contra el sistema central.

3.2. Display

Para el despliegue de información se utiliza el display del dispositivo original. Este cuenta con información seteada para desplegar por defecto y además permite mostrar mensajes personalizados lo cual es suficiente para el alcance de este proyecto.

3.3. Conexión con el vehículo

En esta parte hay dos conexiones que son importantes mencionar. Por un lado, la comunicación entre el vehículo y el OpenEVSE (descrita en el capítulo 2.1.3) y por otro lado la comunicación entre la Raspberry Pi y el OpenEVSE. Esta última se realiza a través de una Remote API facilitada por OpenEVSE la cual se puede ejecutar mediante la conexión al encabezado de seis pines de la placa [21]. Se utiliza en este prototipo una conexión USB-serial mediante la cual se ejecutarán los comandos RAPI para la comunicación. En Anexo 10.1 se encuentran los comandos que se pueden utilizar para obtener o grabar información.

A los efectos del alcance deseado en este proyecto se utiliza una arquitectura de una Raspberry Pi y periféricos por cada cargador, por lo cual la conexión USB-serial es viable, Pero existe la posibilidad de querer elevar el nivel y la eficiencia del producto final y tener una arquitectura determinada por un Raspberry Pi y periféricos por N cantidad de cargadores. Esta arquitectura aplicaría, por ejemplo, para el caso de tener una playa de cargadores donde el usuario se pueda identificar y la Raspberry Pi asignarle un cargador libre.

Para el caso planteado la conexión física de USB-serial no sería adecuado, por lo que se deja planteado para la continuidad del proyecto la posibilidad de implementación de conexión entre Raspberry Pi y OpenEVSE utilizando módulos RS-485.

RS-485 [22] es un estándar industrial de capa física del Modelo OSI, por lo que no habría que modificar el desarrollo realizado para la comunicación dado que no altera la forma en que se envía la información. Utilizando esta conexión en capa física, es posible implementar Modbus, por lo que se considera como un sistema de comunicación en bus diferencial multipunto. Tiene una buena tasa de transmisión para las necesidades del proyecto dado que hasta los 10m de distancia puede transmitir a una velocidad de 35 Mbps y puede llegar a transmitir 100 kbps en 1200m. El medio físico es un par de cobre trenzado y admite 32 estaciones en 1 solo bus. La transmisión se puede dar sin problemas a través de canales ruidosos dado que emplea voltajes diferenciales, es decir, mientras que un cable transmite la señal original, el otro transmite su inversa. La conexión RS-485 define una arquitectura maestro-esclavo o maestro-múltiple esclavo en la que el maestro controla a todos los dispositivos esclavos conectado en el mismo bus. Para el caso planteado de una Raspberry Pi y N OpenEVSE se utilizará la arquitectura maestro-múltiple esclavo siendo la Raspberry Pi el maestro y los OpenEVSE los esclavos.

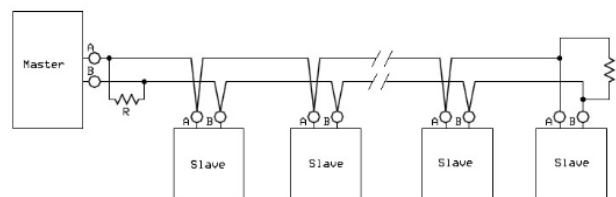


Ilustración 15 – Arquitectura maestro esclavo

Por otro lado, la conexión USB-serial debe ser modificada dado que su velocidad de respuesta es muy lenta, lo que resultaría en una experiencia de usuario débil pudiendo generar confusiones.

3.4. Blockchain

A los efectos de lograr el alcance del proyecto, se buscará implementar una solución blockchain utilizando Hyperledger Fabric a bajo nivel, con el fin de comprobar la viabilidad de que la Raspberry Pi se parte de la red como un “light client” y pueda invocar un chaincode (contrato inteligente o *smart contract* en Hyperledger Fabric). Sin embargo, cabe destacar que, si bien UTE ha estado experimentando con la tecnología blockchain utilizando la plataforma Hyperledger Fabric, aún no ha realizado ninguna prueba para el sector de movilidad eléctrica y tampoco han definido qué función cumpliría cada cargador en la red.

Hyperledger Fabric [23] es una plataforma para soluciones contables distribuidas de código abierto, diseñada para su uso en contextos empresariales. Ofrece algunas capacidades de diferenciación clave sobre otras plataformas de blockchain populares. Se basa en una arquitectura modular que ofrece altos grados de confidencialidad, resistencia, flexibilidad y escalabilidad, adaptándose a las complejidades que existen en todo el ecosistema económico.

Un punto clave de diferenciación es que Hyperledger se estableció bajo la Fundación Linux, que tiene una larga y exitosa historia de fomentar proyectos de código abierto bajo una gobernanza abierta que hacen crecer comunidades sólidas y sostenibles y ecosistemas prósperos. Hyperledger está gobernado por un comité directivo técnico diverso, y el proyecto Hyperledger Fabric por un conjunto diverso de múltiples organizaciones.

Tiene una arquitectura modular y configurable, que permite la innovación, la versatilidad y la optimización para una amplia gama de casos de uso de la industria, incluidos la banca, las finanzas, los seguros, la atención médica, los recursos humanos, la cadena de suministro e incluso la entrega de música digital.

Es la primera plataforma de contabilidad distribuida que admite contratos inteligentes creados en lenguajes de programación de uso general como Java, Go y Node.js. Esto significa que la mayoría de las empresas ya tienen los conocimientos necesarios para desarrollar contratos inteligentes, y no necesitan de capacitación adicional para aprender un nuevo idioma o DSL.

La plataforma Fabric es permissionada, lo que significa que los participantes se conocen entre sí. Esto significa que si bien los participantes pueden no confiar completamente el uno en el otro (pueden ser, por ejemplo, competidores en la misma industria), una red puede funcionar bajo un modelo de gobernanza que se basa en la confianza que existe entre los participantes, como un acuerdo legal o marco para el manejo de disputas.

La combinación de estas características de diseño diferenciales hace que Fabric sea una de las plataformas de mejor rendimiento disponible en la actualidad tanto en términos de procesamiento de transacciones como de latencia de confirmación de transacciones, y permite la privacidad y confidencialidad de las transacciones y los contratos inteligentes que los implementa.

Esta posibilidad de crear *chaincodes* propios de forma que cumplan con los requisitos de la red, tiene tokens pre-minados y una característica que hace que esta plataforma sea atractiva es la capacidad de separar las transacciones por canales. De esta forma, se puede crear un canal para cada sector de UTE, o por cada funcionalidad que se le quiera dar a la blockchain, pudiendo ser pagos mediante blockchain, trazabilidad o almacenamiento de documentos.

Actualmente UTE se encuentra analizando cómo seguir luego de realizar procesos experimentales con Hyperledger Fabric y de llevar adelante un proceso de integración a la Energy Web Foundation (EWF). La EWF es una organización global sin fines de lucro cuyo principal objetivo es acelerar la transición mundial de un sistema energético descentralizado, democratizado, descarbonizado y digitalizado. Esta organización funda en 2017 una plataforma

blockchain con base en Ethereum llamada Energy Web Chain (EW Chain), y es lanzada como una blockchain especializada en energía en enero del 2017.

La EW Chain es una blockchain de código abierto basada en Ethereum. Está diseñada especialmente para ser aplicada al sector energético, utilizando un mecanismo de consenso de PoA (Proof of Authority) para de esta forma aumentar significativamente la capacidad de transacción (treinta veces más) y disminuir el consumo de energía, utilizando light clients, en comparación con la red principal de Ethereum.

Los criterios clave para un mecanismo de consenso compatible con el sector energético son:

- alta capacidad
- seguridad
- eficiencia de recursos
- regulabilidad
- fidelidad

Un grupo de computadoras conocidas y confiables, llamadas nodos de validación, son responsables de validar las transacciones y crear los bloques. De esta forma se obtiene seguridad, transparencia, beneficios en cuanto a la capacidad, pero se logra que la red no esté descentralizada. Dado que la blockchain sería controlada al menos en un principio por UTE y puesta a disposición para sus clientes, la falta de descentralización no tiene mayor importancia.

El light client almacena una versión "light" de toda la cadena de bloques en dispositivos de baja potencia (desde teléfonos celulares hasta medidores inteligentes que tengan acceso completo a la red) y proporciona un protocolo para solicitar datos adicionales de la red según sea necesario a los nodos validadores. La información almacenada en los light clients incluye encabezados de bloque y variables de estado relevantes, lo que permite un nivel similar de conocimiento y usabilidad como nodos completos, mientras que sólo almacena una pequeña fracción de los datos de blockchain.

EW Chain cuenta con un token propio llamado Energy Web Token (EWT), y tiene como propósito general el de brindar seguridad y recompensar al validador. Se han preminado 90 millones de EWT y se espera liberar 10 millones más durante diez años en recompensa de validación de bloque.

El token protege la red contra el mal comportamiento (intencional o no) de la ejecución de la transacción (por ejemplo, bucles infinitos) o comportamiento indeseable. El mecanismo utilizado es similar al Ethereum. Las cuentas que envían una transacción a la cadena EW pagan un costo de transacción basado en el esfuerzo computacional de ejecutar la transacción. Ese esfuerzo es estimado por un proxy de esfuerzo computacional llamado "gas". La estimación de la tarifa de gas de la cadena EW es la misma que la red principal de Ethereum.

Se recompensa a los validadores por los costos de ejecutar un nodo. Las recompensas son por fees de transacción y por validación de bloques.

Cuenta con módulos SDK (software development kit) de código abierto que pueden copiarse, modificarse o simplemente usarse como referencia para desarrollar aplicaciones comerciales.

A los efectos del alcance del proyecto se realizará el estudio sobre la base de una Raspberry Pi funcionando como un light client dentro de una red blockchain [24]. Un light client es capaz de ejecutar funciones previamente definidas para interactuar con una red blockchain a través del nodo cuando se requiera, es decir, no necesita estar activo todo el tiempo. No necesita mucho espacio de memoria dado que no se descarga los bloques enteros, sino que solo el encabezado. El encabezado contiene el Merkle Tree de todos los bloques. Esta información es

una huella de toda la información en la red blockchain, si una parte por más ínfima que sea de la red cambia entonces también lo hará esta huella. Esta huella le permite interactuar con la red sin necesidad de confiar completamente en el nodo dado que puede verificar si hubo alteraciones no deseables en la red. Un light client requiere únicamente de 100MB de almacenamiento y una capacidad de cómputo baja.

Dado que UTE aún se encuentra en etapas de pruebas experimentales respecto a blockchain, para el alcance de este proyecto alcanza con corroborar que la Raspberry Pi puede ejecutar transacciones blockchain.

La siguiente arquitectura representa una idealización de un posible caso de uso para UTE, donde ellos ejercen el rol de administradores de la red y otros participantes forman parte de a red como organizaciones que gestionan múltiples redes de recarga.

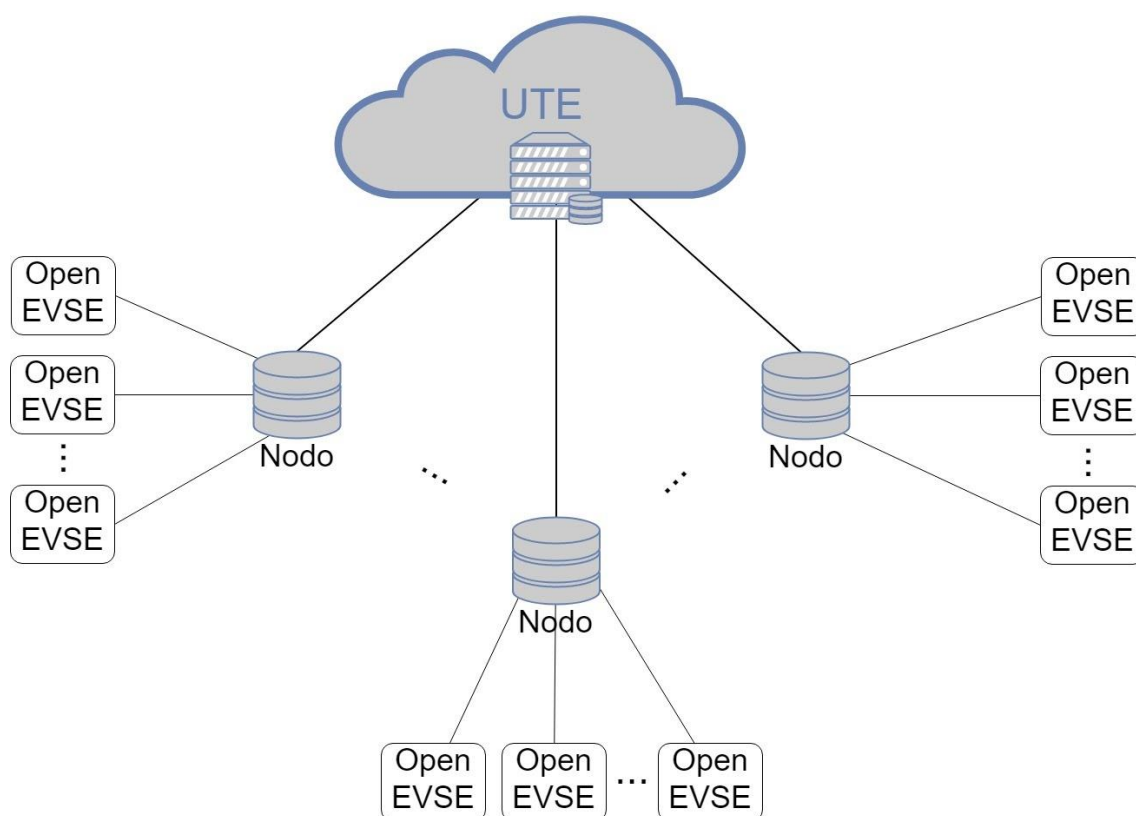


Ilustración 16 – Estructura red blockchain

Para lograr desplegar una red blockchain donde UTE no sea el único dueño de todos los nodos, se podría buscar una oportunidad de negocio en la que UTE venda o financie el cargador OpenEVSE a entes/empresas/interesados y junto con estos se incluye la instalación de un nodo de la red y así ingresar como una organización a la red. La idea sería que, por ejemplo, la IMM, compre más de un cargador, pero todos ellos tengan como punto de contacto con la red blockchain al mismo Nodo. De esta forma, se cobra al que hizo uso del cargador que se identificó mediante RFID y se puede beneficiar a la organización con incentivos acordes a la cantidad de “servicio” que haya vendido.

Para lograr esto los nodos serán los encargados de simular y firmar las transacciones y una vez validado, se envía al ordering que va a ser de propiedad de UTE.

Para realizar un caso de uso de Hyperledger Fabric hay que crear un entorno que permita crear y desarrollar la aplicación. Una red Fabric puede ser creada de varias formas [25]:

- Hyperledger Fabric en red local, esto consiste en descargar localmente los archivos de configuración de las distintas partes de la red y crear la red localmente.
- Utilizando la IBM Blockchain Platform situada en la nube de IBM, esta opción permite que la nube de IBM proporcione BaaS (Blockchain as a Service), es decir, desplegar componentes en diferentes entornos del gusto del desarrollador.
- Utilizando Kubernetes API en Kubernetes Service en la nube de IBM, lo que permite generar la red en la nube de IBM (Nodos, entidad certificadora, smart contracts, etc) y acceder utilizando un cliente SDK conectado desde la web.

3.5.OCPP 1.6

Como fue mencionado anteriormente, se ha llegado a un acuerdo entre proveedores de puntos de carga acerca de los mensajes que deben intercambiar con el sistema central. En ese sentido, hoy en día conviven varias versiones del protocolo que son incompatibles entre sí. A pesar de que no es la última versión, se propone la implementación de OCPP en la versión 1.6. La versión 1.5 ofrece poca flexibilidad y funciones para que un sistema de carga gestione de forma remota el punto de carga, mientras que la versión 2.0 recién se está divulgando en formato de prueba. La comunicación se realiza a través de la salida a internet.

El OpenEVSE no tiene este protocolo por lo que se debe implementar. El desarrollo se describe en el capítulo 4.

3.6.Comunicación 3G

Para la salida a internet con el fin de comunicarse con el CargaME y con la red blockchain se utilizó un modem 3G que utiliza un servicio de UTE con IP privada y fija dentro de la red corporativa del ente lo que permite la conexión con el sistema central.

Actualmente UTE cuenta con conectividad 3G y 4G de Antel tanto para los puntos de carga como para los medidores de energía eléctrica inteligentes que se están instalando. En el caso puntual de movilidad eléctrica es al menos 3G.

3.7.Forma de pago

A futuro se utilizará como sistema de pago un servicio pre-pago ya desarrollado por SimpleTECH el cual cuenta con respaldo en Mercado Pagos. Al realizar la lectura RFID se disparará una consulta para obtener el saldo de la tarjeta identificada, en caso de que la tarjeta cuente con saldo disponible, se procederá a con la identificación en el sistema de carga de UTE para realizar la carga. Esto puede ser utilizado también para el pago del espacio da parking que ocupa el vehículo para su carga.

Posteriormente se podría desarrollar la utilización de una plataforma blockchain para realizar los pagos utilizando tokens obtenidos por los clientes como incentivos digitales transables por realizar actividades consideradas de buena acción y premiadas por UTE o por otra entidad del estado como el municipio.

4. Funcionalidades desarrolladas

Con las conexiones físicas del prototipo establecidas, se plantea el desarrollo en la Raspberry Pi de la lógica del flujo de una carga, desde la energización del punto de carga, la identificación del usuario y la conexión del vehículo hasta la finalización de la transacción. Una primera etapa consiste en la inicialización del punto de carga una vez energizado. Para esto se debe evaluar qué funcionalidades deben quedar operativas a la espera del inicio de carga. Luego, se debe establecer el flujo de carga a seguir y detectar posibles escenarios de interrupciones y su manejo. Finalmente, el proceso mediante el cual se da la finalización de la carga y cierre de la transacción, posteriormente volviendo a un estado de espera.

Para llevar a cabo el desarrollo de la lógica, se parte del esquema de UTE de conexión e inicio de carga para la correcta utilización de sus puntos de carga. Este diagrama puede encontrarse en la página web de UTE [4] y en la Ilustración 17.



Ilustración 17 - Diagrama para ejecutar la carga de un vehículo

De modo general, se desarrolla una sesión de carga completa, incluyendo la conectividad, la implementación de OCPP 1.6 y el manejo de errores. En este último punto, se tomó la decisión de que ante cualquier falla que pueda producirse durante la sesión, se desenergiza el vehículo y queda a la espera de que el usuario se identifique nuevamente para cerrar la transacción y quedar disponible. El objetivo de no liberar el punto de carga hasta que no se produzca dicha identificación es asegurar que se informa el error surgido y que la sesión de carga fue trunca. OCPP y UTE no son explícitos sobre el manejo de las fallas, pero es fácilmente moldeable la respuesta del punto de carga frente al abanico de errores que se quiera tomar en cuenta. El siguiente diagrama representa el flujo desarrollado.

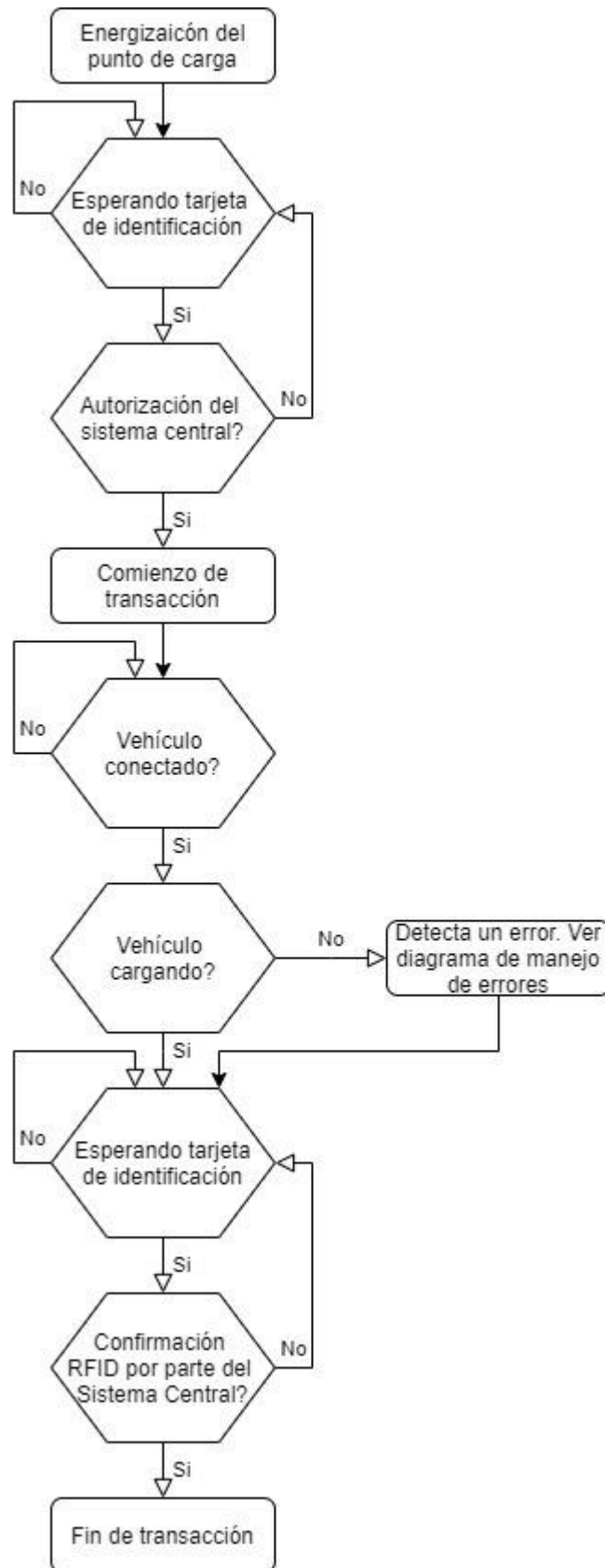


Ilustración 18 – Flujo de una sesión de carga

El siguiente diagrama de flujo representa el proceso de manejo de errores implementado.

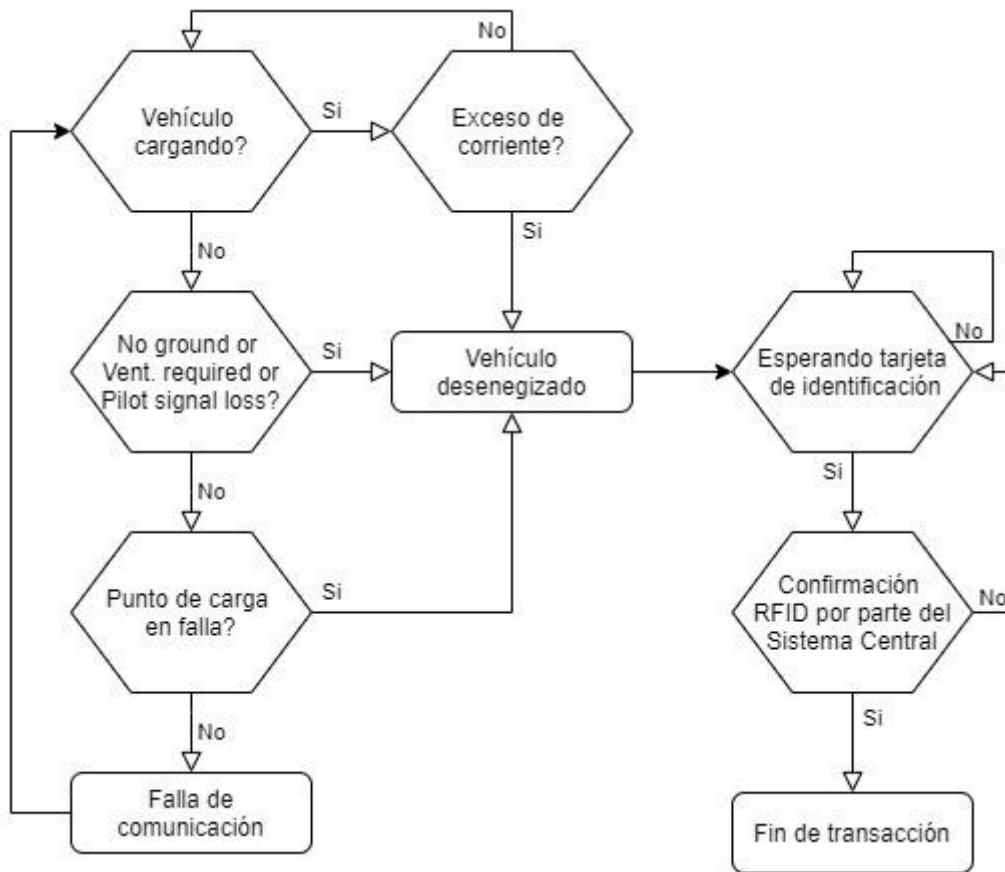


Ilustración 19 – Flujo para resolución de errores durante sesión de carga

El desarrollo de la lógica de funcionamiento del prototipo se realizó en el lenguaje de programación Python. Se utilizó la versión de Python 3.6 en la Raspberry Pi y se utilizaron algunas librerías ya establecidas para el manejo de la conexión entre los dispositivos. La versión de Python fue seleccionada para lograr la compatibilidad con las librerías necesarias para llevar adelante el desarrollo de la lógica. A continuación, se mencionan algunas de ellas con una breve reseña de su utilidad.

Pyserial [26]

Esta librería permite establecer la conexión entre la Raspberry Pi y el OpenEvs que cuenta con entrada serial, se debe especificar el puerto en el cual se desea establecer la conexión en la Raspberry Pi, que en este caso es el usb0 y la velocidad de transmisión (baudrate). La librería detecta el sistema operativo sobre el cual está corriendo y ajusta el backend para poder llevar a cabo la comunicación.

GPio y NFC [27]

Estas librerías son las establecidas por el auspiciante para manejar el módulo acoplado a la Raspberry Pi para la lectura RFID. La ejecución de una de las funciones espera en loop la lectura de una tarjeta RFID, la cual es guardada en una variable quedando disponible para utilizar.

Asyncio [28]

Esta librería permite ejecutar aplicaciones asíncronas facilitando el manejo de distintos hilos para la comunicación de entrada-salida de datos.

Websockets [29]

Esta librería permite la implementación de diálogos cliente-servidor en la cual se pueden construir, analizar y validar solicitudes y respuestas http. Está construida sobre asynco.

Python SDK Fabric [30]

Implementación de Hyperledger Fabric SDK en Python. Permite gestionar credenciales así como información de una red blockchain.

OCPP [31]

Esta librería implementa las funcionalidades de OCPP 1.6 para la versión en formato JSON. Maneja todas las funcionalidades especificadas en la documentación [13] tanto para el punto de carga como para el sistema central. Funciona sobre la librería websockets antes mencionada la cual es complementaria para el intercambio de datos.

En base a estas librerías antes mencionadas es que se realizó el desarrollo de varios scripts con el fin de llevar adelante la lógica requerida para realizar una carga. A su vez se realizó un script que simula el sistema central de UTE para poder llevar adelante pruebas previas a la conexión con un vehículo.

4.1. Scripts desarrollados

El desarrollo consta de un script principal que se encarga de establecer las conexiones y llevar adelante el flujo de la carga. Para el diálogo con los distintos periféricos necesarios para el proyecto, se desarrollaron tres scripts. El primero se encarga de la gestión de la comunicación RFID, el siguiente se encarga de la definición de funciones para utilizar las REST API del OpenEVSE y por último un script que se encarga de la definición de las funciones para la comunicación con el sistema central CargaME. El manejo de estas conexiones en conjunto con el manejo del flujo de la carga está centralizado en otro script, siendo éste el principal. A su vez, se crearon scripts auxiliares para las pruebas realizadas durante el desarrollo. Entre ellos se encuentra una simulación del sistema central OCPP del UTE y de un lector RFID.

4.1.1. rfid_reading_loop.py

Script que levanta las librerías *GPIO* y *NFC* antes mencionadas y las utiliza para leer una tarjeta RFID que se aproxima al módulo MFRC522. Surgió un problema de compatibilidad de estas librerías con Python 3.6, por lo que este script debe correr en Python 2.7. El script consta de un loop infinito escuchando la entrada RFID y una vez recibida una tarjeta, la guarda en un archivo de texto junto con el timestamp de la lectura y vuelve al loop.

4.1.2. evse_functions.py

Este script levanta la librería *pyserial* y es donde se definen todas las funciones a utilizar para obtener y setear datos entre la Raspberry Pi y el OpenEVSE. Las funciones se definen en base a la documentación de la aplicación REST API que tiene el OpenEVSE seteado por defecto (ver Anexo 10.3).

Para cumplir con los objetivos del proyecto se desarrollaron las siguientes funciones:

- `start_connection`, establece la conexión USB-serial.
- `end_connection`, cierra la conexión USB-serial.
- `set_enable`, permite habilitar el OpenEVSE.
- `set_reset`, resetea el OpenEVSE.
- `set_disable`, permite deshabilitar el OpenEVSE.
- `set_current`, permite setear nivel de corriente.
- `get_energy_usage`, obtiene los valores de energía en Watts acumulados.
- `get_charging_data`, obtiene los valores de corriente y voltaje actuales.
- `get_current_settings`, obtiene la capacidad de corriente seteada.

- `get_status`, obtiene el estado del OpenEVSE.
- `set_display_color`, setea el color del display.
- `set_display_text`, setea texto en el display.
- `disable_by_current`, deshabilita el OpenEVSE en caso de que la corriente consumida sea mayor a la seteada. Esta función se crea para darle una protección lógica extra.

La API del OpenEVSE permite realizar otro tipo de intercambios de información como por ejemplo, relacionado a fechas, versiones de firmware, y otras acciones que se pueden encontrar en su documentación.

4.1.3. `ocpp_chargepoint_class.py`

Este script levanta la librería `ocpp` y define las funciones necesarias para la comunicación OCPP con el sistema central de UTE. Estas funciones están definidas en base a la documentación del protocolo OCPP 1.6, teniendo en cuenta los datos necesarios y opcionales en la documentación. Para cumplir con los objetivos del proyecto se desarrollaron las siguientes funciones:

- `send_boot_notification`, es el primer mensaje que se debe enviar a la hora de energizar el equipo. Reporta al sistema central el booteo de un nuevo punto de carga especificando valores propios de éste. Para la confirmación de la comunicación debe recibir una respuesta del sistema central que incluya el valor del heartbeat y otros datos (opcionales), como puede ser el timestamp con fines de sincronización.
- `Send_heartbeat`, función que envía cada determinado tiempo (definido por el sistema central) un keep alive al sistema central para notificar que sigue en funcionamiento.
- `Send_authorize`, función que envía un ID de usuario con el fin de obtener autorización del sistema central.
- `Send_start_transaction`, función que envía los datos necesarios para comenzar una transacción. Recibe como respuesta un identificador único correspondiente a la transacción inicializada.
- `Send_meter_values`, función que envía datos durante una carga. Estos datos deben ser enviados cada determinado tiempo (definido por el sistema central).
- `Send_stop_transaction`, función que envía los datos necesarios para finalizar una transacción.
- `on_remote_start_transaction`, la librería OCPP utilizada permite ejecutar funciones luego de recibida una acción, en este caso recibir un `start_transaction` externo. Esta función graba en el archivo donde se almacenan las entradas RFID el ID recibido y el timestamp correspondiente.
- `On_remote_stop_transaction`, luego de recibido un `stop_transaction` externo, graba el archivo donde se almacenan las entradas RFID el identificador único de una transacción a finalizar junto con el timestamp.

El protocolo OCPP 1.6 define otras funciones que no fueron implementadas.

4.1.4. charge_point.py

Este script se encarga de llevar adelante la lógica principal estableciendo la conexión con los periféricos a la hora de energizar el equipo y luego de establecido, se encarga del manejo del flujo de la carga. Para esto utiliza las librerías *asyncio*, *websockets* y levanta los scripts anteriores para utilizar sus funcionalidades.

Al bootear la Raspberry Pi, el script se ejecuta automáticamente inicializando en primer lugar la conexión con el OpenEVSE. Una vez establecida la conexión, procede a conectar con el sistema central OCPP estableciendo dos hilos de comunicación. Uno se encarga de la notificación inicial y el envío de heartbeats, el otro es el loop que gestiona el manejo de la carga vinculado al ingreso de un ID del módulo RFID. Esta instancia de booteo se ve reflejada en el display con dos mensajes. El primero luego de la conexión serial entre la Raspberry Pi y el OpenEVSE (Ilustración 20(a)) y el segundo luego de conectado al sistema central (Ilustración 20(b)).

El flujo principal consta de un loop infinito que se mantiene en espera de que se aproxime una tarjeta RFID para el inicio de una carga (Ilustración 20(c)). Una vez recibida esa tarjeta se chequea contra el sistema central si ese ID está autorizado. Si no está autorizado despliega en pantalla el mensaje correspondiente (Ilustración 20(d)) y vuelve al loop inicial. En caso de estar autorizado (Ilustración 20(e)) obtiene datos del estado del OpenEVSE e inicializa la transacción con el sistema central (Ilustración 20(g)).

Durante la carga realiza constantes chequeos del OpenEVSE, obteniendo datos de su estado con los cuales envía los meter values requeridos por OCPP y a vez chequea si el vehículo está conectado (Ilustración 20(f)(j)), si hay exceso de corriente (Ilustración 20(l)) o si está atravesando alguno de los errores como ventilación requerida (Ilustración 20(i)) o fuga a tierra.

El corte de una carga puede ser realizado remotamente a través del sistema central OCPP o por identificación del mismo usuario (en caso de que el usuario no sea el mismo que la inició no se corta la carga), ante estos dos casos se dispara la finalización de la transacción (Ilustración 20(h)). Para el caso en que la carga se interrumpa por alguna falla mencionada en el diagrama de flujo de errores (Ilustración 19) quedará a la espera de la identificación del mismo usuario que inició la carga para proceder a finalizar la transacción. Una vez finalizada la transacción se vuelve al loop inicial.



Ilustración 20 – Display de los estados y errores en el OpenEVSE

4.2. Blockchain

A nivel blockchain no se realizó ningún desarrollo sobre el punto de carga. Si se realizaron pruebas dentro del marco de Hyperledger Fabric, utilizando las distintas funcionalidades disponibles en la librería Python SDK Fabric antes mencionada. Es decir, la plataforma ya cuenta con el desarrollo de funciones necesario para implementar una red blockchain. Por lo tanto, lo que se estudia es la arquitectura y qué rol puede cumplir cada punto de carga dentro de la red.

La arquitectura se describe en el Capítulo 3.4.

5. Pruebas y simulaciones del prototipo

En el presente capítulo, se describirán las tres principales pruebas realizadas al equipo. Las mismas son:

- Prueba con simulador proporcionado por UTE y consumos domésticos
- Prueba con vehículo eléctrico 7,4 kW
- Prueba de gestión remota desde sistema central de UTE

En los tres casos, se excluyó de las pruebas la transacción mediante mercado pago, dado que es una funcionalidad ya implementada en la placa entregada por SimpleTECH y que no agrega valor a la validación del prototipo.

También se documenta las pruebas realizadas en relación con la red blockchain.

5.1. Proceso de carga

En primer lugar, se aclara que sobre la placa OpenEVSE no se realizó ninguna modificación, por lo que se dan como válidas las pruebas de cumplimiento con el protocolo IEC 62169 desarrolladas en el capítulo 2.1.3. Además, se mantienen activos los chequeos que realiza el dispositivo al energizarse, de forma de chequear que todas las componentes y conexiones se encuentran aptas para operación.

En las secciones siguientes, se presentará el proceso de carga dividido en los cinco bloques principales, indicando como se verifica que el prototipo cumple los objetivos planteados y necesarios para operar como punto de carga público.

5.1.1. Etapa 0: Energización del punto de carga

Es el momento en que se conecta el punto de carga a la alimentación eléctrica. Lo esperado es que, al energizar, el controlador del punto de carga (Raspberry Pi en nuestro caso) haga un booteo automático de los scripts necesarios para su funcionamiento. Luego, debe establecer una comunicación con el sistema de carga, esto implica que exista comunicación con el sistema central de OCPP mediante 3G. En caso de que se establezca correctamente la comunicación, el punto de carga debe quedar disponible y a la espera de una identificación de usuario. En caso contrario, no quedará operativo.

A continuación, se muestra los chequeos y forma de verificación correspondientes a este bloque de funciones.

Descripción	Acciones	Verificación
Energización punto de carga	Scripts corren automáticamente al encenderse controlador (Raspberry PI)	Display muestra "Iniciando... Por favor espere"
	Establecer comunicación con sistema central OCPP (función Boot Notification)	Sistema central OCPP envía "BootNotification.conf" Display muestra "Conectado a sistema central"
	Punto de carga disponible	Display muestra "Disponible. Aproxime tarjeta"

Tabla 8 – Verificaciones etapa 0

Es importante resaltar que, sumado a lo expresado en la tabla anterior, el openEVSE por defecto realiza un chequeo completo de su estado, lo cual incluye las siguientes verificaciones:

- Que existe conexión a tierra.
- Contactor funciona correctamente.
- No hay fugas a tierra.
- Temperatura del dispositivo está en los valores aceptables.

5.1.2. Etapa 1: Identificación de usuario

El siguiente paso consiste en la identificación y autorización del usuario para llevar adelante una carga. El módulo para lectura RFID agregado al openEVSE debe leer un tag y enviar dicha lectura al sistema central de OCPP mediante la función "Authorize.req". El sistema central devolverá la función "Authorize.conf", y en particular, indicará en el campo "IdTagInfo" si el tag enviado se encuentra autorizado o no. Únicamente debe aceptarse el comienzo de la carga (etapa siguiente) si el valor de dicho campo es "Accepted". Frente a cualquier otro valor, no se debe autorizar la carga y el equipo debe quedar disponible para una nueva lectura RFID. A continuación, se resumen estos pasos y la forma en que se verifica si está funcionando correctamente.

Descripción	Acciones	Verificación
Identificación de usuario	RFID no autorizado es rechazado y no se inicia la carga	Punto de carga rechaza RFID y se mantiene disponible. Display muestra "Tarjeta RFID no autorizada"
	RFID autorizado	Sistema central OCPP devuelve "Accepted" en el campo "IdTagInfo" de la función "Authorize.conf" Display despliega "Tarjeta RFID autorizada"
	Se recibe un "RemoteStartTransaction.req" desde el sistema central	Display muestra "Tarjeta RFID autorizada"

Tabla 9 – Verificaciones etapa 1

5.1.3. Etapa 2: Inicio de transacción

En esta etapa, el objetivo principal es que el sistema central inicie una sesión de carga, asignando un número único que permite identificar la misma. Una vez que iniciada la transacción de forma correcta, el cargador dejará de estar disponible y chequea que el vehículo se encuentre conectado. En caso de que así sea, se cierra el contactor y comienza el flujo de carga hacia el vehículo. En caso de que el vehículo no se encuentre conectado, se solicitará la conexión de este, con contactor abierto. Una vez inicializada la carga se podrá ver en el display la corriente instantánea consumida y la energía entregada en la sesión.

Descripción	Acciones	Verificación
Inicio de transacción	Solicitud al sistema central para iniciar transacción ("StartTransaction.req")	"TransactionID" asignado por sistema central
	Chequeo de conexión de vehículo	El cargador censa si hay algo conectado al Proximity Pilot. Display muestra "Conectar vehículo"
	Comienzo de carga	Contactor energiza vehículo. Display muestra información de carga en curso

Tabla 10 – Verificaciones etapa 2

5.1.4. Etapa 3: Carga del vehículo

Una vez iniciada la transacción y habilitado el flujo de carga, se debe permanecer censando el estado del OpenEVSE para verificar la existencia o no de posibles errores o fallas reportados por el vehículo o el OpenEVSE. En caso de existir un error, el punto de carga deberá desenergizar el vehículo en el momento y permanecer con el display fijo reportando el problema hasta que se vuelva a pasar la tarjeta RFID para finalizar la transacción.

En caso de no existir ningún error durante la carga, la misma finalizará una vez se vuelva a leer la tarjeta RFID que inició la transacción y esta sea confirmada por el sistema central.

Descripción	Acciones	Verificación
Carga de vehículo	Envío de datos al sistema central ("MeterValues.req")	Sistema central recibe la información.
	Error: No Ground fault (state = 7)	Display muestra "Fuga a tierra"
	Error: Ventilation required (state = 4)	Display muestra "Ventilación requerida"
	Error: Pilot signal (state = 1)	Display muestra "Vehículo desconectado"
	Error: Otro estado recibido por el OpenEVSE	Display muestra "Punto de carga en falla"
	Error: Exceso de corriente	Contactor abre si corriente medida mayor a corriente seteadas. Display muestra "Exceso de corriente"

Tabla 11 – Verificaciones etapa 3

5.1.5. Etapa 4: Cierre de sesión de carga

Para terminar la sesión y darle fin es necesario volver a leer la tarjeta RFID. Se debe verificar si la tarjeta leída es la misma que la tarjeta que inició la transacción. En caso afirmativo se debe enviar un "Authorize.req" con el valor de la tarjeta RFID al sistema central, y éste debe responder con un "Authorize.conf". A continuación, se envía al sistema central un mensaje de "StopTransaction.req" con los datos de la carga, el Id de la transacción y el "TagId". Este mensaje debe de responderse con un "StopTransaction.conf" que es la confirmación a la solicitud de finalización de la transacción. De esta forma, el cargador volverá a quedar disponible, volviendo al estado final del bloque 0 (Disponible).

Descripción	Acciones	Verificación
Finalizar la carga	Cierre de sesión de carga	Lectura tarjeta RFID que inició carga display muestra "Fin de carga en proceso". Otra tarjeta RFID display muestra "Tarjeta RFID no valida".
	Solicitud al sistema central para finalizar transacción ("StopTransaction.req")	Se recibe un "StopTransaction.conf" del sistema central. Display muestra "Transacción exitosa".
	Se recibe un "RemoteStopTransaction.req" desde el sistema central	Display muestra "Fin de carga en proceso".

Tabla 12 – Verificaciones etapa 4

5.2. Pruebas con simulador de UTE

Previo a realizar simulaciones con un vehículo eléctrico, se probó el equipo con el simulador de UTE. Para ello, se coloca como consumo una lámpara de tungsteno 60 W que simulará la presencia de un vehículo, y ayuda a verificar que el punto de carga cierra contactor y energiza el conector. Con el simulador se fuerzan los estados de error por lo que se puede chequear que el punto de carga responde de acuerdo con lo esperado.

Las mismas pruebas fueron realizadas con una jarra eléctrica y un secador de pelo (1600 W cada uno) conectados al mismo tiempo. En esta última prueba, adicional a lo visto con la lámpara se puede probar el corte de energía en caso de que el vehículo consuma más de lo disponible por el punto de carga. Para ello, se fija la corriente máxima en 10 A. Dado que ambos equipos conectados al mismo tiempo consumirán aproximadamente 13 A, es de esperar que el punto de carga corte el flujo por exceso de corriente. La primera vez que se realiza la prueba, el OpenEVSE no cortó la energía. En el proceso de carga habitual, a través del control pilot el punto de carga informa al vehículo cuál es la máxima corriente disponible, pero es el vehículo quien fija la corriente a consumir en base a la información recibida. En el caso de la prueba con la jarra y secador, los mismos consumen su corriente nominal porque no tienen capacidad de analizar la información enviada en el control pilot.

Por lo anterior, fue necesario desarrollar una lógica que asegure que el vehículo no consume más de lo que la red o el punto de carga pueden despachar. Dicha funcionalidad se describe en el Capítulo 4.1.2. Los resultados a continuación muestran las pruebas con la lógica implementada.



Ilustración 21 – Circuito auxiliar con simulador de UTE

5.3. Prueba de carga con vehículo eléctrico

Se realizaron pruebas de carga en un vehículo eléctrico de propiedad de UTE para corroborar el correcto funcionamiento del cargador desarrollado en un ambiente real. Para esta prueba contamos con una camioneta Renault Kangoo Z.E. de 7,4kW.

La prueba constó de un proceso completo desde que se energiza el punto de carga (etapa 0) hasta que finaliza la transacción (etapa 4). Considerando que al utilizar un vehículo no se puede simular fallas, la etapa 3 no se consideró dentro del protocolo de pruebas para un vehículo eléctrico real ya habiendo sido probadas las fallas con el simulador. Las pruebas dieron resultados correctos considerando la carga del vehículo eléctrico exitosa.



Ilustración 22 – Carga de vehículo eléctrico en proceso

5.4. Pruebas remotas

Un factor importante que probar es el manejo de la carga de manera remota, si bien ya se probó que ninguna carga se realiza sin la autorización del sistema central, en todos los casos se utilizó en los extremos del proceso la identificación del usuario a través de RFID.

Para esta prueba se dejaron los equipos conectados y operativos, con el punto de carga a la espera de la identificación de un usuario. Se contactaron personas de Movilidad Eléctrica con acceso al sistema CargaME y desde la interfaz web del sistema se ejecutó la función OCPP “RemoteStartTransaction.req”, la cual envía un ID de usuario. Se probó que el punto de carga recibe la notificación y comienza el proceso de carga. Luego de iniciada se probó ejecutar la función OCPP “RemoteStopTransaction.req”, la cual brinda un número de transacción. Esta notificación también fue recibida, y según las funcionalidades desarrolladas se corrobora que el número de transacción recibido coincide con el que se está ejecutando para empezar el proceso de finalización de carga. En el anexo 10.4 se puede ver el log del sistema central CargaME brindado por personal de Movilidad Eléctrica de UTE

Esta prueba tuvo sus contratiempos, en sus primeros intentos las funciones ejecutadas remotamente no eran recibidas por la Raspberry Pi. Se realizó un estudio más a fondo del funcionamiento de las librerías websockets y asyncio, y esto llevó a tener que cambiar algunas de las funcionalidades desarrolladas. Además, se realizaron cambios en la configuración del firewall del modem del punto de carga, dado que, en pruebas dentro de una red local con un modem de hogar contra el sistema central simulado en un pc particular, sí funcionaba. Al tratarse de un modem con IP privada dentro de la red corporativa de UTE, no se identifican peligros asociados a ataques, por lo que se decide bajar el nivel de seguridad del firewall para

permitir el pasaje de solicitudes a dispositivos dentro de la LAN. Luego de estos cambios la comunicación se realizó sin inconvenientes, dando esta prueba resultados exitosos.

5.5.Resultados

Descripción	Acciones	Verificación	Cargas domésticas	Vehículo eléctrico
0 - Energización punto de carga	Scripts corren automáticamente al encenderse controlador (Raspberry PI)	Display muestra "Iniciando... Por favor espere"	OK	OK
	Establecer comunicación con sistema central OCPP (función Boot Notification)	Sistema central OCPP envía "BootNotification.conf"	OK	OK
		Display muestra "Conectado a sistema central"		
Punto de carga disponible	Display muestra "Disponible. Aproxime tarjeta"	OK	OK	
1 -Identificación de usuario	RFID no autorizado es rechazado y no se inicia la carga	Punto de carga rechaza RFID y se mantiene disponible.	OK	OK
		Display muestra "Tarjeta RFID no autorizada"		
	RFID autorizado	Sistema central OCPP devuelve "Accepted" en el campo "IdTagInfo" de la función "Authorize.conf"	OK	OK
		Display despliega "Tarjeta RFID autorizada"		
Se recibe un "RemoteStartTransaction.req" desde el sistema central	Display muestra "Tarjeta RFID autorizada"	OK	OK	
2 - Inicio de transacción	Solicitud al sistema central para iniciar transacción ("StartTransaction.req")	"TransactionID" asignado por sistema central	OK	OK
	Chequeo de conexión de vehículo	El cargador censa si hay algo conectado al Proximity Pilot. Display muestra "Conectar vehículo"	OK	OK
	Comienzo de carga	Contactor energiza vehículo.	OK	OK
Display muestra información de carga en curso				

3 - Carga de vehículo	Envío de datos al sistema central ("MeterValues.req")	Sistema central recibe la información.	OK	OK
	Error: No Ground fault (state = 7)	Display muestra "Fuga a tierra"	OK	No testado
	Error: Ventilation required (state = 4)	Display muestra "Ventilación requerida"	OK	No testado
	Error: Pilot signal (state = 1)	Display muestra "Vehículo desconectado"	OK	No testado
	Error: Otro estado recibido por el OpenEVSE	Display muestra "Punto de carga en falla"	OK	No testado
	Error: Exceso de corriente	Contactador abre si corriente medida mayor a corriente seteada.	OK	No testado
Display muestra "Exceso de corriente"				
4 - Finalizar la carga	Cierre de sesión de carga	Lectura tarjeta RFID que inició carga display muestra "Fin de carga en proceso".	OK	OK
		Otra tarjeta RFID display muestra "Tarjeta RFID no valida".		
	Solicitud al sistema central para finalizar transacción ("StopTransaction.req")	Se recibe un "StopTransaction.conf" del sistema central. Display muestra "Transacción exitosa".	OK	OK
	Se recibe un "RemoteStopTransaction.req" desde el sistema central	Display muestra "Fin de carga en proceso".	OK	OK

Tabla 13 – Resultados de las pruebas

5.6. Pruebas blockchain

5.6.1. Red local con Hyperledger Fabric

Dentro de las posibilidades que brinda Hyperledger Fabric una de ellas es el despliegue de una red blockchain on-premise. Como primera prueba, se llevó adelante en un PC particular el procedimiento establecido en el tutorial *Build You First Network* [32] establecido en el release 1.4. Esto implicó la instalación de Docker, Docker-compose, el lenguaje de programación Go, así como todos los archivos binarios, imágenes y otros requisitos menores de Hyperledger Fabric (redes predefinidas para testear, etc.). Se buscó con estas pruebas conocer el procedimiento completo de la puesta en marcha de una red blockchain, desde la definición de los actores (orderer, organizations and peer) hasta la creación de canales, inicialización y consultas a una chaincode. Esta prueba, si bien no fue sencilla dada la cantidad de etapas que se debe atravesar, fue exitosa y cumplió con el objetivo de conocer las distintas etapas y sus implicancias.

Se realizó un intento de utilizar la Raspberry Pi como un peer, lo cual implicó las instalaciones antes mencionadas. Esto generó que el dispositivo quedara sumamente lento solo

por estar corriendo docker e intentar levantar la imagen correspondiente a un peer (hyperledger/fabric-peer:1.4.0), impactando directamente en el flujo de carga ya operativo. Por este motivo se descartó esta propuesta, ya que el hardware utilizado para el prototipo no es lo suficientemente robusto.

5.6.2. Light client en Raspberry Pi

Hyperledger fabric ofrece dentro de sus opciones un kit de desarrollo (SDK) para poder realizar una red blockchain. Hoy en día cuenta con el SDK desarrollado en los lenguajes Go, Node.js y Python. Se utilizó el SDK de Python [30] para continuar con la misma línea de lo desarrollado anteriormente. Se buscó en este caso levantar una red blockchain en un pc particular y en lugar de utilizar la Raspberry Pi como un peer, el SDK ofrece la funcionalidad de levantar un fabric-network, lo que permite utilizar el dispositivo como light client. Esto limita sus funcionalidades, pero mantiene la posibilidad de solicitar credenciales al servidor de credenciales definido en la red blockchain, así como crear canales e iniciar y consultar chaincodes. Siendo esta una solución ideal para el fin del proyecto.

A la hora de llevar a cabo estas pruebas se logró sin inconvenientes crear y levantar una red blockchain en un PC particular, publicando en internet el puerto correspondiente a la imagen hyperledger/fabric-ca:1.4.0 de docker que ejerce como servidor certificador. A la hora de realizar la instalación y puesta a punto del ambiente en la Raspberry Pi comenzaron a surgir incompatibilidades entre algunas librerías Python requeridas y el sistema operativo utilizado en el dispositivo (Linux/Raspian Jessie). Esto llevó a que no se pudiera realizar la prueba deseada de pedir un certificado desde la Raspberry al PC particular a través de internet desde el punto de carga.

5.6.3. Red en IBM Cloud

IBM Cloud en un esfuerzo en conjunto con Hyperledger, brinda la posibilidad de levantar una red blockchain en su totalidad en la nube [33]. En su web, suscripción mediante, ofrece mes gratis de pruebas para poder levantar y experimentar el servicio ofrecido, que incluye capacidad de almacenaje, cpu y memoria, así como las librerías de Python correspondientes para el manejo en CLI desde un pc particular con salida internet. Esto se probó en un pc particular, procediendo a instalar las librerías particulares de IBM Cloud y kubernetes permitiendo levantar una red con varios peers utilizando las herramientas brindadas. A la hora de llevar adelante la instalación de las librerías de IBM Cloud en la Raspberry Pi, con el fin de poder pedir certificados y gestionar transacciones, se volvieron a generar incompatibilidades entre librerías, quedando trunca la prueba de funcionamiento.

6. Viabilidad de implementación del prototipo en el mercado

6.1. Costos

Para evaluar costos de implementación de un prototipo desarrollado localmente, se debe tener un enfoque tanto cuantitativo como cualitativo. Esto apunta a analizar por un lado el costo de materiales, armado del punto de carga y desarrollo de funcionalidades, además del costo de mantenimiento. Por otro lado, se analizan las ventajas y desventajas relacionadas a ser fabricante del punto de carga para la realización de mantenimientos y actualizaciones de forma que el mismo prolongue su vida útil.

Análisis cuantitativo

Para realizar el análisis de costo de producción de un punto de carga, se realizan algunas suposiciones de forma de llegar a un valor realista pero conservador, que contemple un peor caso. En ese sentido, se define la utilización de equipamiento de la marca ABB [34], tomando valores de venta minoristas en plaza. Lo mismo aplica para la envolvente, que se considerará de un tamaño estándar ofrecido por la empresa Rittal [35], líder mundial en envolventes eléctricas. Es decir, en ambos casos se están considerando materiales de primera calidad, lo cual tiende a encarecer el costo del punto de carga, pero a hacerlo más confiable y de una vida útil superior. En cuanto al controlador, se utiliza el OpenEVSE que fue testeado dentro del alcance del proyecto. Se debe realizar cambios menores para que pueda interpretar datos tanto para una carga trifásica como monofásica, pero esto no cambia el costo de compra. Por último, se considera un cargador con cable y conector tipo 2, lo cual también tiene valores conocidos en el mercado y es necesario importar dichos materiales, ya que en plaza aún no hay oferta.

Para la mano de obra, se considera que el armado del tablero insume dos días de trabajo de un oficial electricista, a partir que se cuentan con todos los componentes. Se estimó su valor hora en \$700 en base a cotizaciones de instalaciones eléctricas de 2019. Dentro de las tareas de armado, se considera tiempo para testeo del correcto funcionamiento del cargador.

Teniendo en cuenta lo anterior en la Tabla 14 se resumen los costos estimados para la producción de un cargador trifásico de 22 kW (32 A).

Para que el análisis sea completo, hay que considerar el esfuerzo de desarrollo necesario para que el prototipo quede operativo de forma fluida y confiable, con todas las funcionalidades de OCPP 1.6 operativas. Para ello se requiere un análisis a fondo de las funcionalidades pendientes de desarrollar completamente. Esto es:

- Optimización del script cargado en Raspberry PI
- Implementación completa de OCPP 1.6

Opcionalmente, se podría desarrollar todo lo que se encuentra implementado en el OpenEVSE (IEC 62169, detección de errores y medición de energía) para desarrollarlo también dentro de la Raspberry. Esto no fue estudiado dentro del proyecto.

Adicionalmente, se debería contemplar un costo anual por mantenimiento del equipo, estimado en media jornada (4 horas) de trabajo por equipo vendido por año (USD 80 anuales por equipo).

En comparativa con los cargadores trifásicos 22 kW tipo tótem que incorporó UTE en su red de carga el precio promedio de un cargador con esas características ronda los USD 1500 sin impuestos. En una primera aproximación, salta a la vista que sería competitivo la posibilidad de desarrollar un prototipo nacional cuyo principal diferenciador sea la flexibilidad para adaptarse a las tecnologías venideras.

Previo a pasar al análisis cualitativo, es pertinente realizar la aclaración que todo el desarrollo de software es reutilizable para cargadores de mayor potencia (43kW, por ejemplo). En ese caso, los costos a actualizar serían de los equipos eléctricos (contactor, interruptor y diferencial). Tanto la placa OpenEVSE como el display, equipos de medida y tiempo de fabricación de cargador no se verían afectados.

Item	Referencia	Fabricante	Cantidad	Unidad	Costo unitario (USD)	Subtotal (USD)
Interruptor general 4p, 32A	MCB S204 - C32	ABB	1	un	30.00	30.00
Diferencial 4p, 32A, 30mA	F204 A - 32A, 30mA	ABB	1	un	45.00	45.00
Contactador trifásico 32A, bobina 230V	A50-30-00	ABB	1	un	65.00	65.00
Transformador corriente	CR8450-1K-T7QC	Magnetics Inc.	3	un	22.00	66.00
Controlador SAVE	openEVSE v5	openEVSE	1	un	100.00	100.00
Display	ER-TFT050	EastRising	1	un	13.00	13.00
Cable 32A con terminal Tipo 2	EV-T2G3C	Phoenix Contact	1	un	300.00	300.00
Gabinete	EB 1578.500	Rittal	1	un	90.00	90.00
Raspberry PI	Raspberry PI 3	Raspberry	1	un	70.00	70.00
Módulo RFID	RC522	Mifare	1	un	5.00	5.00
Armado y accesorios tablero	Cotización año 2019	-	16	Horas Hombre	17.00	272.00
						1056.00

Tabla 14 – Rubrado de costos de producción de un cargador

Análisis cualitativo

Para realizar el análisis cualitativo, se debe tener presente el contexto actual del desarrollo de la movilidad eléctrica. En particular, la velocidad de los cambios que se vienen implementando y las dificultades que conlleva convivir con puntos de carga de diversos fabricantes, varias versiones OCPP (incompatibles entre sí) e interfaces para el usuario diferentes. Esto deriva en que una compra de un gran volumen de cargadores implica una optimización económica pero también lleva asociado el riesgo de la obsolescencia de ese grupo de cargadores en poco tiempo. A modo de ejemplo: la gran mayoría de los SAVE comprados por UTE para la primera instalación fueron en base a OCPP 1.5 que era la versión del protocolo válida en ese momento. Menos de un año después, los fabricantes comienzan a ofrecer únicamente OCPP 1.6, que según lo mencionado en el capítulo 2.1.2 es incompatible con su versión anterior. Esto implicó actualizar el sistema central de carga para poder continuar extendiendo la red de puntos de carga y, además, contactar a los fabricantes para desarrollar un plan de actualización para pasar los puntos de carga existentes a OCPP 1.6. En definitiva, en una red nacional para carga de vehículos eléctricos en pleno desarrollo, ya se invirtió tiempo y dinero para evitar su obsolescencia a pesar del poco tiempo de uso.

La problemática descrita en el párrafo anterior es la principal justificación para el desarrollo de un SAVE propiedad de UTE (u otro organismo local). Permitiría total flexibilidad para las modificaciones sobre el equipo, permitiendo planificar las actualizaciones a desarrollar e implementarlo de forma ordenada sin perder la disponibilidad de los puntos de carga. Además, desarrolla una fuente de trabajo y conocimiento nacional, en un área de interés mundial. Esto podría derivar también en la venta de un nuevo producto de valor agregado hacia el exterior, principalmente en la región donde la movilidad eléctrica se viene desarrollando a un ritmo mucho más lento que en Europa o Uruguay mismo.

Si nos enfocamos en las desventajas, el primer punto a mencionar es el tema del desarrollo de la marca y certificaciones. Se competiría contra fabricantes con experiencia desarrollada en

tema de puntos de carga. Dado que en Uruguay existe un monopolio (UTE) que es en definitiva quien debería aprobar la calidad, no sería mayor problema para los privados o incluso para la UTE la incorporación de un punto de carga de desarrollo nacional. Es decir, se debería desarrollar un convenio para la implementación y mejora continua del mismo, en colaboración entre todas las empresas interesadas en incorporarlo y ofrecerlo a la red de carga.

Si lo enfocamos en un plan estratégico a largo plazo, la tendencia mundial es a tercerizar de forma de aprovechar el “know-how” de empresas especializadas. Ese conocimiento aún no está desarrollado lo suficiente por lo que aún no hay suficiente especialización que suponga una ventaja frente al desarrollo propio. Como fue desarrollado anteriormente, los estándares están en proceso de creación por lo que las bases de funcionamiento son las que están variando continuamente. Es de suponer que largo plazo se impongan protocolos y formas de funcionamiento básicas definidos por estándares internacionales. En ese contexto, el valor agregado va a estar en la inteligencia del punto de carga para ser parte de redes más complejas, que permitan flujos de potencia tanto para cargar el vehículo (“grid-to-vehicle”) como para que el vehículo vuelque potencia a la red (“vehicle-to-grid”). También entra en juego la capacidad para realizar cargas inteligentes, en función de la disponibilidad de energía renovable en la red. Pero entonces ya no estamos hablando de un flujo de carga básico, sino que estamos hablando de una operativa mucho más compleja, donde la especialización tomaría un rol central. Si bien esto se encuentra en desarrollo, su auge sería posterior a que el vehículo eléctrico este masivamente instalado en las ciudades, que se estima será entre 2025-2030.

En conclusión, a corto plazo y para el desarrollo de conocimiento y productos de valor agregado nacionales, además de la optimización operativa y económica, se considera viable la producción de un punto de carga propio. Se debería realizar dentro de un plan estratégico que derive en la venta de la marca de dicho punto de carga, para luego tercerizar la especialización de este a empresas con el conocimiento acorde y destinado a ello.

6.2. Modelo de negocio para el cargador

La problemática a partir de la cual surgió la posibilidad de realizar este proyecto se explica en el Capítulo 1.3. Como solución se encontró la posibilidad de manipular un cargador de software abierto preexistente para uso personal y convertirlo en uno de uso público para que sea utilizado por UTE en sus puntos de carga.

El modelo de negocio descrito a continuación se basa en que el punto de carga ya se encuentra en etapa de producción (no de prototipo) y con la posibilidad de realizar transacciones en una red blockchain.

6.2.1. Quién

El modelo de negocio se basa en que el cliente es UTE con el cual se puede establecer una relación de creación colectiva en la que las dos partes, tanto cliente como proveedor, puedan colaborar en el desarrollo de las necesidades expresadas por el cliente para el cumplimiento de normativas o funcionalidades deseada.

6.2.2. Qué - Game Changing

Si buscamos establecer un modelo de negocio futuro para el prototipo desarrollado, deberíamos basarlo en su diferenciador que es la posibilidad de trabajar dentro de una red de nodos de blockchain. Como fue mencionado anteriormente, todas las funcionalidades desarrolladas fueron para llevar el cargador pensado para uso doméstico a uno de uso comercial. Pero la prueba de concepto de blockchain es la que agrega valor respecto al resto de puntos de carga del mercado.

En ese sentido, todo lugar público o privado (estacionamientos, empresas, supermercados, restaurantes u otros) que dispongan de un punto de carga y el mismo esté integrado a la red blockchain, podrá ofrecer el servicio de recarga como un valor agregado a su comercio, pero el costo de la energía y validación de la transacción no será su responsabilidad, sino que irá directamente al gestor de energía (UTE en el caso de Uruguay). De esta forma se estaría incentivando a la movilidad eléctrica en la rutina de cada persona y agregando un servicio de valor a los comercios existentes.

Más aún, consideremos el caso de un punto de carga en Argentina, para fijar ideas, y un vehículo matriculado en Uruguay. En las condiciones actuales, sería imposible que se pueda realizar la carga fuera del país, a menos de realizar algún tipo de trámite en el país donde está ubicado el punto de carga. Supongamos ahora que dicho punto de carga pertenece a la misma red blockchain en la cual está UTE. Esto permitiría tener un smart contract conocido por los puntos de carga de ambos países, generando interoperabilidad entre cualquier punto de carga que esté dentro de la red, sin importar el país o región donde esté ubicado, sin perder trazabilidad ni transparencia. Se cambiaría el concepto de venta de un punto de carga (producto) para pasar a comercializar un servicio de carga.

Dentro del mismo modelo de negocio, hay una optimización de recursos ya que se estaría prescindiendo de un data center y centro de control perteneciente a una única empresa, distribuyendo y respaldando la información de forma segura entre todos los miembros de la red. Otra de las ventajas generadas es la transparencia que genera para llevar adelante las transacciones, dado que entre los agentes pertenecientes a la red deberían regularse y sería muy difícil caer en transacciones abusivas para el usuario final.

6.2.3. Cómo

Para poder llevar adelante la producción, el desarrollo y la venta del SAVE es necesario crear asociaciones tanto con el cliente como con los posibles proveedores de material para la producción del cargador, teniendo como principal interés una asociación con la marca OpenEVSE (cargador base sobre el que se desarrolla el proyecto). Es necesario contar con un equipo de desarrolladores, un equipo de soporte posventa y un equipo comercial para llevar a cabo el trato con el cliente.

6.2.4. Cuánto

Para este modelo de negocio la fuente de ingreso estará dada por la venta del SAVE a UTE y por la venta de servicio de mantenimiento. Los costos serán variados, dado que se necesita de personal para el desarrollo del producto y para el mantenimiento, por lo tanto, el costo de horas hombre será necesario tenerlo en cuenta. Los otros costos involucrados son la compra del OpenEVSE, caja estanco, mangón y conector, placa RFID, Raspberry Pi, y algún otro accesorio que se requiera, los cuales fueron estimados en el Capítulo 6.1.

7. Plan de Trabajo y Cronograma

En la Ilustración 23 se muestra el diagrama de Gantt inicial del proyecto. A grandes rasgos consiste en tres etapas principales:

- Estudio del cargador y tecnología
- Desarrollo de nuevas funcionalidades
- Pruebas y demo del prototipo

Respecto a la planificación, hubo un atraso relacionado principalmente a un desvío de alcance identificado durante el desarrollo del proyecto. Dicho desvío se trata del desarrollo del protocolo OCPP 1.6. El paquete de trabajo insumió aproximadamente dos meses de trabajo y afectó el cumplimiento de los plazos propuestos. Sin OCPP el cargador no cumpliría los estándares mínimos para funcionamiento y tampoco podría comunicar con el sistema de gestión central de UTE. Además, las pruebas con el vehículo fueron realizadas en paralelo durante todo el desarrollo del proyecto, en lugar de un período de tiempo concreto como fue planificado al inicio. Es decir, ante cada desarrollo de una nueva funcionalidad, se procedía a probar el OpenEVSE, es por eso por lo que la tarea dura desde que se completó la etapa de estudio hasta el final del trabajo.

Por otro lado, también se quitó del alcance la modificación del cargador para permitir que el mismo pueda realizar cargas trifásicas. Se evaluó que no agregaba valor al análisis del prototipo y era una inversión en costo un que no se justificaba. Todo el análisis de funcionalidades nuevas es independiente de que el cargador sea monofásico o trifásico.

Teniendo en cuenta dichos desvíos, en la Ilustración 24 se presenta el cronograma real del proyecto. Las tres columnas sombreadas en celeste refieren a la línea base trazada al comienzo del proyecto. La sombreada en rojo refiere a los plazos reales del proyecto. Como se puede ver, las tareas de estudio y evaluación del cargador fueron completadas según la planificación sin presentar mayores desvíos. Las tareas de desarrollo insumieron más tiempo del planificado principalmente por un error en la estimación del esfuerzo.

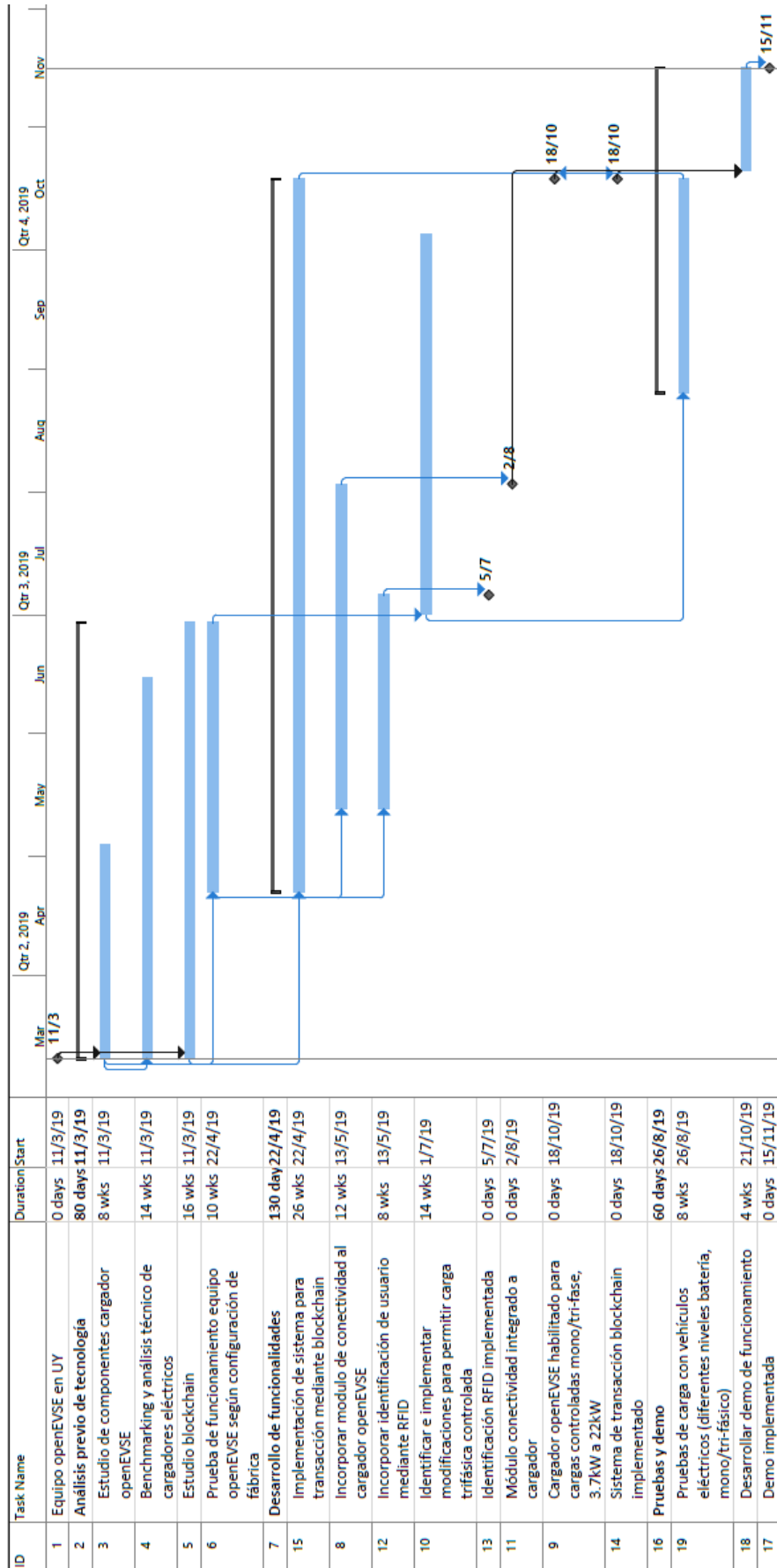


Ilustración 23 – Cronograma inicial

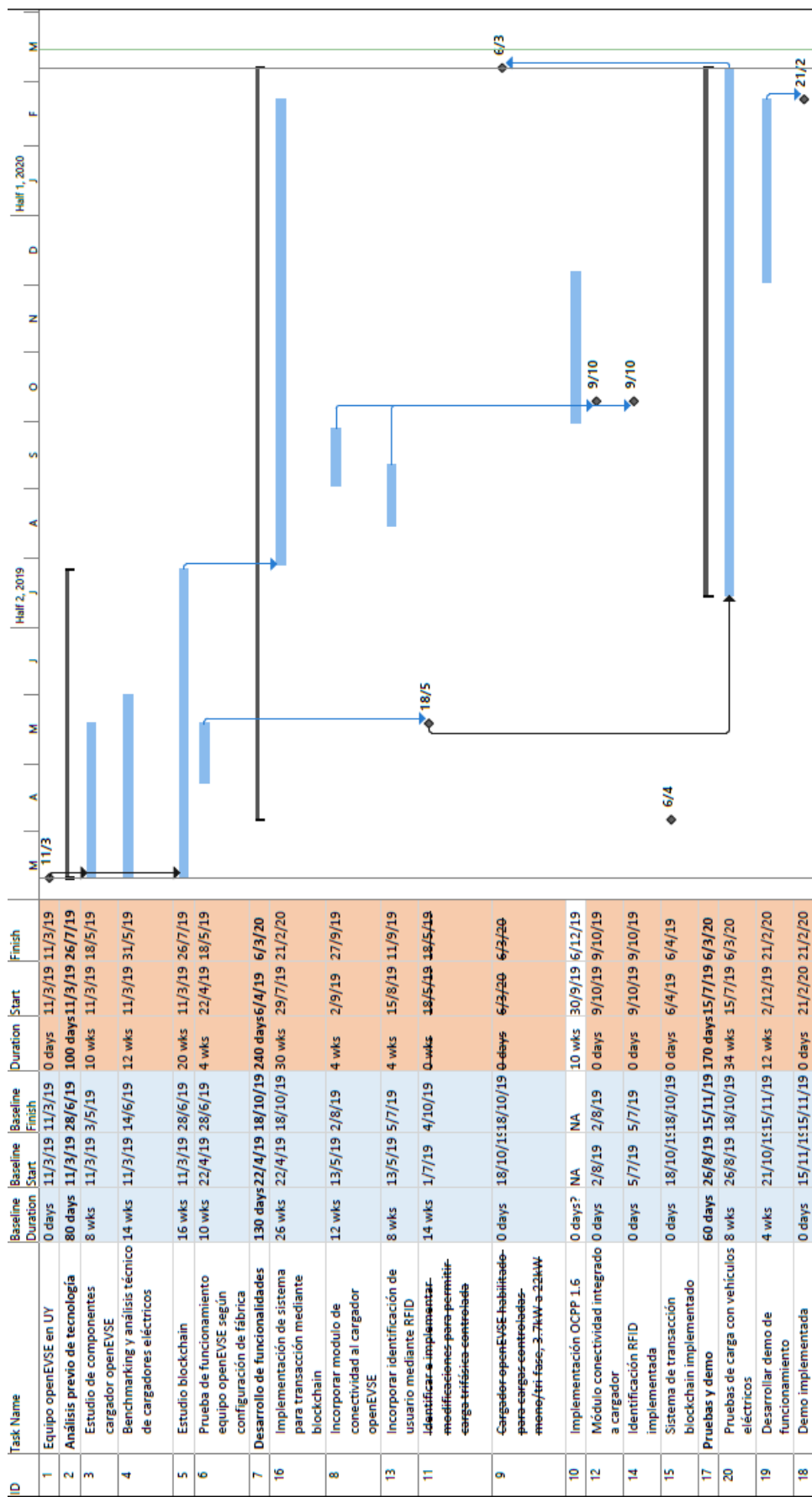


Ilustración 24 – Cronograma real vs línea base

8. Conclusiones

Retomando los planteado al inicio del proyecto de grado, el mismo consistía en el desarrollo de un SAVE público a partir de uno destinado al uso privado en una vivienda. Dicho objetivo fue logrado con éxito, habiendo implementado y probado sobre un kit de código abierto el protocolo OCPP 1.6, comunicación 3G e identificación de usuario. Dichos componentes permiten instalar el equipo en cualquier punto con suministro eléctrico y realizar una sesión de carga validada y monitoreada por el sistema central de UTE.

Tomando en cuenta los cuatro objetivos específicos planteados en el capítulo 1.2, se cumplieron casi en su totalidad.

El primero planteaba que el punto de carga pueda ser operado por UTE, permitiendo realizar el inicio y cierre de una sesión de carga remotamente. Si bien se deben realizar mejoras a nivel del script que gestiona el punto de carga para que el RFID y la operación remota de UTE convivan sin interferencias o errores, se probó satisfactoriamente el inicio y detención remota. El principal punto de mejorar está relacionado a los tiempos de espera de las funciones internas al script.

El segundo objetivo apuntaba a incorporar la identificación de usuario mediante la tecnología RFID y la implementación de un método de pago. Al integrar la placa suministrada por SimpleTECH (Raspberry PI) al OpenEVSE, se logra incorporar ambas funcionalidades. Dicha placa cuenta con lector RFID, que trabaja en la misma frecuencia que las tarjetas de identificación de UTE. También tiene implementado un sistema de prepago, de forma de pagar el servicio a medida que se va consumiendo energía.

En cuanto al tercer objetivo, se planteaba estudiar la viabilidad de que el punto de carga integrara una red blockchain, pero no buscaba la implementación de este como un nodo. Dentro de las pruebas realizadas se detectó cual es la solución adecuada para el prototipo (light client). Si bien dentro de las pruebas que se realizaron en la Raspberry PI no se logró concretar ninguna que demuestre su viabilidad, esta no se descarta ya que no se realizaron pruebas en los otros lenguajes que está desarrollado el SDK de Hyperledger Fabric, ni tampoco realizar actualizaciones en el sistema operativo de la Raspberry Pi.

Por último, se planteaba la realización de una demo, utilizando un vehículo eléctrico. Esto fue logrado al ejecutar una sesión de carga con una Renault Kangoo Z.E. 7,4 kW propiedad de UTE. La prueba chequeaba desde la energización del punto de carga, hasta el cierre de la sesión, incluyendo identificación de usuario, conexión con sistema central OCPP y la regulación de corriente de forma de que la instalación esté protegida. La sesión de carga fue ejecutada con éxito según se describió anteriormente.

Sumado a la anterior, una conclusión adicional es que es viable desarrollar un punto de carga público en Uruguay, tanto desde el punto de vista técnico como económico. Es decir, convertir el prototipo en un producto implicaría horas de desarrollo e inversión en materiales, pero lograrían un producto competitivo respecto al resto de cargadores del mercado. La gran ventaja competitiva sería la flexibilidad para actualizar y adaptar a los nuevos modelos de negocio relacionados a la movilidad eléctrica. La contra está relacionada a la patente y certificación del producto, aunque en este rubro se requiere un análisis más a fondo que no fue parte del alcance del presente proyecto.

Las mejoras a realizar en el prototipo desarrollado se resumen a continuación:

- **Hardware:** es necesario integrar las protecciones (termomagnética y diferencial) además de sustituir los cables por unos de dimensión acorde, según explicitado en capítulo 2.3. También se debería sustituir el contactor y agregar transformadores de corriente, de forma tal de que el equipo permita una carga trifásica en 400V y

hasta 32A (22 kW). Esto lleva a la necesidad de instalar todas las componentes en una nueva envolvente de tamaño adecuado.

- Software: se debe optimizar el script de forma que disminuyan los tiempos de espera mientras se ejecutan las funciones de OCPP. Actualmente, la experiencia de usuario no es la óptima debido a la espera en cada paso (encendido, identificación, cierre de sesión).
- Comunicaciones: entre la Raspberry PI y el OpenEVSE, se debería buscar una alternativa al serial-USB, dado que el mismo es poco robusto a la pérdida de información y su velocidad es lenta.

Un punto que no fue analizado en detalle en el proyecto es la posibilidad de sustituir la placa OpenEVSE, incorporando sus funcionalidades (a partir de un código abierto existente [36]) en la Raspberry PI. Esto implicaría adaptar el hardware para el manejo de las entradas de los equipos de medición de corriente, control pilot, proximity pilot y display. Se deberían volver a realizar las pruebas de cumplimiento del protocolo IEC 62169 y las protecciones del OpenEVSE, similar a lo realizado en Capítulo 2.

Mirando a futuro, sería de gran utilidad e interés investigar el pago a través de incentivos digitales transables y la implementación de funcionalidades de OCPP 2.0. La primera área, permitiría crear modelos de negocios innovadores, que incentiven a la movilidad eléctrica utilizando tecnologías modernas como lo es blockchain. También podría ayudar a la optimización del uso de la energía relacionada a los vehículos eléctricos, a través de una *Smart Grid* por ejemplo. Por otro lado, OCPP 2.0 está en desarrollo y también va rumbo a la gestión inteligente de la carga, por lo que dominar dicho protocolo permitiría estar trabajando con puntos de carga de última generación a nivel mundial.

Para finalizar, el trabajo permitió al equipo de proyecto ahondar en la movilidad eléctrica, un tema en desarrollo y con mucho potencial para el futuro, así como también interiorizarse en nuevas tecnologías como lo es blockchain, obteniendo un conocimiento más aplicado de la misma.

9. Referencias

- [1] «ICT4V,» [En línea]. Available: <https://ict4v.org/es/inicio>.
- [2] «openEVSE,» [En línea]. Available: <https://www.openevse.com/>..
- [3] «SimpleTECH,» [En línea]. Available: <https://www.simpletech.uy/>.
- [4] «Movilidad Eléctrica,» UTE, [En línea]. Available: <https://movilidad.ute.com.uy/carga.html?tab=evolucion-de-la-red>.
- [5] «Ministerio de Industria, Energía y Minería,» [En línea]. Available: <https://www.miem.gub.uy/energia/movilidad-electrica>.
- [6] «Open Charge Alliance,» [En línea]. Available: <https://www.openchargealliance.org/>.
- [7] «AGESIC,» [En línea]. Available: <https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/>.
- [8] «Presidencia de la República,» 23 Setiembre 2019. [En línea]. Available: <https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/politicas-y-gestion/plan-de-gobierno-digital-uruguay-2020>.
- [9] *Curso "Electric Vechiles: Introduction"*, TU Delft, 2019.
- [10] MIEM, «Pruebas de campo automóvil 100% eléctrico,» Montevideo, 2014.
- [11] S. Nakamoto, «Bitcoin: A Peer-to-Peer Electronic Cash System,» 2008.
- [12] *Curso "IBM Blockchain Foundation for Developers"*, Coursera.org, 2019.
- [13] «Open Charge Point Protocol 1.6,» Open Charge Alliance, Arnhem, 2017.
- [14] «Electric Vehicle Charging: J1772 / OpenEVSE,» Chis Howell, 2015.
- [15] «Smart Electric Vehicle Charging System,» University of Agder, Noruega, 2017.
- [16] UTE, «Pliego Tarifario,» Montevideo, 2019.
- [17] *Plataformas blockchain y escenarios de uso*, Montevideo: Facultad de Ingeniería UDELAR, Marzo 2019.
- [18] «OpenEVSE Guides | Advanced Charging Station,» Christopher Howell, 2018. [En línea]. Available: https://www.openevse.com/guide_advanced.html#.
- [19] «Reglamento de baja tensión. Capítulo 2. Anexo: Instalaciones interiores o receptoras,» UTE, Montevideo, 2001.
- [20] «MFRC522. Standard performance MIFARE and NTAG frontend,» NXP Semiconductors N.V., Eindhoven, Holanda, 2016.
- [21] «Serial Communications with OpenEVSE,» OpenEVSE LLC, 2020. [En línea]. Available: <https://openevse.dozuki.com/Guide/Serial+Communications+with+OpenEVSE/13?lang=en>.
- [22] «Comunicación RS485 con Arduino,» Naylamp Mechatronics, 2016. [En línea]. Available: https://naylampmechatronics.com/blog/37_Comunicaci%C3%B3n-RS485-con-Arduino.html.

- [23] «Hyperledger fabric master documentation,» Hyperledger, 2019. [En línea]. Available: <https://hyperledger-fabric.readthedocs.io/en/release-1.4/blockchain.html>.
- [24] «What is a light client and why you should care?,» Parity, 26 Julio 2018. [En línea]. Available: <https://www.parity.io/what-is-a-light-client/>.
- [25] «GitHub - IBM/blockchain-network-on-kubernetes,» [En línea]. Available: <https://github.com/IBM/blockchain-network-on-kubernetes#7-connect-the-network-using-client-sdk>.
- [26] «pySerial 3.4 documentation,» Chris Liechti, 2016. [En línea]. Available: <https://pyserial.readthedocs.io/en/latest/pyserial.html>.
- [27] «A module to control Raspberry Pi GPIO channels,» Python Software Foundation, 21 julio 2019. [En línea]. Available: <https://pypi.org/project/RPi.GPIO/>.
- [28] «Asyncio Documentation,» Victor Stinner, 2016. [En línea]. Available: <https://asyncio.readthedocs.io/en/latest/>.
- [29] «Getting started - websockets 8.1 documentation,» Aymeric Augustin and contributors, 2019. [En línea]. Available: <https://websockets.readthedocs.io/en/stable/intro.html>.
- [30] «Hyperledger Fabric Python SDK,» Hyperledger TSC, 08 setiembre 2016. [En línea]. Available: <https://github.com/hyperledger/fabric-sdk-py>.
- [31] «OCPP 0.2.0 documentation,» Auke Willem Oosterhoff, 2019. [En línea]. Available: <https://ocpp.readthedocs.io/en/latest/README.html>.
- [32] «Building your first network,» Hyperledger Fabric, 2019. [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/build_network.html.
- [33] «Getting started with IBM Blockchain platform for IBM Cloud,» IBM, 07 febrero 2020. [En línea]. Available: <https://cloud.ibm.com/docs/services/blockchain?topic=blockchain-ibp-v2-deploy-iks>.
- [34] «ABB Group,» ABB, 2020. [En línea]. Available: <https://new.abb.com/>.
- [35] «Rittal - Armarios de distribución,» Rittal Disprel S.A. , 2020. [En línea]. Available: <https://www.rittal.com/es-es/content/es/start/>.
- [36] «How to load OpenEVSE Firmware (WinAVR),» OpenEVSE LLC, 2020. [En línea]. Available: [https://openevse.dozuki.com/Guide/How+to+Load+OpenEVSE+Firmware+\(WinAVR\)/7?lang=en](https://openevse.dozuki.com/Guide/How+to+Load+OpenEVSE+Firmware+(WinAVR)/7?lang=en).
- [37] «The Energy Web Chain: Accelerating the Energy Transition with an Open-Source, Decentralized Blockchain Platform,» Energy Web Foundation, Zug, Suiza, 2019.

10. Anexos

10.1. Registro pruebas IEC 62169

Las medidas se realizaron de forma similar a lo descrito en el Capítulo 2.2.2. En ese capítulo se presentaban los resultados para 24A. En este anexo, se muestran mediciones adicionales para otros valores de corriente.

Lo único que se varía es la configuración en el OpenEVSE de la corriente máxima disponible. Se chequea que el Duty Cycle y amplitud de señal cumpla lo especificado por el estándar para cada valor de corriente.

10.1.1. Medidas para 10A

CP conectado: *Estado B - EV conectado*

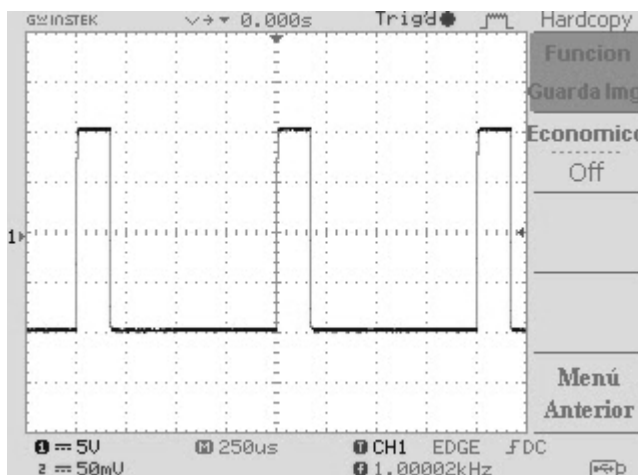


Ilustración 25 – 10A EV conectado

Vehicle ready: *Estado C - EV cargando*

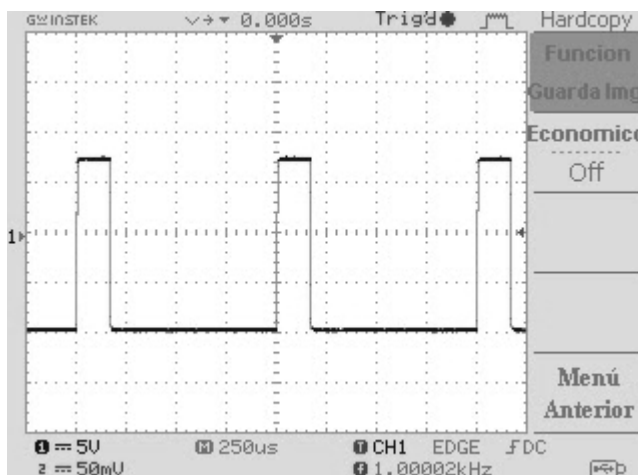


Ilustración 26 – 10A EV cargando

Ventilation required: *Estado D - Ventilación requerida*

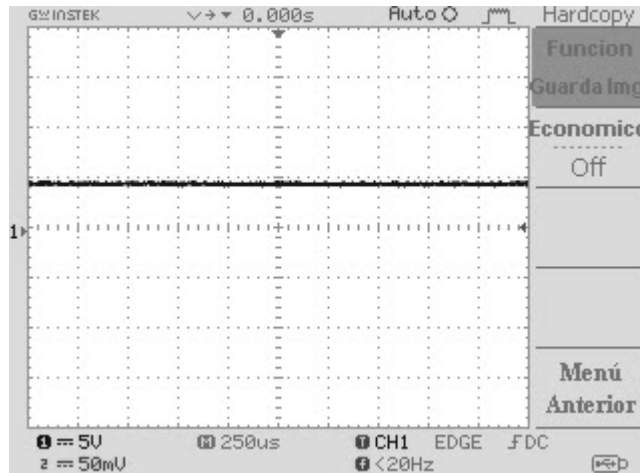


Ilustración 27 – 10A ventilación requerida

10.1.1. Medidas para 30A

CP conectado: *Estado B - EV conectado*

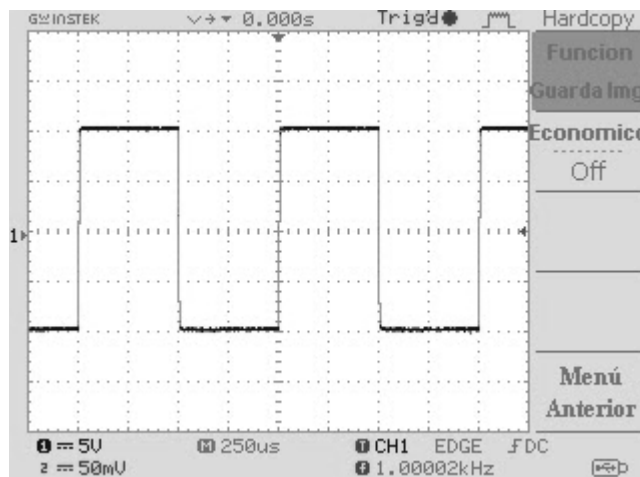


Ilustración 28 – 30A EV conectado

Vehicle ready: *Estado C - EV cargando*

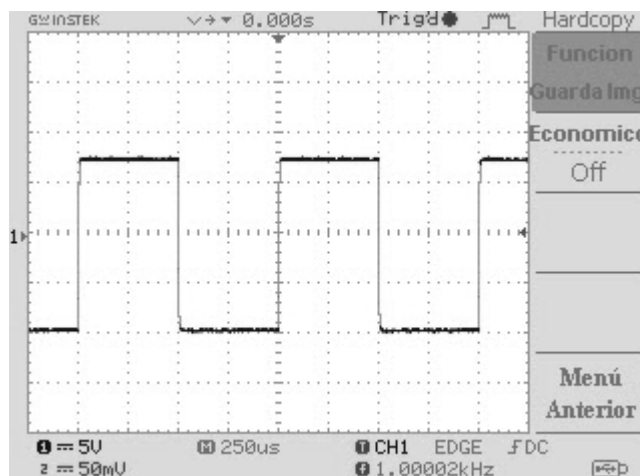


Ilustración 29 – 30A EV cargando

CP desconectado: *Estado A - No conectado*

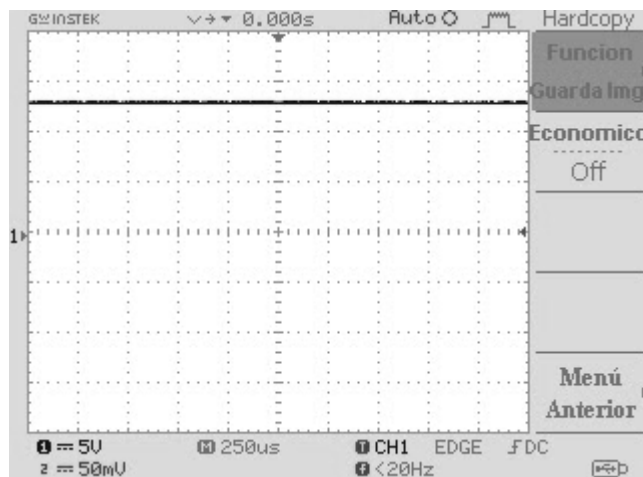


Ilustración 30 – 30A no conectado

Disable: *Estado F - Error*

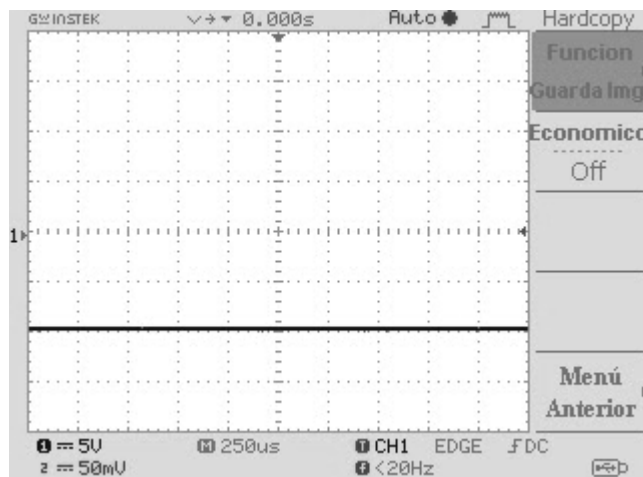


Ilustración 31 – 30A error

10.1.1. Medidas para 60A

CP conectado: *Estado B - EV conectado*

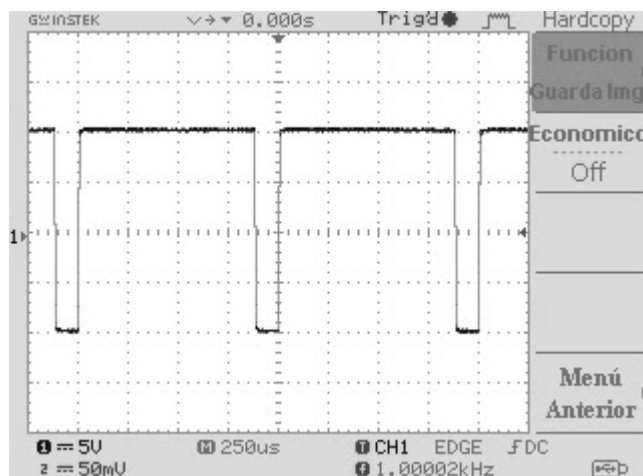


Ilustración 32 – 60A EV conectado

Vehicle ready: *Estado C - EV cargando*

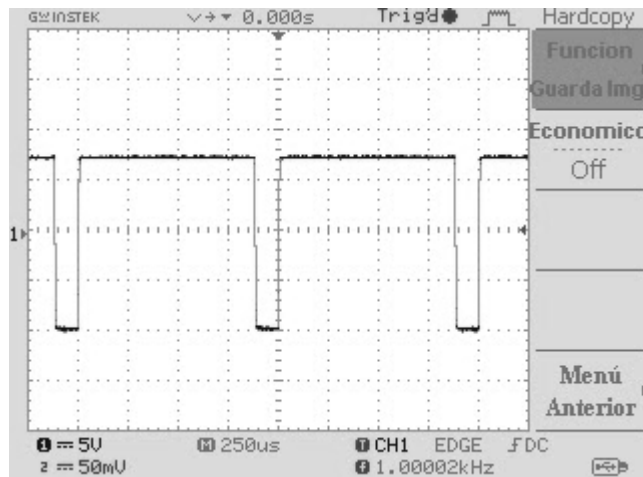


Ilustración 33 – 60A EV cargando

CP desconectado: *Estado A - No conectado*

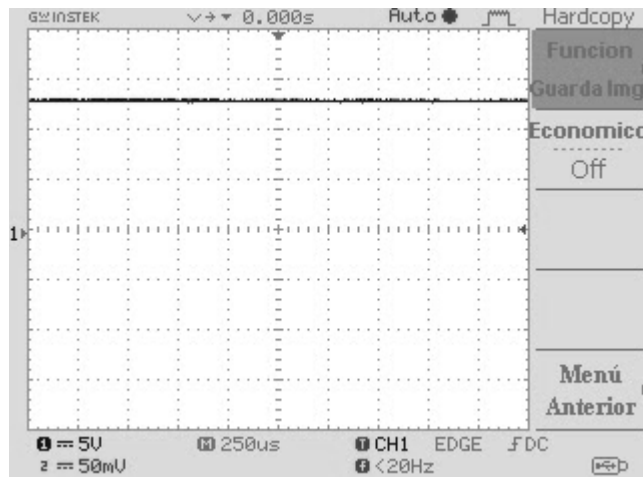


Ilustración 34 – 60A no conectado

Ventilation required: *Estado D - Ventilación requerida*

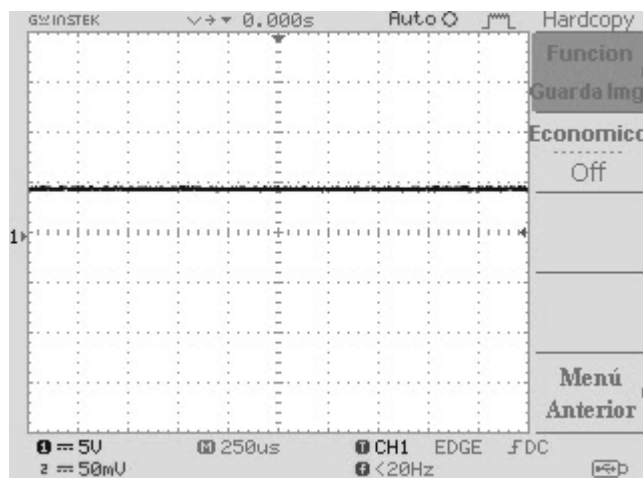
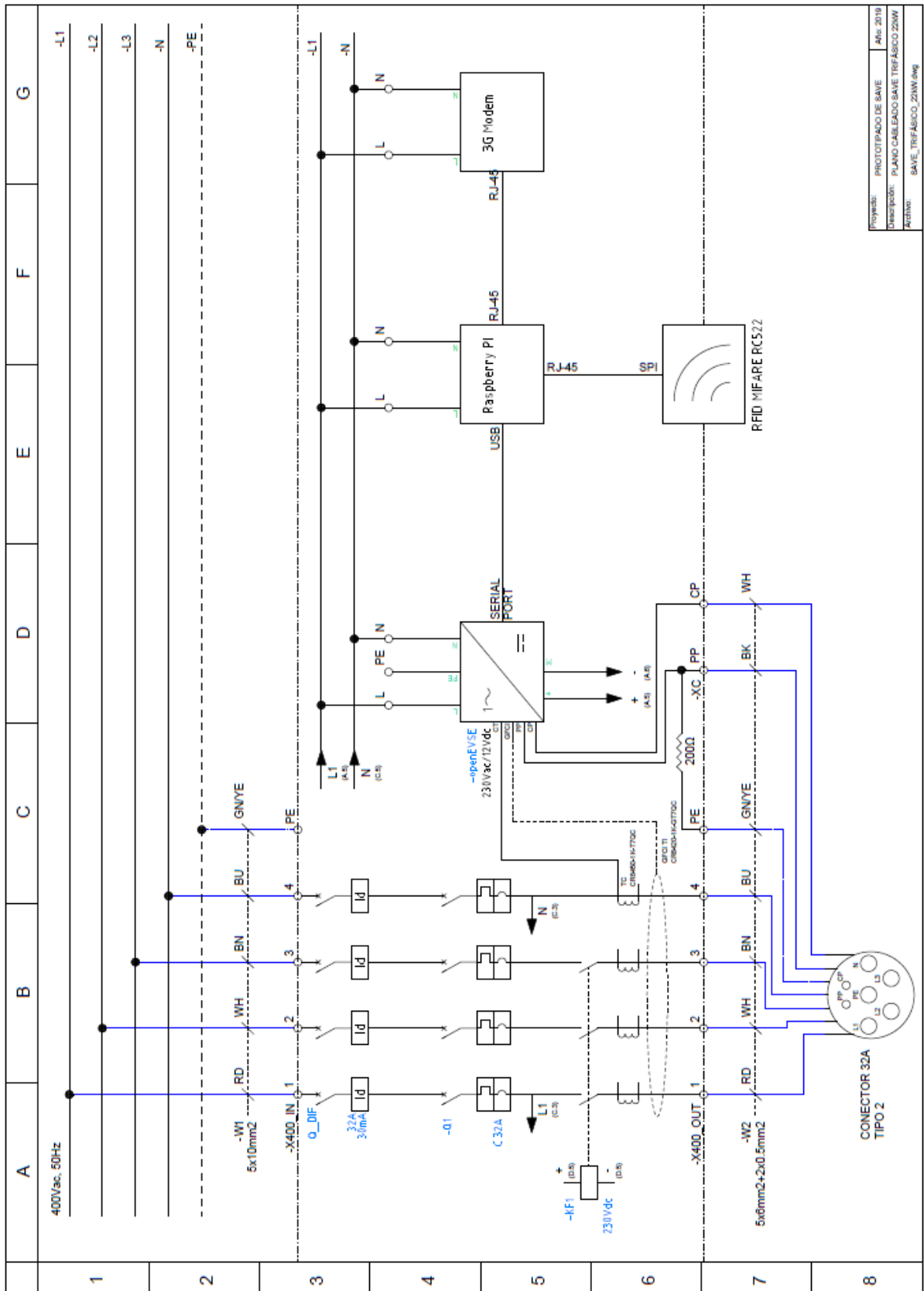


Ilustración 35 – 60A ventilación requerida

10.2. Plano punto de carga trifásico 22 kW



Proyecto: PROTOTIPO DE SAVE | Año: 2019
 Descripción: PLANO CABLEADO SAVE TRIFÁSICO 22kW
 Archivo: SAVE_TRIFÁSICO_22kW.dwg

10.3. Comandos RAPI OpenEVSE

**** RAPI protocol ****

Fx - function

Sx - set parameter

Gx - get parameter

command formats

1. with XOR checksum (recommended)

\$cc pp pp ...^xk\r

2. with additive checksum (legacy)

\$cc pp pp ...*ck\r

3. no checksum (FOR TESTING ONLY! DON'T USE FOR APPS)

\$cc pp pp ...\r

\r = carriage return = 13d = 0x0D

cc = 2-letter command

pp = parameters

xk = 2-hex-digit checksum - 8-bit XOR of all characters before '^'

ck = 2-hex-digit checksum - 8-bit sum of all characters before '*'

response format (v1.0.3-)

\$OK [optional parameters]\r - success

\$NK [optional parameters]\r - failure

response format (v2.0.0+)

\$OK [optional parameters]^xk\r - success

\$NK [optional parameters]^xk\r - failure

xk = 2-hex-digit checksum - 8-bit XOR of all characters before '^'

asynchronous messages

\$ST

state\r - EVSE state transition - sent whenever EVSE state changes

state: EVSE_STATE_XXX

\$WF mode\r - Request client WiFi mode

mode: WIFI_MODE_XXX

(currently very long press (10 sec) of menu btn on OpenEVSE will send WIFI_MODE_AP_DEFAULT

v2.0.1+: 2-hex-digit XOR checksum appended to asynchronous messages

commands

FB color - set LCD backlight color

colors:

OFF 0

RED 1

YELLOW 3

GREEN 2

TEAL 6

BLUE 4

VIOLET 5

WHITE 7

\$FB 7*03 - set backlight to white

FD - disable EVSE

\$FD*AE

FE - enable EVSE

\$FE*AF

FP x y text - print text on lcd display

FR - reset EVSE

\$FR*BC

FS - sleep EVSE

\$FS*BD

S0 0|1 - set LCD type

\$S0 0*F7 = monochrome backlight

\$S0 1*F8 = RGB backlight

S1 yr mo day hr min sec - set clock (RTC) yr=2-digit year

S2 0|1 - disable/enable ammeter calibration mode - ammeter is read even when not charging

\$S2 0*F9

\$S2 1*FA

S3 cnt - set charge time limit to cnt*15 minutes (0=disable, max=255)

SA currentscalefactor currentoffset - set ammeter settings

SC amps [N]- set current capacity

response:

if amps < minimum current capacity, will set to minimum and return \$NK amps

if amps > maximum current capacity, will set to maximum and return \$NK amps

otherwise return \$OK amps

default action is to save new current capacity to EEPROM.

if N is specified, then new current capacity is volatile, and will be

reset to previous value at next reboot

SD 0|1 - disable/enable diode check

\$SD 0*0B

\$SD 1*0C

SE 0|1 - disable/enable command echo

\$SE 0*0C

\$SE 1*0D

use this for interactive terminal sessions with RAPI.

RAPI will echo back characters as they are typed, and add a <LF> character

after its replies. Valid only over a serial connection, DO NOT USE on I2C

SF 0|1 - disable/enable GFI self test

\$SF 0*0D

\$SF 1*0E

SG 0|1 - disable/enable ground check

\$SG 0*0E

\$SG 1*0F

SH kWh - set cCharge limit to kWh

SK - set accumulated Wh (v1.0.3+)

\$SK 0*12 - set accumulated Wh to 0

SL 1|2|A - set service level L1/L2/Auto

\$SL 1*14

\$SL 2*15

\$SL A*24

SM voltscalefactor voltoffset - set voltMeter settings

SO ambientthresh irthresh - set Overtemperature thresholds

thresholds are in 10ths of a degree Celcius

SR 0|1 - disable/enable stuck relay check

\$SR 0*19

\$SR 1*1A

SS 0|1 - disable/enable GFI self-test //removed because same as SF //bhc

\$SS 0*1A

\$SS 1*1B

ST starthr startmin endhr endmin - set timer

\$ST 0 0 0 0*0B - cancel timer

SV 0|1 - disable/enable vent required

\$SV 0*1D

\$SV 1*1E

G3 - get time limit

response: OK cnt

cnt*15 = minutes

= 0 = no time limit

GA - get ammeter settings

response: OK currentscalefactor currentoffset

\$GA*AC

GC - get current capacity range in amps

response: OK minamps maxamps

\$GC*AE

GD - get Delay timer settings //added bhc

response: OK starthour startmin stophour stopmin enabled

\$GD*AF

GE - get settings

response: OK amps(decimal) flags(hex)

\$GE*B0

GF - get fault counters

response: OK gfitripcnt nogndtripcnt stuckrelaytripcnt (all values hex)

maximum trip count = 0xFF for any counter

\$GF*B1

GG - get charging current and voltage

response: OK milliamps millivolts

AMMETER must be defined in order to get amps, otherwise returns -1 amps

VOLTMETER must be defined in order to get voltage, otherwise returns -1 volts

\$GG*B2

GH - get cHarge limit

response: OK kWh

kWh = 0 = no charge limit

GL - get volatile flags //bhc

response: OK flags(hex) //bhc

GM - get voltMeter settings

response: OK voltcalefactor voltoffset

\$GM^2E

GO get Overtemperature thresholds

response: OK ambientthresh irthresh

thresholds are in 10ths of a degree Celcius

\$GO^2C

GP - get temPerature (v1.0.3+)

\$GP*BB

response: OK ds3231temp mcp9808temp tmp007temp

ds3231temp - temperature from DS3231 RTC

mcp9808temp - temperature from MCP9808

tmp007temp - temperature from TMP007

all temperatures are in 10th's of a degree Celcius

if any temperature sensor is not installed, its return value will be 0

GS - get state

response: OK state elapsed

state: EVSE_STATE_XXX

elapsed: elapsed charge time in seconds (valid only when in state C)

\$GS*BE

GT - get time (RTC)

response OK yr mo day hr min sec yr=2-digit year

\$GT*BF

GU - get energy usage (v1.0.3+)

\$GU*C0

response OK Wattseconds Whacc

Wattseconds - Watt-seconds used this charging session, note you'll divide Wattseconds by 3600 to get Wh

Whacc - total Wh accumulated over all charging sessions, note you'll divide Wh by 1000 to get kWh

GV - get version

response: OK firmware_version protocol_version

\$GV*C1

T commands for debugging only #define RAPI_T_COMMMANDS

T0 amps - set fake charging current

response: OK

\$T0 75

*

*/

10.4. Log de carga remota

10/03/2020 10:25:13 (v1.6J)(172.26.63.46) - SendMessage - chargeBoxId: proyectoD2V - msg: [2, "185060", "RemoteStartTransaction", {"idTag":"AB205D23"}]

10/03/2020 10:25:13 (v1.6J)(172.26.63.46) - Enviando mensaje: [2, "185060", "RemoteStartTransaction", {"idTag":"AB205D23"}]

```
10/03/2020 10:25:15 (v1.6J)(172.20.30.82) - Contestacion: proyectoD2V - Mensaje: [
  3,
  "185060",
  {
    "status": "Accepted"
  }
]
```

10/03/2020 10:25:23 (v1.6J)(172.20.30.82) - Authorize: proyectoD2V - idTag: AB205D23

10/03/2020 10:25:23 (v1.6J)(172.20.30.82) - Respuesta: Accepted

10/03/2020 10:25:23 (v1.6J)(172.20.30.82) - Respuesta JSON: [3,"a2c48fc9-214f-42ac-b585-c6edcd79258d", {"idTagInfo":{"status":"Accepted"}}]

10/03/2020 10:25:32 (v1.6J)(172.20.30.82) - StartTransaction: proyectoD2V - conectorId: 1 - idTag: AB205D23 - meterStart: 537 - reservationId: 0 - timestamp: 2020-03-09 18:11:00

10/03/2020 10:25:59 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:26:08 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:26:18 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:26:28 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:26:41 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:26:50 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:26:58 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:27:07 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:27:16 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:27:26 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:27:35 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:27:46 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:27:56 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:28:06 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:28:15 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:28:25 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:28:34 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:28:44 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:28:44 (v1.6J)(172.26.63.46) - SendMessage - chargeBoxId: proyectoD2V - msg: [2, "185060", "RemoteStopTransaction", {"transactionId":1132}]

10/03/2020 10:28:45 (v1.6J)(172.26.63.46) - Enviando mensaje: [2, "185060", "RemoteStopTransaction", {"transactionId":1132}]

10/03/2020 10:28:45 (v1.6J)(172.20.30.82) - Contestacion: proyectoD2V - Mensaje: [3, "185060", {"status": "Accepted"}]

10/03/2020 10:28:54 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:29:04 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:29:13 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:29:23 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:29:32 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:29:42 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:29:44 (v1.6J)(172.20.30.82) - Heartbeat: proyectoD2V

10/03/2020 10:29:52 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:30:02 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:30:06 (v1.6J)(172.26.63.46) - SendMessage - chargeBoxId: proyectoD2V - msg: [2, "185060", "RemoteStopTransaction", {"transactionId":1132}]

10/03/2020 10:30:06 (v1.6J)(172.26.63.46) - Enviando mensaje: [2, "185060", "RemoteStopTransaction", {"transactionId":1132}]

10/03/2020 10:30:06 (v1.6J)(172.20.30.82) - Contestacion: proyectoD2V - Mensaje: [
3,
"185060",
{
"status": "Accepted"
}
]

10/03/2020 10:30:11 (v1.6J)(172.20.30.82) - MeterValues: proyectoD2V - conectorId: 1 - transactionId: 0

10/03/2020 10:30:21 (v1.6J)(172.20.30.82) - Authorize: proyectoD2V - idTag: AB205D23

10/03/2020 10:30:21 (v1.6J)(172.20.30.82) - Respuesta: Accepted

10/03/2020 10:30:21 (v1.6J)(172.20.30.82) - Respuesta JSON: [3,"7503bc8d-a09a-4f36-a6b7-55b4edde2664", {"idTagInfo":{"status":"Accepted"}}]

10/03/2020 10:30:23 (v1.6J)(172.20.30.82) - StopTransaction: proyectoD2V - idTag: - meterStop: 537 - timestamp: 2020-03-09 18:15:52 - transactionId: 1132 - reason: EmergencyStop