ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Grado

PROTOCOLOS OCPP Y OICP EN LA COMUNICACIÓN PARA LA RECARGA DE VEHÍCULOS ELÉCTRICOS
(OCPP AND OICP PROTOCOLS IN THE COMMUNICATION FOR THE RECHARGING OF ELECTRIC VEHICLES)

Para acceder al Título de

Graduado en
Ingeniería de Tecnologías de Telecomunicación

Autor: Ana Blanco Díaz
24 de Julio de 2019

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: Ana Blanco Díaz
Director del TFG: Javier Izquierdo López

Título: "PROTOCOLOS OCPP Y OICP EN LA COMUNICACIÓN PARA LA RECARGA DE VEHÍCULOS ELÉCTRICOS"

Title: "OCPP AND OICP PROTOCOLS IN THE COMMUNICATION FOR THE RECHARGING OF ELECTRIC VEHICLES"

Presentado a examen el día:

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:
Presidente (Apellidos, Nombre): García Arranz, Marta
Secretario (Apellidos, Nombre): García Gutiérrez, Alberto Eloy
Vocal (Apellidos, Nombre): Irastorza Teja, José Ángel

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado
Nº
(a asignar por Secretaría)

Agradecimientos

A toda mi familia, en especial a mis padres, por estar siempre ahí, apoyándome en el día a día y ayudándome a salir de los bucles de desánimo en los que más de una vez me he encontrado. Gracias a vosotros, a lo largo de estos años he aprendido mucho más de mí y de lo que soy capaz de conseguir.

A mi tía Marta, porque sin su ayuda jamás habría tenido la oportunidad de trabajar con gente tan maravillosa en EDP.
A Javier, por estar más seguro de mí que yo misma desde el principio y enseñarme tantas cosas durante estos meses. También a Camino, Agustín, Bernardo, José y todos los que trabajan para que esto siga adelante, por permitirme ser una pequeña parte de este fantástico proyecto que estáis creando y hacerlo siempre tan divertido.

Por supuesto, a todos los compañeros con los que he ido viviendo cada etapa, hemos sufrido juntos, hemos luchado juntos y al final lo hemos conseguido.

A Inés y Alba, porque siempre están, porque somos tres.
A San, por encontrarnos a medio camino y no separarnos nunca más, por nuestros eternos ataques de risa cuando todo parecía ir fatal, por reñirme siempre que era necesario y por conseguir que una tirase de la otra hasta llegar a la meta de la mano.
A Su y Marti, por tener siempre las palabras exactas en el momento exacto, por darme esos chutes de alegría, por estar siempre.
A Luis, por todos esos conciertos cuando peor nos venía, pero mejor nos hacía y (porque es necesario nombraros en la misma frase) a Javi y Dani, los cuatro sabemos que es necesario darle las gracias también a Bruce.

A cada persona que ha aparecido en mi vida, los que ya no están y los que llegaron para quedarse, por ayudarme a aprender tanto.

Resumen

El principal objetivo del presente trabajo será el análisis y estudio del funcionamiento de los protocolos OCPP y OICP.
Dentro del proyecto MoveOn de EDP, se atenderá el protocolo OCPP dado en la comunicación entre el servidor central de la compañía y el punto de recarga instalado.
También, a partir del mismo proyecto, se analizará el protocolo OICP para la recarga de vehículos a nivel europeo, realizándose la conexión entre los puntos de recarga y los servidores correspondientes a través de una aplicación IoT en la nube, así como la interconexión entre plataformas.

Abstract

The main objective of the work will be the analysis and study of the operation of the OCPP and OICP protocols.
Within EDP MoveOn project, the OCPP protocol given in the communication between the company's central server and the installed charging point will be addressed.
Also, from the same project, the OICP protocol for the recharging of vehicles in an european level will be analysed, being the conection between the charging points and the corresponding servers through an IoT application in the cloud, as well as the interconnection between platforms.

Índice general

Índice de figuras

Índice de tablas

Acrónimos

API
Application Programming Interface

APN
Access Point Name

B2B
Business to Business

B2C
Business to Consumer

CCGT
Combined Cycle Gas Turbine

CEE 7
ICEE International Comission on the Rules of the Approval of Electrical Equipment

CHAP
Challenge Authentication Protocol

CPO
Charging Point Operator

EDP
Energías de Portugal

EMP
Electric Mobility Provider

EMS
Energy Management System

EV
Electric Vehicle

GEI
Gases de Efecto Invernadero

HTTPS
Hypertext Transfer Protocol Secure

ID
Identificador

IoT
Internet of Things

IP
Internet Protocol

JSON
JavaScript Object Notation

LAN
Local Area Network

MD5
Message-Digest Algorithm 5

OCA
Open Charge Alliance

OCPP
Open Charge Point Protocol

OCPI
Open Charge Point Interface

OICP
Open InterCharge Protocol

PAP
Password Authentication Protocol

PC
Personal Computer

PdR
Punto de Recarga

PDU
Protocol Data Unit

QR
Quick Response

REST
Representational State Transfer

RFC
Request For Comments

RFID
Radio Frequency Identification

RPC
Remote Procedure Call

SaaS
Software as a Service

SIM
Subscriber Identify Module

SSL
Secure Sockets Layer

TCP
Transfer Control Protocol

TLS
Transport Layer Security

URL
Uniform Resource Locator

V2G
Vehicle-to-Grid

VPN
Virtual Private Network

XML
Extensible Markup Language

W3C
World Wide Web Consortium

Capítulo 1: Introducción

La conciencia y preocupación que hoy en día existe en cuanto a la contaminación puede verse en todo tipo de áreas: laborales, negocios, gobiernos, procesos industriales y especialmente en la moralidad de la sociedad en general. Esta filosofía defiende una actitud que se preocupa por el impacto de nuestras acciones en el medio, la conservación de este y la mejora de su estado actual para lograr una mejor calidad de vida. Este movimiento ha ido adquiriendo mayor importancia en los últimos años debido a que se ha podido ir constatando científicamente que la acción del ser humano tiene consecuencias negativas directas que afectan a la integridad del entorno y los seres que lo habitan. Por ello, el concepto de sostenibilidad está adquiriendo cada vez más relevancia en los ámbitos empresariales. La habilidad para satisfacer las necesidades actuales de los clientes sin comprometer las capacidades de las generaciones futuras se ha convertido en una cuestión primordial a la hora de ofrecer nuevos servicios. A raíz de esto, aparece el concepto de "Responsabilidad Social", el compromiso de una organización de asumir las consecuencias que sus decisiones y actividades puedan tener en la sociedad y en el medio ambiente.

No es desconocido que la principal fuente artificial que genera gases contaminantes es la quema de combustible fósil; y esto, unido a que hoy en día alrededor del 42% de la energía final de nuestro país deriva del consumo en carretera, la cual además genera el 24% de los *GEI (Gases de Efecto Invernadero)*, establece la necesidad de promover formas de negocio más afines al desarrollo sostenible. Se puede verificar que la preocupación por el cambio climático va unida a la evolución y aparición de nuevas tácticas empresariales y producción de servicios, lo que también ha afectado a la transformación del sector del transporte. Debido a este gran número de cambios vemos que actualmente oír hablar del vehículo eléctrico es cada vez más común. Las compañías fabricantes de vehículos empiezan a disponer de modelos con estas características entre sus ejemplares con mayor frecuencia, debido por supuesto a la percepción social de la contaminación, pero también a los grandes avances que se han alcanzado en cuanto a su fabricación y capacidades. La aparición de las baterías de litio ha supuesto una revolución en cuanto a la autonomía que ahora este tipo de vehículos son capaces de alcanzar. El aumento de la capacidad de potencia de estas, hace que sea posible incluso comparar las competencias de estos vehículos con los modelos de gasolina. Estas baterías son más estables, por lo que la vida útil de las mismas es mayor y además también son más fiables. Otra de sus características principales es que no es necesario descargarlas completamente para poder volver a recargarlas, lo cual facilita la experiencia como usuario del vehículo eléctrico. Por lo tanto, la actividad empresarial que rodea estas competencias se centra ahora en la implantación de *PdR (Puntos de Recarga)* de vehículos eléctricos para fomentar su uso e impulsar una forma de movilidad acorde a las preocupaciones y exigencias de la sociedad.

1.1 Contexto

En el presente trabajo se tratarán los conceptos previos y se analizará cómo están afectando actualmente a la población y la movilidad en España, relacionándolo con el estudio del desarrollo del proyecto *EDP MoveOn* de la compañía *EDP (Energías de Portugal).*

Como una de las mayores organizaciones eléctricas de Europa y la mayor de Portugal, *EDP* impulsa el concepto de movilidad sostenible, estrechamente unido a la gran evolución de la fabricación de vehículos eléctricos. Tras la puesta en marcha de la primera *Ecoestación* (*estación de servicio que provee energías alternativas*) en el norte de España, con el plan *EDP MoveOn, EDP* pretende instalar en distintas localizaciones

de la geografía española y de Europa, *PdR* eléctricos destinados a todo tipo de usuarios. Para ello diferencia los siguientes casos:

- Tipos de usuarios:
  - Usuarios de *EV* (*Electric Vehicle*) y cliente de *EDP*.
  - Usuarios de *EV* y cliente de una compañía distinta.

- Tipos de recargas:
  - Recarga del *EV* en instalación privada.
  - Recarga del *EV* en infraestructura pública de *EDP*, en España o en Europa.

En base a los tipos de usuarios, se pueden distinguir varias maneras de atención a los clientes que, dependiendo de su condición, dispondrán de unos privilegios u otros. Estos conceptos conciernen más bien a asuntos relacionados con el mundo empresarial, el estudio de las ganancias obtenidas en cada caso y los beneficios finales de la compañía. El actual documento se centrará esencialmente, dentro de los tipos de recargas, en la descripción del último punto: la recarga de *EV* utilizando infraestructura pública de *EDP*. Se describirán los diferentes *PdR* disponibles y sus características como conceptos incluidos en la explicación del proyecto *EDP MoveOn*; que a su vez servirá como base para la inmersión en la operatividad y el funcionamiento de los dispositivos que intervienen. Una vez introducidos los aspectos anteriores, se centrará la atención en el modo de comunicación e intercambio de paquetes que se da tras una solicitud de recarga y más concretamente en la especificación de los protocolos que intervienen en cada ejemplo.

1.2 Objetivos

El principal objetivo será el estudio de los protocolos de comunicación protagonistas en el intercambio de información entre el *PdR* y el servidor o la plataforma encargada tras una solicitud de recarga utilizando la infraestructura pública desplegada, relacionando su operatividad con las funcionalidades disponibles en la aplicación *EDP MoveOn*. Diferenciando los anteriormente mencionados tipos de recarga en infraestructura pública de *EDP*, también se distinguirán dos protocolos de comunicación principales para la ejecución de la recarga del vehículo una vez solicitada por el usuario.

En base al primer tipo de recarga: la solicitud de carga eléctrica dentro del territorio español, la comunicación tiene lugar entre el mismo *PdR* y un servidor central alojado en la nube que es gestionado por *EDP,* el cual a su vez transmite datos e información sobre las recargas a un servidor que funciona como *Base de Datos*. El intercambio de mensajes entre estas entidades se realiza generalmente mediante el protocolo *OCPP (Open Charge Point Protocol)* versión *1.6*, aunque ya existe una nueva actualización, *OCPP 2.0*. El *Protocolo Abierto de Punto de Carga* es un protocolo de aplicación para la comunicación directa entre los puntos físicos de recarga de *EV* y el correspondiente sistema central de gestión. Con la característica de protocolo abierto se pretende la implantación de un sistema de comunicación que permita el intercambio de mensajes entre entidades que provienen de diferentes compañías o fabricantes. Este concepto sirve de ayuda para introducir el segundo tipo de recarga, ocasión en la que es más probable que intervengan dispositivos de distintas entidades: la solicitud de recarga eléctrica en algún punto de Europa, o por parte de un usuario extranjero en un *PdR* en España. En estos casos, la comunicación se establece entre el *PdR* de *EDP* instalado y la plataforma *eRoaming Hubject*. En esta ocasión, el protocolo de comunicación protagonista dado entre los extremos es el protocolo *OICP (Open InterCharge Protocol)*. Este protocolo es actualmente el más extendido en Europa para la comunicación entre los *Puntos de Recarga* y los proveedores de movilidad eléctrica. Se introducirá también

una descripción de la plataforma *IoT (Internet of Things) Hubject* y de su funcionamiento, para el posterior análisis del protocolo *OICP* en su versión *2.2*.

1.3 Estructura

El presente documento se organiza en seis capítulos, divididos a su vez en varios subcapítulos en los que se tratarán cuestiones específicas de cada uno. Al comienzo de cada capítulo se incluye una introducción sobre el tema a exponer.

En el cuerpo del trabajo, a lo largo del capítulo segundo, se presenta la compañía *EDP*, evaluando sus características y actividades y enfocando las principales en el marco de desarrollo de este proyecto. Se expone el plan *EDP MoveOn* para comenzar a tratar el concepto de movilidad sostenible. En el mismo capítulo se consideran también los distintos tipos de infraestructuras de recarga de las que *EDP* dispone, explicando las características de cada cargador, así como los diferentes tipos de conectores que existen y las diferencias entre ellos. Una vez teniendo lo anterior en cuenta, se explica el funcionamiento de la aplicación móvil *EDP MoveOn*, haciendo una pequeña introducción de la empresa encargada de su programación e incluyendo ciertos ejemplos de la misma en el sistema operativo *iOS*.
El tercer capítulo se centra en las características y protocolos involucrados en las comunicaciones entre los cargadores y el servidor encargado durante las solicitudes por parte de los clientes, el proceso de recarga del vehículo y el funcionamiento de la red desde el punto de vista de los datos intercambiados.
En el capítulo cuatro se comienza ya a tratar el protocolo de aplicación *OCPP*, comenzando por una pequeña introducción sobre su historia y orígenes, y pasando luego a tratar las diferentes versiones del mismo, así como las diferentes acciones que se puede llevar a cabo con cada versión, organizando las *PDUs* en tablas.
El quinto capítulo trata el protocolo descrito en el capítulo cuatro mediante ejemplos. Utilizando el programa Wireshark para el trabajo, se ha capturado la transferencia de paquetes entre un *Punto de Recarga* especialmente configurado para ello y el servidor en comunicaciones *OCPP*. Se han realizado las tres tareas más comunes que los usuarios de *EV* llevan a cabo, esto es: reserva del cargador, recarga del vehículo y detención de la recarga. Asimismo, se incluyen tres ejemplos más de operaciones entre *PdR* y servidor utilizando *OCPP*: *Heartbeat, BootNotification* y *MeterValues*.
En el último capítulo se hace una pequeña descripción de cómo se gestiona la recarga de vehículos eléctricos a nivel internacional mediante el protocolo *OICP* de la plataforma *Hubject,* de la que también se incluye una breve referencia. Para terminar, se nombra el protocolo *OCPI* , surgido a partir del anterior y que está teniendo cada vez más repercusión.

Capítulo 2: EDP – EDP MoveOn

En este capítulo se realiza una introducción a lo que es el grupo *EDP* nacional e internacionalmente y a sus puntos de operación. A partir de los mismos, tratando los conceptos de innovación y nuevos propósitos, se entrará a hablar del servicio de movilidad, donde se presentan más concretamente los principios de *EDP MoveOn*, que resulta de necesaria mención para la compresión del desarrollo del trabajo.

2.1 EDP – Energías de Portugal

EDP [1] es una compañía que se encarga de la generación, distribución y comercialización de energía eléctrica, gas y servicios; área de gran importancia para el desarrollo económico y social. Centrando su actividad principal en este campo, produce electricidad mediante energía hidráulica y eólica, carbón, energía nuclear, *CCGT (Combined Cycle Gas Turbine – Turbina de Gas de Ciclo Combinado)* y por medio de cogeneración y residuos. En el año 2017 generó 16.439 GWh. La energía total distribuida alcanzó los 9.331 GWh, repartidos a través de una red de 20.613 km que cubre todo el territorio español. A su vez, la cantidad total de gas distribuido llegó hasta los 15.014 GWh. Estas cifras convierten a *EDP* en una de las empresas energéticas líderes tanto en nuestro país como fuera de España; y como tal, centra su misión en tres objetivos fundamentales: la creación de valor, la orientación de sus actividades a los clientes y un enfoque en el potencial humano para lograr ser la compañía más competitiva de su sector. Para satisfacer estos puntos realiza su cometido de forma transparente, respetando el entorno y ayudando al desarrollo de las regiones en las que se encuentra presente.

Figura 2.1 – Logo EDP.

Trabaja de forma global defendiendo políticas afines a procesos industriales sostenibles, asumiendo las responsabilidades sociales y sobre todo medioambientales que puedan aparecer de forma directa o indirecta como consecuencia de sus actividades de producción. Gracias a estos métodos de trabajo ha conseguido ya evitar la emisión de 268 kt de CO_2. Centrándose en su actividad en España, todas las características anteriores han posicionado a *EDP* como una de las compañías más

influyentes en innovación y nuevos proyectos, siendo líderes en la gestión ambiental de negocios. Considerando que es posible crear y ofrecer valor a largo plazo mediante sus propósitos e ideales, actualmente ha apostado por avanzar en el plano de la movilidad sostenible. Aprovechando el compromiso de España con la Unión Europea, en base al acuerdo de que para el año 2020 deberían circular alrededor de 300.000 vehículos eléctricos en el país, la compañía eléctrica está desarrollando un programa para fomentar su uso basándose en la oferta de un numeroso despliegue de *Puntos de Recarga* accesibles a todo tipo de usuarios. La aceptación de este significativo cambio en el concepto de transporte y movilidad se irá produciendo de manera paulatina, aunque en 2025 debería haber entre uno y dos millones de vehículos eléctricos, aumentando hasta los seis millones para el año 2030. Esta nueva concepción de lo que serán los medios de transporte en el futuro significa para *EDP* una muy buena oportunidad para el desarrollo de las energías alternativas en este sector, y por tanto la perfecta ocasión para implantar *Puntos de Recarga* por toda España.

Como iniciación a esta nueva evolución, la compañía ha establecido un plan estratégico y estudio previo; esta táctica empresarial puede dividirse en cuatro puntos principales: regulación, flota propia y empleados, infraestructuras y clientes.

2.2 EDP MoveOn

Organizado en los cuatro puntos mencionados anteriormente, *EDP MoveOn* es el proyecto mediante el cual se pretende impulsar la movilidad *EDP*. Para poder encontrarse involucrada en los diferentes marcos que constituirán la regulación del proyecto europeo, la empresa ha estado participando de manera activa en foros, asociaciones, grupos de trabajo, patronales y administraciones de la *UE*; impulsando también iniciativas tanto estatales como a nivel europeo e interviniendo a su vez en la organización de planes municipales. Asimismo, para potenciar tanto el programa como su ideología dentro de la misma compañía, ha lanzado planes y promociones para sus empleados, generando una flota propia de vehículos eléctricos para reducir la emisión de gases contaminantes y con el fin último de realizar una conversión completa de la misma. También, para animar a sus proveedores a sumarse a la movilidad sostenible ha puesto en marcha el concepto de “car sharing”, con el objetivo de reducir el uso del vehículo propio. Enfocándose en los clientes, ha establecido nuevas relaciones con los mismos, tanto con sus clientes *B2B (Business To Business)*, es decir negocios que se sirven de la oferta de servicios de *EDP*, como con los *B2C (Business To Consumer)*, los clientes individuales. Para poder orientar de la manera más eficaz posible el nuevo modelo de sostenibilidad con los mismos, ha desarrollado nuevos ciclos comerciales, modos de operación, o establecido plataformas de interoperabilidad para conseguir un trato más eficiente. Para tener aún más poder de incidencia en el mercado y en el impulso de la evolución hacia lo sostenible, se encuentra en pleno desarrollo de diversos acuerdos con distintos fabricantes de vehículos, ofreciendo la posibilidad de operación conjunta de la plataforma de movilidad *EDP* y las nuevas características de navegación disponibles en los vehículos eléctricos de fabricación actual. Igualmente lo hace con empresas de gestión de flotas para el apoyo al desarrollo sostenible de las ciudades en las que tiene una gran presencia. Como punto clave en el establecimiento de este nuevo proyecto está la tarea de implantación de *Puntos de Recarga*. La instauración de una red de infraestructura pública para la recarga de vehículos eléctricos es el punto central dentro del trabajo de *EDP* en su proyecto *EDP MoveOn*.

De esta forma, es actualmente la compañía que más *PdR* eléctricos ha instaurado en España en el año 2018, llegando a los 81 puntos en total. La empresa ha sido capaz de alcanzar este objetivo gracias a diversos convenios establecidos con los ayuntamientos de las regiones donde se han implantado *PdR*, y también con diferentes estaciones de servicio que poseen *Puntos de Recarga EDP* dentro de sus propiedades. Aparte de estos acuerdos, también ha querido posicionarse en la recarga de *EV* en el

marco europeo, trabajando para ofrecer sus servicios tanto fuera de la península a cualquier usuario que solicite el servicio, como dentro de la misma a usuarios extranjeros. Ahora la tarea se centra en el desarrollo, homogenización e instalación de *Puntos de Recarga* propios, para consagrarse como una de las compañías pioneras en su instauración.

A continuación, como precedente a la inmersión en la forma estructural del sistema de comunicación y el funcionamiento del proyecto, se presentan los diferentes *Puntos de Recarga* disponibles que *EDP* ofrece a sus usuarios.

2.2.1 Infraestructura

La infraestructura de recarga ofertada se divide principalmente en dos grupos: infraestructura pública e infraestructura privada. Aunque se explicarán ambas, el presente trabajo se centrará en la infraestructura pública, pues es en la que intervienen los protocolos de comunicación en los que se basa esta memoria. Para el análisis de los tipos de *Puntos de Recarga* ofrecidos también será necesaria la definición de ciertos aspectos técnicos que caracterizan a los mismos.

2.2.1.1 Tipos de conectores

Existen diversos tipos de conectores para la recarga de vehículos eléctricos actualmente en el mercado; a continuación, se describen sus principales características:

- Conector Schuko: Se trata del tipo de conector más común en Europa, por lo que se encuentra en la mayoría de los dispositivos eléctricos domésticos. Se define en el estándar *CEE 7 (ICEE – International Commission on the Rules of the Approval of Electrical Equipment)*. Este tipo de enchufes tienen un diseño sin polarizar, es decir, fase y neutro pueden ser invertidos; además la toma de tierra siempre está activada. En general, se utilizan para valores de tensión de 230 V y de 16 A de intensidad; por lo que la potencia soportada estaría alrededor de los 3,7 kW. En el ámbito que se trata, estos conectores se utilizan para la recarga de bicicletas y motocicletas eléctricas; y algún modelo de pequeño tamaño de coche eléctrico.

Figura 2.2 – Conector de tipo Schuko (izquierda). Base de conector Schuko (derecha).

- Conector Tipo 1 – SAE J1772: Estándar aceptado para la recarga de vehículos eléctricos en corriente alterna. Compuesto por cinco bornes, su conectividad es igual que una toma de corriente monofásica: haciendo referencia al cableado, está formado por fase, neutro y tierra; los otros dos conectores complementarios sirven para la comunicación con el vehículo. Posee además una protección extra que permite el bloqueo del conector para evitar la desconexión inesperada. Se diferencian dos escalas de recarga: recarga lenta, de hasta 16 A, y recarga rápida, que alcanzaría hasta los 80 A.

Figura 2.3 – Conector SAE J1772 (izquierda). Base de conector SAE J1772 (derecha).

- Conector Tipo 2 – IEC 62196-2: Este es el modelo de conector homologado como estándar europeo. Trabaja con corriente alterna y, aunque es muy parecido al conector Tipo 1, este tipo permite tanto cargas monofásicas como trifásicas. Debido a esto, incorpora dos bornes más, correspondientes a las fases complementarias que permiten cargar en trifásico, alcanzando casi los 44 kW. En su modalidad monofásica estos cargadores poseen una potencia de 7,4 kW.

Figura 2.4 – Conector IEC 62196-2 (izquierda). Base de conector IEC 62196-2 (derecha).

- Conector Tipo 3 – Scame: Presenta las mismas características que los conectores descritos anteriormente, pero hoy en día está en desuso. Se pueden diferenciar dos tipos: 3A, para cargas monofásicas de hasta 16 A; y 3C, para cargas trifásicas de 32 A. La máxima potencia ofrecida es de 22 kW.

Figura 2.5 – Conector Scame (izquierda). Base de conector Scame (derecha).

- Conector CHAdeMO: Este conector tiene diez bornes, comunicación y toma de tierra. Trabaja con corriente continua y está diseñado para realizar lo que se denominan recargas rápidas y ultra-rápidas, aquellas que alcanzan los 50 kW de potencia, aunque este tipo de conector puede proporcionar hasta 62,5 kW.

Figura 2.6 – Conector CHAdeMO (izquierda). Base de conector CHAdeMO (derecha).

- Conector CCS – conector único combinado: Consta de cinco bornes, para flujo de corriente, toma de tierra y comunicación con el vehículo. Su nombre es debido a que con este tipo de conector es posible recargar tanto en corriente alterna como en continua. El conector AC es un conector Tipo 2 – IEC 62196-2, descrito anteriormente, y con el que son admisibles hasta 44 kW; por otra parte, el conector DC alcanzaría una potencia de 100 kW, aunque actualmente solo es tolerable trabajar con cantidades alrededor de los 50 kW de potencia.

Figura 2.7 – Conector CCS (izquierda). Base de conector CCS (derecha).

2.2.1.2 Tipos de cargadores

En función de los tipos de conectores descritos en el apartado anterior, *EDP* oferta los siguientes tipos de cargadores [2], de los que se describen principalmente sus características técnicas de funcionamiento:

- Infraestructura privada: Para la recarga de *EV* de forma particular se encuentra el cargador *EDP KEY C16230-3, 6-TII*. Se trata del modelo de carga homologado en Europa, es decir, posee el conector de Tipo 2 – IEC 62196-2, cargando en monofásico con una tensión de 230 V y una corriente de 16 A, alcanzando así los 3,7 kW. Dado que este modelo está diseñado para la recarga del vehículo de forma personal, no dispone de programación ni protocolos que intervengan para comunicar el dispositivo con un servidor central externo de la empresa; incluye una llave on/off para la activación opcional del mismo.

Figura 2.8 – Cargador EDP KEY C16230-3, 6-TII.

- Infraestructura pública: Para la recarga de los vehículos eléctricos utilizando la infraestructura dispuesta por *EDP* en distintos puntos, se pueden establecer dos grupos basándose en la potencia máxima ofrecida por cada uno: de carga semi-rápida y de carga rápida:

    - Carga semi-rápida: Grupo en el que se encuentran los cargadores capaces de alcanzar potencias de hasta 22 kW. Existen dos modelos, *EDP Eve Mini* y *EDP Pro*. Ambos presentan conectividad con el centro de gestión mediante los protocolos de comunicación *OCPP* y *OICP*; y comunicaciones *LAN (Local Area Network)* y 3G. Disponen de lectores de tarjetas *RFID (Radio Frequency Identification)* para poder solicitar recargas mediante tarjetas de fidelización personalizadas.

Figura 2.9 – Cargador EDP Eve Mini (izquierda). Cargador EDP Pro (derecha).

    - Carga rápida: donde se encuentran el cargador *EDP QC45* y el *EDP QC24*. El primero llega hasta los 50 kW de potencia en corriente continua, posee triple salida, con diferentes tipos de conectores en cada una y también tiene diseñada una integración con el centro de control. El cargador *EDP QC24* se encuentra en este grupo porque es capaz de llevar la recarga del vehículo del 0% al 80% en menos de una hora, aunque sólo alcanza una potencia de 24 kW en DC.

Figura 2.10 – Cargador EDP QC24.

Figura 2.11 – Cargador EDP QC45 (Izquierda). Conectores integrados en el cargador EDP QC45 (derecha).

En ambos casos, los cargadores disponen de programación para establecer conexiones mediante los protocolos *OCPP* y *OICP*, dependiendo de la comunicación que se deba establecer con el servidor central que intervenga en cada tipo de recarga; como se ha comentado anteriormente, recarga en infraestructura pública de *EDP* por clientes de la compañía en territorio español, o recarga en infraestructura pública de *EDP* fuera de España por cualquier tipo de usuario, o dentro en España por usuarios extranjeros. Además, al igual que los cargadores semi-rápidos, incorporan conexiones *LAN*, lector de tarjetas y conectividad 3G.

En general, los *EV* incorporan en su interior un rectificador para transformar la corriente alterna recibida de la red eléctrica en corriente continua, adaptada para la recarga de la batería. Estos rectificadores difieren dependiendo del modelo de vehículo y su potencia máxima alcanzable, es por esto por lo que, en ocasiones, es conveniente que el *PdR* proporcione corriente continua directamente, lo cual se da normalmente en los cargadores anteriormente denominados de carga rápida.

Para la utilización de estos *Puntos de Recarga*, *EDP* ha desarrollado, junto a la empresa *O2E – Oxígeno Empresarial*, una página web y una aplicación móvil para la recarga de *EV*.

2.3 Aplicación móvil

Previamente a la descripción del funcionamiento de la aplicación móvil, se realiza una pequeña introducción a la empresa *O2E – Oxígeno Empresarial*, encargada de la creación y de la programación tanto de esta como de los *PdR*.

2.3.1 O2E – Oxígeno Empresarial

Se trata de una pequeña empresa ubicada en el Parque Tecnológico de Gijón que centra su actividad en el sector de Tecnologías de la Información y Servicios. Se especializa en el desarrollo software para la gestión, mantenimiento e inspección de activos e instalaciones, combinando ingeniería y software.
Una de sus tareas actuales es el *SaaS (Software as a Service – Software como Servicio)* para la gestión de recarga de vehículos eléctricos; en el proyecto que se trata, se han encargado de la correcta programación y mantenimiento de los *PdR*, así como de la creación de una página web y una aplicación móvil, disponible para *Android* e *iOS*. Estas dos plataformas sirven al cliente como acceso para la utilización de los *Puntos de Recarga*, accediendo mediante sus credenciales para realizar la recarga de su vehículo. Ambas se han programado acorde a las funcionalidades que las versiones de los protocolos de comunicación utilizadas entre los *PdR* y los servidores son capaces de soportar, como por ejemplo la reserva de un determinado *Punto de Recarga* sin necesidad de encontrarse en la localización exacta en ese momento, o la visualización del precio de la recarga en pantalla antes de que el usuario la inicie.

A continuación, se exponen las características de la aplicación móvil para el sistema operativo *iOS*, para indicar las funcionalidades disponibles para el usuario; para los usuarios de *Android*, las funcionalidades son las mismas.

2.3.2 EDP MoveOn – Aplicación móvil en iOS

Como se ha comentado anteriormente, para la utilización de los cargadores desplegados por *EDP*, los usuarios pueden utilizar una tarjeta *RFID* asociada, la página web o la aplicación móvil, en este apartado se describe esta última.

Figura 2.12 – Pantalla inicial de la aplicación móvil en iOS.

En la pantalla de entrada a la aplicación [Figura 2.12] aparecen dos pestañas para acceder a la enumeración de los *Puntos de Recarga* instalados, que puede visualizarse en forma de lista, ordenada en función de la cercanía al usuario, o en vista de mapa, en la que también aparece la geolocalización del cliente. También está la pestaña de inicio de sesión, para introducir las credenciales de acceso, la información legal y normas de la aplicación, y un tutorial de utilización tanto de esta como de los *PdR* en cuanto a la conexión de los vehículos a estos para realizar una recarga.
Una vez el cliente ha accedido con su usuario y contraseña, aparecerán dos nuevos accesos en la pantalla de inicio: uno será para visualizar un listado del histórico de recargas realizado, en el que aparecerán el *Punto de Recarga* utilizado, el día en que se ha efectuado la recarga y los kWh consumidos en esta; el segundo será un acceso al perfil del usuario, en el que aparecen su nombre y apellidos, su *Login*, que normalmente será su correo electrónico y su número de tarjeta *RFID* asociada.
Actualmente existen *Puntos de Recarga* instalados en Madrid, Valencia, País Vasco y Cantabria, aunque la mayoría se concentran en Asturias.

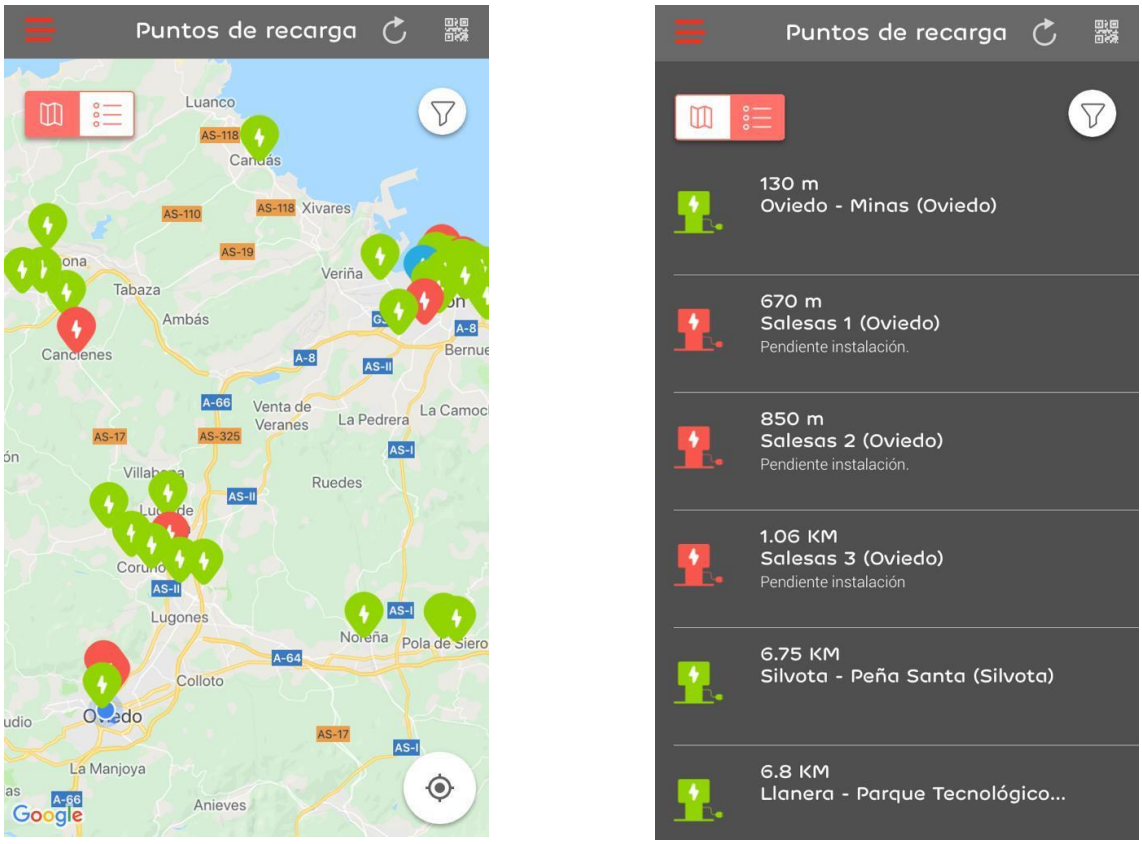

Figura 2.13 – Pantallas con diferentes cargadores instalados: visualización en mapa (izquierda), visualización en lista (derecha).

En las imágenes superiores [Figura 2.13] se pueden ver los *Puntos de Recarga* instalados, así como su estado en tiempo real. El estado de los *PdR* viene dado por el color en el que aparece su icono en la aplicación; se diferencian tres colores:

- Verde: si el *PdR* se encuentra activo y disponible para su uso.
- Azul: si el *PdR* se encuentra activo, pero está siendo utilizado por otro cliente en ese momento.

- Rojo: si el *PdR* se encuentra inactivo, fuera de servicio, ya sea porque se ha perdido la conexión con este, está pendiente de instalación o debido a que el equipo de mantenimiento está realizando cambios o actualizaciones de software.

En función del tipo de vehículo del que disponga, el usuario deberá utilizar un determinado tipo de cargador, con uno de los tipos de conectores descritos anteriormente y atendiendo a la potencia que el *Punto de Recarga* es capaz de ofertar. En la aplicación móvil existen filtros de búsqueda que facilitan al usuario la tarea de encontrar el *PdR* adecuado en función de su localización:

Figura 2.14 – Pantalla de filtros de búsqueda de PdR.

Como se ve en la imagen anterior [Figura 2.14], se pueden buscar *Puntos de Recarga* según su disponibilidad, su potencia y los tipos de conectores que posee:

Figura 2.15 – Filtro de búsqueda en función del tipo de conector.

Figura 2.16 – Filtro de búsqueda en función de la potencia en la recarga (izquierda), filtro de búsqueda en función del estado del cargador (derecha).

Al seleccionar cualquiera de los *PdR* se abre una nueva ventana en la que se muestran cuántos, y de qué tipos de conectores dispone, el estado de cada uno de ellos y la potencia de recarga que son capaces de alcanzar, así como el precio del kWh en la recarga:

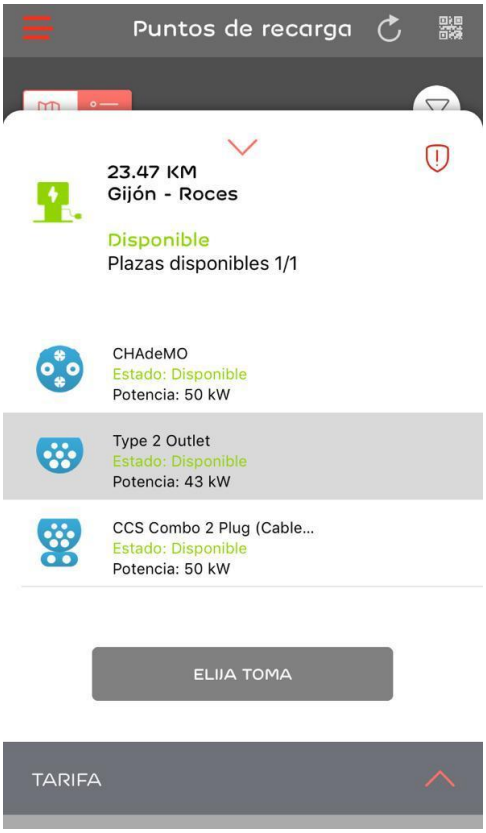

Figura 2.17 – Características del Punto de Recarga seleccionado.

Por ejemplo, en la imagen superior [Figura 2.17] aparecen las características del *PdR* cuyo nombre comercial es “Gijón – Roces”: se encuentra a 23,47 km de distancia del usuario y posee una plaza para recargar un vehículo eléctrico, con tomas CHAdeMO, tipo 2 o CCS.

Si de las anteriores imágenes se observan las respectivas al listado completo de cargadores [Figura 2.13] o la correspondiente a las características de un *Punto de Recarga* en concreto [Figura 2.17], se puede apreciar que en la esquina superior derecha aparece un pequeño icono que representa un código *QR (Quick Response)*. Cada *PdR* que *EDP* ha instalado tiene un código *QR* en su parte frontal, de esta manera, para los usuarios de la aplicación móvil es posible determinar el *Punto de Recarga* en el que se encuentran realizando la lectura de este código mediante la aplicación si no se conoce con exactitud de qué cargador se trata. Este método se ha diseñado para instalaciones en las que hay más de un *PdR* en la misma localización, como por ejemplo el centro comercial Herón City en Valencia, donde hay dos cargadores instalados, cada uno de ellos con dos plazas reservadas:

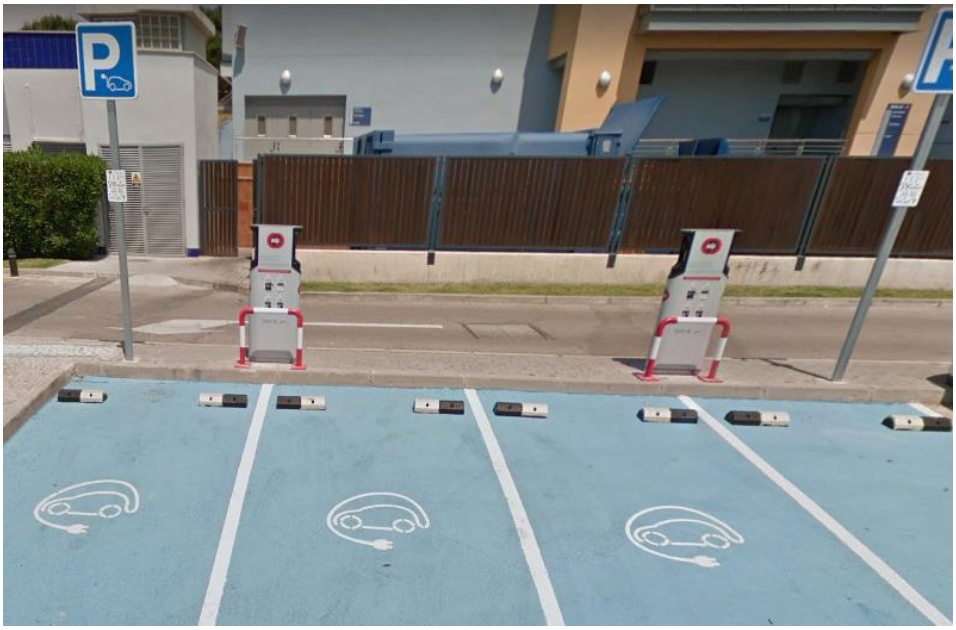

Figura 2.18 – Puntos de Recarga instalados en el centro comercial Herón City en Valencia, ejemplo de situación de uso del código QR.

Una vez el usuario haya seleccionado en la aplicación el *Punto de Recarga* y conector correctos y con su vehículo ya enchufado, podrá solicitar el inicio de la recarga. La primera vez que realice una recarga, será redirigido a una nueva ventana en la que debe introducir el número de la tarjeta de crédito con la que desee realizar los pagos. Se realiza una preautorización de cobro en la tarjeta del cliente antes de iniciar la recarga, el cual se anulará al finalizar la misma, momento en el que se hará efectivo el cobro real de los kWh consumidos. Este procedimiento sirve a *EDP* para asegurar que las recargas realizadas con sus cargadores serán abonadas.

A continuación, se incluye una tabla en la que se resume la variedad de precios asociados a las recargas en función de la potencia consumida:

Tipo de recarga	Potencia PdR (kW)	Precio recarga (€/kWh)	Tipo de conector
Convencional (hasta 7,5 kW)	3,7	0,242	Schuko
	7,5	0,242	Tipo 2
Semi-rápida (7,5 kW – 25 kW)	7,5	0,242	Tipo 2
	22	0,2783	Tipo 2
Rápida (> 25 kW)	43	0,363	Tipo 2
	50	0,363	CHAdeMO
			CCS

Tabla 2.1 – Precio de la recarga del vehículo eléctrico con EDP MoveOn en el año 2019 (21% IVA incluido).

Capítulo 3: Plataforma EDP MoveOn

Se presenta ahora la estructura física y la disposición de los elementos que participan en las comunicaciones que es necesario establecer para finalizar la recarga de los *EV* correctamente. Se desarrollará una explicación introductoria del proceso de solicitud, efectuación y finalización de recarga.

3.1 Introducción

La organización general del sistema incluye principalmente al *PdR* en el que se haya solicitado la recarga y al servidor *EDP* central. Dependiendo del tipo de recarga que se haya solicitado intervendrán diferentes elementos. Para exponerlos, a partir de ahora se referirá como "Recargas tipo 1" a las realizadas por clientes *EDP* en España, y "Recargas tipo 2" a las solicitadas fuera de España por usuarios *EDP* o dentro de España por usuarios extranjeros.

Como ha sido mencionado anteriormente, la plataforma *EDP MoveOn* tiene dos modos diferentes de acceso: mediante su página web o a través de la aplicación móvil. Tanto para solicitudes de "Recarga tipo 1" como de "Recarga tipo 2", es necesario que el interesado disponga de un *Login* de acceso como cliente y su respectiva contraseña, de forma que todas las transacciones intermedias se realicen de forma fiable.

Durante las comunicaciones entre cliente y servidor, la mayor parte del trabajo recae sobre este último, ya que, en el funcionamiento de los componentes del sistema, el cliente solo necesita disponer de un navegador web o aplicación para solicitar las transacciones.

El presente proyecto tratará más adelante los protocolos de comunicación *OCPP* y *OICP* entre el servidor y el *Punto de Recarga*; ahora se expone la operatividad general establecida, indicando las normas que intervienen y que en definitiva forman la arquitectura conjunta de la red.

3.2 Proceso de recarga del vehículo eléctrico

Se exponen a continuación las fases llevadas a cabo por los principales elementos que intervienen en la comunicación una vez se ha realizado una solicitud de "Recarga tipo 1".

El usuario, a través de la plataforma *EDP MoveOn* o aproximando su tarjeta *RFID* al lector instalado en el *PdR*, inicia el procedimiento de recarga de su *EV*. En el canal de comunicación existente, esta solicitud es transmitida desde el cargador hacia el servidor central como un requerimiento de permiso para que el cliente pueda utilizar el servicio. Si se ha solicitado la recarga a través de la aplicación móvil, el servidor, al recibir las credenciales correspondientes y tras chequear la *Base de Datos* y verificar que son correctas, responderá con una autorización para el inicio de esta. De la misma manera se procede en caso de que el cliente solicite la recarga utilizando su tarjeta de fidelización. Al solicitar la recarga mediante este modo, la tarjeta funciona como un *token de seguridad* único asociado al cliente, lo que equivale al traspaso del *Login* y contraseña del usuario, por lo que el servidor realiza la misma comprobación antes de permitir la recarga del vehículo.

Una vez se ha confirmado la identidad del usuario, el *Punto de Recarga* solicita el inicio de la transacción, en este caso la recarga, a lo que el servidor responde con una confirmación. En este momento el *EV* comienza a recargarse, recibiendo la energía que el cargador involucrado en la operación sea capaz de proporcionar.

Para terminar la recarga, el usuario debe o bien detenerla mediante la aplicación móvil o pasando su tarjeta por el lector. Si al inicio se realizó la solicitud de inicio mediante la

aplicación móvil, para terminar, es indiferente qué método utilizar, pues la tarjeta está asociada únicamente a un usuario, por lo que, si en ese momento este se encuentra ejecutando un proceso de recarga, al realizar la lectura de la tarjeta el proceso se detendrá. Por esto, de la misma manera, si la solicitud de inicio fue realizada a través de acceso *RFID*, se puede terminar el proceso de las dos formas.
Si durante la recarga, en alguna de las fases de comunicación entre entidades se pierde la conectividad, el cliente no tendrá posibilidad de realizar ninguna solicitud más: será incapaz de detener la recarga y también de desconectar su vehículo del *PdR*, pues el mismo está dotado de métodos de seguridad y procedimientos de conservación que lo impiden; será tarea del *CPO (Charging Point Operator – Operador del Punto de Recarga),* esto es, *EDP*, manipular remotamente el cargador para detener la recarga y posibilitar la desconexión del *EV*.

3.3 Características de la red

En todo momento es necesario considerar que los componentes se encuentran formando tanto una red *LAN* como una red de datos 3G; a excepción del servidor central de *EDP*. Aunque en cada instalación exista una red de área local, los cargadores siempre se encontrarán funcionando dentro de la red 3G, pues en ocasiones las instalaciones no se encuentran en zonas geográficas de fácil acceso inalámbrico y de esta manera se evitan posibles caídas del sistema.

Los *Puntos de Recarga* se encuentran en una comunicación constante con el servidor central, pues es necesario actualizar en todo momento su disponibilidad o estado en el mapa de cargadores accesible desde las distintas plataformas de *EDP MoveOn*. Existe una conexión *HTTPS (Hypertext Transfer Protocol Secure)* entre el dispositivo móvil o *PC (Personal Computer)* involucrado y el servidor de *EDP*. Protocolo adaptado para realizar las comunicaciones en Internet, recoge la sintaxis y las normas que los elementos software involucrados adoptarán durante el intercambio de información. El uso de este protocolo está actualmente más que impuesto a la hora de realizar cualquier acceso o conexión a Internet de modo seguro. Basándose en los protocolos criptográficos *SSL (Secure Sockets Layer) / TLS (Transport Layer Security)*, que trabajan con certificados X.509 y criptografía asimétrica, aporta la certeza de autenticación de usuarios. Esto es un concepto de gran interés dentro del procedimiento de recarga, pues los datos de acceso e información de los usuarios no se verán comprometidos, y las referencias en cuanto a la transacción del coste de la recarga al finalizar esta, tampoco serían accesibles por terceros. Por tanto, la solicitud de recarga enviada desde el dispositivo al servidor desde el punto de vista de la transferencia de información relevante simplemente sería un flujo de datos cifrados entre los dispositivos, imposible de descifrar y que aporta protección a la arquitectura tanto por la parte del cliente como del servidor.
Como protocolo de transferencia sobre *HTTPS*, los *PdR* pueden trabajar tanto con *SOAP (Simple Object Access Protocol)*, como con arquitectura *REST (Representational State Transfer – Transferencia de Estado Representacional)*. En las instalaciones de cargadores, se utiliza el protocolo *SOAP* cuando se realizan “Recargas tipo 1” e interviene el protocolo *OCPP;* por otro lado, la transferencia se realiza utilizando *REST* cuando se utiliza el protocolo *OICP* en las “Recargas tipo 2”. Este último modelo de comunicación entre la aplicación y el servidor permite la transmisión de datos en cualquier formato, sin restricciones; se basa en un principio cliente-servidor, con una interfaz uniforme que se centra en alcanzar escalabilidad y rendimiento a gran escala para conseguir trabajar con grandes cantidades de información. Es por esto que este es el formato más utilizado en la arquitectura que se describe cuando la comunicación se

realiza mediante el protocolo *OICP*, ya que es el modelo generalizado adaptado mundialmente. Para las comunicaciones que tienen lugar durante las "Recargas tipo 1" los *Puntos de Recarga* trabajan en base a *SOAP:* con los protocolos *OCPP SOAP 1.5* y *OCPP SOAP 1.2*; es por esto por lo que en estos se utiliza el protocolo *OCPP* versión *1.6*, ya que su última actualización, *OCPP 2.0*, no permite el uso del modelo *SOAP* durante el intercambio de mensajes. Actualmente, todos los cargadores instalados por *EDP* trabajan en la versión *1.6*. El protocolo *SOAP* se basa en *XML (Extensible Markup Language)*, que resulta en que los mensajes sean más largos y por tanto más lentos de transmitir, lo que finalmente hace que esta tecnología resulte menos flexible que utilizando *REST* para el contexto en el que se necesita integrar.

Los *Puntos de Recarga* desplegados correspondientes al grupo de infraestructura pública están configurados dentro de *APN (Access Point Name – Nombres de Puntos de Acceso)*, de esta forma se les asigna una dirección *IP (Internet Protocol)* que servirá como su forma de identificación para las posteriores comunicaciones con el servidor central. Cada uno de los *PdR* que *EDP* haya puesto a disposición de los usuarios debe estar configurado dentro de esta tecnología para que el acceso a la red móvil 3G sea posible. Existen dos tipos de configuraciones: *APN privada* y *APN pública*. En la disposición del proyecto se dan ambas opciones. La comunicación entre el servidor *EDP* y el *PdR* puede realizarse a través de una *APN pública*, en la que los usuarios comparten los recursos existentes, pero siempre manteniendo una privacidad entre ellos. En general, es el formato predeterminado existente cuando se dispone de tarjetas *SIM (Subscriber Identify Module)* en las redes. Cuando la comunicación es mediante una *APN privada*, los usuarios no tienen que compartir los recursos, lo que aporta más capacidad de prevención ante ataques. Por defecto, en esta estructura se intenta establecer la conexión a través de una *APN privada*, dado que aporta mayor seguridad; será cuando se den incompatibilidades con el sistema de autenticación y algunos equipos cuando se realicen las conexiones mediante *APN públicas*. Estas incompatibilidades suelen darse debido a que el usuario tenga su dispositivo de acceso configurado en modo *PAP (Password Authentication Protocol)* y no *CHAP (Challenge-Handshake Authentication Protocol)*. En el protocolo *PAP* se establece una vía con el sistema de autenticación mediante la cual se envían el usuario y contraseña sin cifrar. Por el contrario, en el protocolo *CHAP* se utiliza el algoritmo de encriptación *MD5 (Message-Digest Algorithm 5)*, que calcula un valor que sólo los dispositivos involucrados conocen, lo que hace al protocolo resistente a ataques de restablecimiento de la conexión o ensayo y error. Además, tanto el *ID (Identificador)* de usuario como la contraseña viajan cifrados.

Existen routers como puntos de acceso de las tarjetas *SIM* para ambos tipos de *APN*.

Al mismo tiempo, los *Puntos de Recarga* disponen de conectividad *LAN* y capacidad de conexiones 3G. Las tarjetas *SIM* necesarias para las mismas, son configuradas dentro de una *VPN (Virtual Private Network – Red Privada Virtual)*. Esta tecnología permite extender la red *LAN* sobre Internet de forma segura. La verificación de identidad de usuario es obligatoria para el acceso, de forma que se procedería a la denegación de la recarga en el supuesto de que el solicitante no tuviese credenciales. De esta manera, la comunicación entre el *PdR* y el router se establece a través de una red propiedad de la compañía telefónica.

Continuando el proceso de comunicación, la transferencia de información entre el router y el servidor central se realiza a través de una puerta de enlace en *Azure*. Este servicio en la nube funciona como plataforma para establecer comunicaciones seguras y lo que se denomina "Federación entre aplicaciones".

También, la comunicación entre el servidor central de *EDP* y el servidor *Base de Datos* se realiza a través de una *VPN*. El intercambio de cualquier información entre

estos, siempre utiliza el protocolo *TCP* (*Transmission Control Protocol*) en la capa de transporte, que permite una transferencia segura, y el protocolo *IP* en la capa de red. La siguiente imagen puede ayudar a comprender el esquema general del sistema:

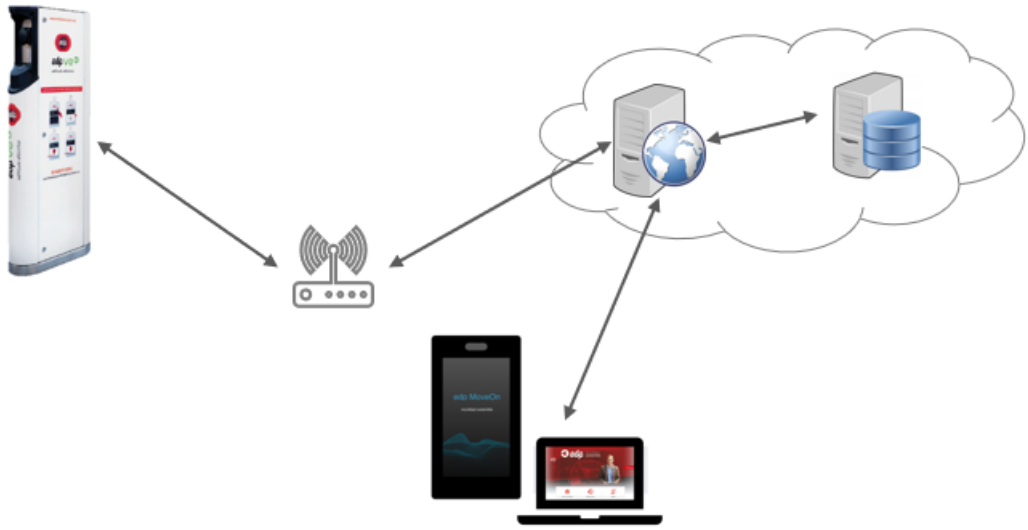

Figura 3.1 – Esquema de comunicación del sistema.

A continuación, se muestra un pequeño diagrama que sirve como introducción visual a la comunicación en una solicitud de recarga:

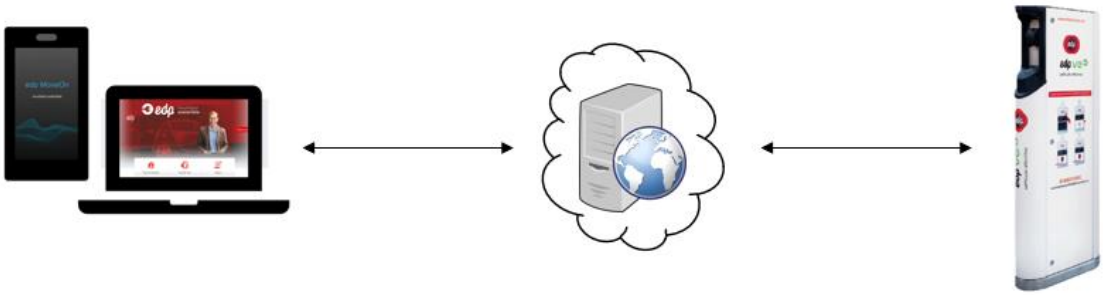

Figura 3.2 – Esquema de comunicación del sistema en 'Recarga tipo 1'. Acceso usuario (izquierda), servidor central EDP (centro), Punto de Recarga (derecha).

En la misma, el cliente enviaría una solicitud de recarga al servidor central, el cual, tras recibirla enviaría la información al *PdR* involucrado. En este momento ya sería posible proceder a la recarga del vehículo, por lo que, desde el momento de solicitud, el *EV* debería estar conectado al punto de recarga a través del conector pertinente. Se trataría en este caso de una "Recarga tipo 1", ya que el servidor se comunica directamente con el *PdR*, haciéndolo mediante el protocolo *OCPP*.

Capítulo 4: Protocolo OCPP

Se introduce ahora el protocolo *OCPP* comentado con anterioridad. En el presente capítulo, se tratará el mismo de forma global, desde sus orígenes hasta su estado actual, centrándose en el funcionamiento de la versión *OCPP 1.5,* y la utilizada por los cargadores de *EDP MoveOn*: *OCPP 1.6*.

4.1 Historia

El *Open Charge Point Protocol* [3] nace como un protocolo abierto dentro de la capa de aplicación para la comunicación entre las estaciones de carga y los servidores pertenecientes al *CPO* para la recarga de *EV*. Tiene su origen en la fundación neerlandesa *E-Laad*, actualmente *ElaadNL*, en el año 2009. La primera versión fue el protocolo *OCPP 1.0*, de pruebas; cuatro años más tarde la versión *OCPP 1.5* ya había impuesto su uso de forma generalizada en la industria de los vehículos sostenibles. Hoy en día es el protocolo mundialmente adoptado para las transacciones e intercambio de datos entre *EV* y la red eléctrica.
La tarea de promover este protocolo para establecerlo como estándar en la comunicación para la recarga de vehículos eléctricos estuvo en manos de la *OCA (Open Charge Alliance)*, un consorcio internacional formado por empresas líderes en la fabricación e implantación de *Puntos de Recarga*. Surgida de forma simultánea al protocolo, ahora sigue con la misma labor, además de impulsar también el uso de los protocolos de interoperabilidad entre plataformas.
A continuación, se describen las diferentes versiones del protocolo *OCPP*.

4.2 Versiones

Desde su inicio hasta hoy, las versiones del protocolo han sido las siguientes: *OCPP 1.*0, *OCPP 1.1, OCPP 1.2, OCPP 1.3* y *OCPP 1.4* como versiones de estudio y ensayo; OCPP *1.5*, la primera lanzada al público, hoy en día en desuso; y *OCPP 1.6* y *OCPP 2.0*, ambas utilizadas en la actualidad. Cada versión se basa en su antecesora, presentando en cada caso ciertas mejoras que responden a las necesidades que van surgiendo mediante su uso.

Las versiones desde el protocolo *OCPP 1.*0 hasta el *OCPP 1.*4 se dieron en tan sólo dos años, con meses de diferencia entre unas y otras. La primera versión, *OCPP 1.0*, surgió en octubre del año 2010, y fue la primera versión aprobada por el grupo *E-Laad*. Un mes más tarde ese mismo año aparece la versión *OCPP 1.1*, y en febrero del siguiente año la versión *OCPP 1.2*, presentando mejoras tanto de funcionalidad como de sintaxis en la programación y formato en la trata de datos. En enero de 2012 la versión *OCPP 1.3* se convierte en el primer borrador del protocolo enviado a revisión, disponiendo ya de acciones que recuerdan a las versiones utilizadas actualmente. Tras el análisis y estudio de errores de esta versión, los cambios sugeridos y anotaciones se indican en una nueva versión, *OCPP 1.4*, borrador final del protocolo.

Figura 4.1 – Logo del protocolo OCPP.

4.2.1 Versión OCPP 1.5

La versión *1.5* del protocolo *OCPP* [4] fue la primera especificación preparada para su lanzamiento, ocurrido en marzo de 2012 y tras las últimas revisiones de la versión *OCPP 1.4*. Gracias a la correcta adaptación de esta versión y de las tareas e impulso llevados a cabo por la *OCA,* los proveedores de software para los diferentes tipos de cargadores, aunque se trate de distintas compañías, sólo deben preocuparse de que sus instalaciones sean capaces de entender y utilizar este protocolo.
El protocolo *OCPP 1.5* es un sistema de comunicación bidireccional muy sencillo, basado fundamentalmente en "Requests" por parte de los *Puntos de Recarga* y "Responses" de los servidores centrales de las *CPO*. Se basa en una arquitectura *SOAP*, la cual se expone a continuación.

4.2.1.1 Protocolo SOAP

El protocolo *SOAP* [5] sirve para definir el intercambio de datos en formato *XML* entre dos entidades. Está estandarizado por la *W3C (World Wide Web Consortium – Consorcio WWW)*. Su formato en *XML*, aunque hace las transferencias más "pesadas" dado que se trata puramente de texto plano, también le otorga una gran interoperabilidad y fácil adopción, puesto que debido a este formato puede ser implementado sobre un gran número de lenguajes y plataformas distintos. Además, su formato de datos estructurados también favorece una comprensión más clara de la información que contiene. El uso de *SOAP* dentro del protocolo *OCPP* es ideal, ya que, como se ha comentado anteriormente, *OCPP* se basa puramente en peticiones y respuestas, y *SOAP* es utilizado principalmente para comunicaciones *RPC (Remote Procedure Call)*. En las *RPC*, es el cliente quien inicia los procesos, enviando solicitudes al servidor para iniciar un cierto procedimiento; en el presente proyecto, esto se traduce como el cliente de *EDP MoveOn* enviando solicitudes de recarga desde el *PdR* al servidor central, y respondiéndole este con una confirmación o denegación de la misma. Con lo descrito hasta ahora, se interpreta que la configuración del sistema se trata de un *Web Service*, ya que las comunicaciones se realizan mediante *HTTPS* entre el navegador y el cliente, y el cliente y el servidor, y con *SOAP* entre este y el backend.

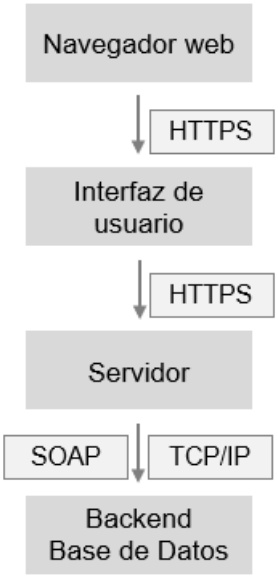

Figura 4.2 – Estructura Web Service del sistema.

El cliente *EDP* solicita una recarga mediante su navegador, entre ellos existe una comunicación *HTTPS*. La información se transmite desde el usuario al servidor central utilizando también *HTTPS*, y este último se comunica con el servidor *Base de Datos* para verificar al usuario. En cuanto a la sincronización en *SOAP*, en este contexto, si se recibe una petición *OCPP* mediante *HTTP*, la respuesta a la misma debe ser incluida en la respuesta *HTTP*, pues no se admite el intercambio asíncrono de mensajes *SOAP* en el protocolo *OCPP*.
El intercambio de información entre el cliente y los *Puntos de Recarga* se considera de la siguiente forma:

Figura 4.3 – Diagrama de comunicación.

El formato de los mensajes *SOAP* consta de las siguientes partes:

- Envelope – Sobre: Es obligatorio, el elemento raíz que envuelve todo el mensaje, contiene la cabecera y el cuerpo e identifica al conjunto como un mensaje *SOAP*.
- Header – Cabecera: Se incluye opcionalmente con información adicional, como la seguridad, encaminamiento del mensaje o su modo de procesamiento.
- Body – Cuerpo: Parte obligatoria que contiene la información relativa a la solicitud por parte de los clientes o las respuestas de los servidores. Es específico de la aplicación y de los datos que se intercambian.
- Fault – Error: Incluye, de forma opcional, información relativa a posibles errores que se hayan podido dar durante la transmisión.

4.2.1.2 Operaciones en OCPP 1.5

La versión *OCPP 1.5* es la primera anunciada para uso público y en ella se describen 25 posibles *PDUs (Protocol Data Unit – Unidad de Datos de Protocolo)* distintas, de las cuales 10 son iniciadas desde el *Punto de Recarga* y las 15 restantes desde el servidor. A continuación, se incluyen dos tablas, en las que se recoge cada una de ellas:

Punto de Recarga	
Authorize	Solicitud para iniciar una recarga
Boot Notification	Información sobre la configuración del cargador para que el servidor pueda verificarlo
Data Transfer	Si el cargador necesita enviar datos no soportados por OCPP al servidor
Diagnosis Status Notification	Notificación al servidor de que una actualización de diagnóstico ha finalizado
Firmware Status Notification	Notificación al servidor para informar del progreso en la actualización del firmware
Heartbeat	Notificación al servidor para informar de que el cargador se encuentra activo. Se envía continuamente en intervalos de tiempo configurados
Meter Values	El cargador puede muestrear el medidor de electricidad y proporcionar información sobre el mismo al servidor
Start Transaction	Notificación al servidor de que el vehículo eléctrico está preparado para iniciar la transacción
Status Notification	Notificación sobre la condición de estado actual del cargador
Stop Transaction	Notificación al servidor de que el vehículo eléctrico está preparado para finalizar la transacción

Tabla 4.1 – Conjunto de PDUs disponibles desde el cargador en la versión OCPP 1.5.

Servidor	
Cancel Reservation	Cancelación de la reserva de un cargador
Change Availability	Solicitud al cargador para que cambie su estado de disponibilidad
Change Configuration	Solicitud para cambiar los parámetros de configuración del cargador
Clear Cache	Solicitud para que el cargador limpie su memoria caché
Data Transfer	Si el servidor necesita enviar datos no soportados por OCPP al cargador
Get Configuration	Para recuperar los ajustes de configuración
Get Diagnostics	Solicitud de datos de diagnóstico
Get Local List Version	Solicitud de la lista local de autorización para soportar sincronizaciones
Remote Start Transaction	Solicitud de forma remota para iniciar una transacción
Remote Stop Transaction	Solicitud de forma remota para finalizar una transacción
Reserve Now	Emisión de un comando de reserva de un conector para ser utilizado con un determinado id-tag
Reset	Solicitud a un cargador para que este se resetee
Send Local List	Envío de una lista de autorización al cargador para que este pueda conocer los id-tags autorizados
Unlock Connector	Comando para desbloquear un determinado conector del cargador
Update Firmware	Notificación al cargador para actualización de su firmware

Tabla 4.2 – Conjunto de PDUs disponibles desde el servidor en la versión OCPP 1.5.

4.2.2 Versión OCPP 1.6

La versión final del protocolo *OCPP 1.6* [6] aparece en octubre del año 2015. Incorpora nuevas características para adaptarse a la evolución del mercado que rodea al transporte sostenible: “Smart Charging”, mejoras en cuanto a las posibilidades de diagnóstico, más variedad de estados de los cargadores o nuevos comandos de estado; aunque la actualización más destacable es la modificación del modo de transporte de los datos, pasando a utilizar *JSON (JavaScript Object Notation)* sobre *WebSockets*, pero soportando todavía el formato *SOAP*.

El concepto de “Smart Charging” hace referencia a la habilidad para influir ahora tanto en la potencia que se transfiere entre el cargador y el *EV* como en el consumo total de energía de un *Punto de Recarga* o de un conjunto de estos cuando se encuentran instalados en la misma localización, como en el ejemplo citado en el segundo capítulo: el centro comercial Herón City en Valencia [Figura 2.18]. Se dan tres posibles casos en los que utilizar las características de “Smart Charging”: balanceo de carga, carga inteligente central y carga inteligente local. En el primer escenario se pretende controlar el reparto de carga interno del *PdR* entre sus distintos conectores, así como entre otros cargadores si se trata de un conjunto de *Puntos de Recarga* localizados en el mismo lugar y que están conectados a la misma red eléctrica. De esta manera, se reparte la potencia en función del consumo necesario de cada cargador en ese momento, así como del tipo de conector al que se encuentre conectado el vehículo. Refiriéndose a los otros dos casos, en ambos se realiza un control remoto implementando ciertos límites de potencia en las recargas. Estos límites son calculados en función a las condiciones de la red eléctrica de cada localización en la que se trabaja, el número de cargas medio de cada *Punto de Recarga* u otros parámetros de disponibilidad o funcionalidades de los mismos. En la carga inteligente central este control queda en manos del servidor central del *CPO*, y en la carga inteligente local la monitorización es gestionada por un cierto controlador local.

En base al resto de eventos que incorpora la versión *OCPP 1.6*, aparecen nuevos comandos en las transacciones.

Seguidamente se comenta el formato *JSON*.

4.2.2.1 Formato JSON

En esta nueva versión del protocolo se incluye la posibilidad de enviar los datos en formato *JSON* [7], lo que hace que la transacción de información viaje de forma mucho más compacta en comparación con *SOAP*. *JSON* determina un conjunto de reglas de formato para la representación de datos estructurados. Su estándar está definido por la *IETF (Internet Engineering Task Force – Grupo de Trabajo de Ingeniería de Internet)* desde marzo de 2014 en el *RFC (Request For Comments) 7159*.

Aunque es simplemente un formato de texto para intercambiar datos, debido a su amplia utilización como elección distinta a *XML*, desde el año 2019 se considera un formato de lenguaje independiente, surgido como un subconjunto de la notación de objetos en *JavaScript.* La principal ventaja de esta nueva característica es que *JSON* se implementa sobre *WebSockets*, de manera que la comunicación se realiza de forma bidireccional sobre un único *socket TCP*. El protocolo *WebSocket* está estandarizado en el *RFC 6455*, y la *API (Application Programming Interface – Interfaz de Programación de Aplicaciones)* del mismo está actualmente siendo normalizada por la *W3C*.

En las comunicaciones entre los *Puntos de Recarga* y el servidor central utilizando *JSON*, ahora los primeros actuarán como un cliente *WebSocket,* y el servidor lo hará como servidor *WebSocket*. El primer paso en este tipo de intercambio de datos es parecido al establecimiento de una sesión *HTTP;* se realiza una negociación o

handshake por parte del cliente para crear una conexión *WebSocket,* a la que el servidor responderá para iniciar el intercambio de datos.
Para el establecimiento de la sesión, es necesario que el cliente proporcione al servidor una *URL (Uniform Resource Locator – Localizador de Recursos Uniforme)* al enviar el *handshake*; en la misma incluye su propio *ID*, de manera que, al recibirlo, el servidor sabrá de qué cargador proviene el *socket.* Si el servidor no reconoce al *PdR*, debería enviar una respuesta *HTTP* con línea de estado *404 – Not Found*, abortando así la conexión *WebSocket.* En el caso contrario, si el servidor ha reconocido al *Punto de Recarga,* pero no está de acuerdo con ciertos aspectos o protocolos sugeridos por el cliente en la negociación o se le sugiere la elección entre varios, responderá completando el *WebSocket* con las características que desea establecer y seguidamente cerrará la negociación, pasando a comenzar el intercambio de datos necesarios respondiendo a la solicitud del usuario.

A continuación, se incluye un ejemplo de negociación para el establecimiento de una comunicación *WebSocket* por parte del cliente, así como su respectiva respuesta desde el servidor:

GET /webServices/ocpp/CP3211 **HTTP/1.1**

Host: some.server.com:33033

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==

Sec-WebSocket-Protocol: ocpp1.6, ocpp1.5

Sec-WebSocket-Version: 13

Figura 4.4 – Handshake del cliente (JSON) [7].

HTTP/1.1 101 Switching Protocols

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=

Sec-WebSocket-Protocol: ocpp1.6

Figura 4.5 – Respuesta del servidor (JSON) [7].

En ambas imágenes, los datos que aparecen en negrita son comunes en todas las peticiones *handshake*, el resto se refieren a características concretas del ejemplo.
Centrándose en el envío del *handshake* [Figura 4.4], el método enviado por el cliente hacia el servidor es un *GET* utilizando la versión *HTTP/1.1*. En la misma petición, como se ha comentado con anterioridad, se indica la *URL* del cargador, que en este caso sería *</webServices/ocpp/CP3211/>*; el servidor al que se envía la petición sería *some.server.com:33033*. En la misma figura, el valor que se especifica en el campo *Sec-WebSocket-Key* es un valor calculado aleatoriamente que el servidor utilizará junto al incluido en el campo *Sec-WebSocket-Accept* en la respuesta [Figura 4.5] para construir un *token* y así confirmar que ha recibido y leído la solicitud del cliente. Continuando con el campo *Sec-WebSocket-Protocol* [Figura 4.4], el cliente informa de las dos versiones del protocolo *OCPP* que es capaz de soportar en la comunicación. En la respuesta del servidor [Figura 4.5], el primer método es la respuesta *HTTP* con código de estado *101*, utilizado para indicar cierta información al cliente, en este caso la elección de la versión del protocolo *OCPP* que se utilizará, señalado mediante el comando *Switching Protocols.* La versión final que se utilizará viene indicada en el campo *Sec-WebSocket-Protocol*, y será *OCPP 1.6.*

4.2.2.2 Operaciones en OCPP 1.6

Al igual que en la versión que le precede, las *PDUs* disponibles que el *Punto de Recarga* es capaz de emitir hacia el servidor son 10, las mismas que en *OCPP 1.5* [Tabla 4.1]; las nuevas operaciones de esta versión son todas enviadas desde el servidor al cargador, y se refieren a las nuevas características introducidas en esta versión.

Servidor	
Clear Charging Profile	Si el servidor desea eliminar algunos o todos los perfiles de carga que hayan sido enviados con anterioridad al cargador
Get Composite Schedule	Si el servidor solicita información acerca de los límites de tiempo que se pueden alcanzar entre las notificaciones de disponibilidad
Set Charging Profile	Solicitud para crear un perfil de recarga al comienzo de una transacción, al comienzo de una transacción remota, durante una transacción para modificar el perfil de la misma o fuera del concepto de transacción, como un mensaje aparte
Trigger Message	Solicitud de envío del último estado conocido del cargador cuando el servidor necesita una actualización del mismo debido a que se ha sobrepasado el límite de tiempo u otras causas

Tabla 4.3 – Conjunto de PDUs disponibles desde el servidor en la versión OCPP 1.6.

En total, es servidor es capaz de llevar a cabo 19 acciones, siendo estas las enumeradas en la tabla anterior [Tabla 4.3] y las ya disponibles en la versión *OCPP 1.5* [Tabla 4.2].

Además, en la versión *OCPP 1.6*, todas las *PDUs* disponibles se organizan en *perfiles de funciones*; dependiendo de la funcionalidad de cada una, se distinguen 6 tipos:

- Core – Núcleo: Funcionalidades básicas ya presentes en *OCPP 1.5*, sin soporte para actualizaciones de firmware, administración de listas de autoridad locales o reservas.
- Firmware Management – Administración del Firmware: Capacidad de actualización de firmware y descarga de archivos de registros de diagnóstico.
- Local Auth List Management – Administración de la Lista de Autoridad Local: Funcionalidades para gestionar las listas de autoridad locales de los cargadores.
- Reservation – Reserva: Soporte para la reserva de cargadores.
- Smart Charging – Carga Inteligente: Soporte básico para el uso de funcionalidades de carga inteligente.
- Remote Trigger – Disparador Remoto: Función para lanzar solicitudes remotas de mensajes ya iniciados.

En la siguiente tabla se recoge el conjunto de *PDUs* disponibles en la versión *OCPP 1.6*, organizadas según su perfil:

Perfil	Mensaje
Core	Authorize, BootNotification, ChangeAvailability, ChangeConfiguration, ClearCache, DataTransfer, GetConfiguration, Heartbeat, MeterValues, RemoteStartTransaction, RemoteStopTransaction, Reset, StartTransaction, StatusNotification, StopTransaction, UnlockConnector
Firmware Management	GetDiagnostics, DiagnosticsStatusNotification, FirmwareStatusNotification, UpdateFirmware
Local Auth List Management	GetLocalListVersion, SendLocalList
Reservation	CancelReservation, ReserveNow
Smart Charging	ClearChargingProfile, GetCompositeSchedule, SetChargingProfile
Remote Trigger	TriggerMessage

Tabla 4.4 – Organización de las PDUs disponibles según su perfil en la versión OCPP 1.6.

4.2.3 Versión OCPP 2.0

La última versión del protocolo, *OCPP 2.0* data de abril de 2018 e incorpora gran cantidad de funcionalidades, de las cuales, la más notable es la eliminación del protocolo *SOAP* como protocolo de transporte [8]. La decisión fue tomada por los miembros de la *OCA*, quienes consideraron que dicho protocolo ya no es el más adecuado para operar con los recursos informáticos que ciertas funcionalidades requieren y bajo las cuales los *Puntos de Recarga* comienzan a operar. El protocolo *SOAP* requiere mayor ancho de banda debido a su formato y esto puede ocasionar transacciones más lentas; lo que a su vez deriva en un aumento del coste en la conexión a través del dispositivo móvil a la red en el cliente.
Otra de las nuevas funcionalidades que incluye es la administración de los cargadores, con capacidad de solicitar informes de inventario, mejoras en las solicitudes de error y estado, avances en la configuración o monitorización personalizable para cada *CPO*. Este conjunto de competencias abarata el mantenimiento físico de los *PdR*.
Debido a la rápida adaptación del protocolo *OCPP* dentro de la movilidad sostenible y al aumento de usuarios de *EV* en pocos años, aparecen funcionalidades para facilitar la gestión de grandes cantidades de transacciones y datos. Con el crecimiento del mercado de la *e-mobility*, los servidores centrales de los operadores de cargadores se ven obligados a manejar una mayor cantidad de información, por lo que se modifica la manera de realizar los intercambios. Las *PDUs* consideradas básicas por tratarse simplemente de transferencia de datos; esto es: *StartTransaction, StopTransaction, MeterValue* y *StatusNotification*, son ahora reemplazadas por una sola: *TransactionEvent*. Además, debido a que a partir de esta versión sólo será posible utilizar el método *JSON*, se incluye soporte de compresión para *WebSocket*, que reduce incluso más la cantidad de datos en la red.
En esta versión se incorporan también mejoras en cuanto a la *ciberseguridad*: perfiles de seguridad, gestión de claves para certificados del lado del cliente, actualizaciones seguras de firmware y registros de eventos de seguridad.
Las funcionalidades en cuanto a la carga inteligente también son extendidas, añadiendo entradas de carga inteligente directas desde el *EMS (Energy Management System – Sistema de Administración de Energía)* hasta los *Puntos de Recarga*, mejora de la misma mediante controladores locales o soporte para integrar las cualidades de los vehículos con los cargadores.
Se introduce además soporte para la norma *ISO 15118*, el estándar que define la comunicación *V2G (Vehicle-to-Grid)* en la carga de *EV*, donde se recogen nuevas características para la conexión y carga de los vehículos, así como métodos de carga inteligente incluidos en la entrada de carga de los coches.
Se tiene en cuenta la experiencia de usuario, incluyendo cambios menores como renombre de algunas *PDUs* y añadiendo nuevas opciones de autorización, nuevos mensajes en las pantallas de los cargadores, mensajes de elección de preferencia de lenguaje o información de tarifas y costes.

4.2.3.1 Bloques funcionales en OCPP 2.0

Aunque en el plan *EDP MoveOn*, la versión *OCPP 2.0* no se utiliza, a continuación, se indican los diferentes bloques funcionales en los que se organiza y en los que se clasifican las *PDUs* disponibles en la versión:

Bloque Funcional	Descripción
Seguridad	Especificación de seguridad para el protocolo OCPP
Aprovisionamiento	Funcionalidades que ayudan a los CPO a administrar sus cargadores
Autorización	Funcionalidades relacionadas con la autorización; mensajes y memoria caché
Administración de Lista Local de Autorización	Funcionalidades para gestionar la Lista Local de Autorización
Transacciones	Funcionalidades para transacciones básicas sobre OCPP
Control Remoto	Control remoto de los cargadores: control de transacción remota, desbloqueo de conectores y trigger remoto
Disponibilidad	Estado del cargador y envío de mensajes de notificación
Reserva	Funcionalidad para reservar un cargador
Tarifas y Costes	Funcionalidades de información acerca de tarifas y costes de recarga al usuario del vehículo eléctrico
Medidas	Funcionalidades de valores de medidas, pruebas periódicas y límites de tiempo
Carga Inteligente	Funcionalidades para influir en las características de la recarga, imponer límites de potencia…
Administración del Firmware	Actualizaciones de firmware
Administración del Certificado ISO 15118	Instalación y actualizaciones de los certificados ISO 15118
Diagnósticos	Funcionalidades que permiten a los CPO requerir diagnósticos de uso, cargar carpetas de evaluación o monitorizar el cargador
Mensajes de Display	Funcionalidades para mostrar mensajes en pantalla que no forman parte del firmware
Transferencia de Datos	Funcionalidades para añadir características de transferencia de datos

Tabla 4.5 – Bloques funcionales que se diferencian en OCPP 2.0.

Capítulo 5: Protocolo OCPP – Aplicación Práctica

Tras haber comentado las diferentes versiones del protocolo *OCPP* en el capítulo anterior, ahora se presenta un ejemplo práctico del mismo, utilizando la infraestructura y accesos de usuario de la plataforma *EDP MoveOn*.
En la implementación del sistema llevada a cabo por *EDP*, todos los *Puntos de Recarga* trabajan utilizando el protocolo de transporte *SOAP*, utilizando la versión *1.6* del protocolo *OCPP*. Los ejemplos que se muestran a continuación pertenecen a una misma captura, y se han realizado utilizando el programa *Wireshark*; en todos los casos capturando desde el servidor, y siempre utilizando el cargador de pruebas disponible para tareas de administración y gestión. Al tratarse de este cargador en concreto, no es posible realizar las solicitudes a través de la aplicación móvil, pues este *PdR* no se encuentra en el mapa ni en el listado de cargadores visible para los usuarios, por lo que se ha utilizado la tarjeta de fidelización para enviar las solicitudes.
Las acciones llevadas a cabo son las siguientes: reserva del cargador, recarga del *EV* y finalización de la recarga.

Por motivos de confidencialidad, EDP ha solicitado que en los ejemplos incluidos en el documento no se muestren los identificadores del Punto de Recarga o del servidor; se mostrarán parcialmente o se hará referencia a los mismos con ejemplos.

IP del PdR: 93.156.191.36
IP del servidor: 192.168.0.110

5.1 Reserva

Dado que no es posible realizar la reserva de un cargador mediante la tarjeta de usuario, se ha procedido a reservar el cargador de pruebas a través de la página de administración del equipo de gestión de *O2E*, accediendo mediante credenciales de mantenimiento. Es por esto por lo que en la captura que se muestra a continuación, la solicitud de reserva tiene su origen en el servidor y va hacia el cargador.

```
192.168.0.110  7.330256  93.156.191.36  HTTP/XML  737 POST / HTTP/1.1
```

Figura 5.1 – Trama de solicitud de reserva del cargador.

En el primer campo de la trama anterior, *192.168.0.110*, corresponde a la *IP* de origen, es decir, el servidor; el segundo es el tiempo transcurrido desde el envío de la primera trama de la captura hasta el envío de la que se muestra, seguido de la dirección *IP* de destino, *93.156.191.36*, que corresponde al cargador. En el envío de la información, se utiliza el protocolo *HTTP 1.1*, abriéndose una conexión nueva para cada petición, y utilizando *XML* como formato de datos. Se indica también la longitud de la trama capturada, en este caso 737 bytes. Por último, la petición realizada, un *POST* para enviar datos desde el servidor al *PdR*, en este caso la solicitud de reserva del mismo.

```
Hypertext Transfer Protocol
▾ POST / HTTP/1.1\r\n
  ▾ [Expert Info (Chat/Sequence): POST / HTTP/1.1\r\n]
      [POST / HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: POST
    Request URI: /
    Request Version: HTTP/1.1
  Host: 93.156.191.36:8081\r\n
  Connection: Keep-Alive\r\n
  User-Agent: PHP-SOAP/7.2.11\r\n
  Content-Type: application/soap+xml; charset=utf-8; action="/ReserveNow"\r\n
▾ Content-Length: 491\r\n
    [Content length: 491]
  \r\n
  [Full request URI: http://93.156.191.36:8081/]
  [HTTP request 1/1]
  File Data: 491 bytes
```

Figura 5.2 – Trama de solicitud de reserva del cargador – detalle HTTP.

Examinando los detalles de la solicitud, se observa que se utiliza el protocolo *SOAP* para la transferencia de datos en formato *XML*; además de la solicitud de reserva, que se indica en el campo *Content-Type: action="/ReserveNow"*.

```
eXtensible Markup Language
▾ <?xml
      version="1.0"
      encoding="UTF-8"
      ?>
▾ <env:Envelope
      xmlns:env="http://www.w3.org/2003/05/soap-envelope"
      xmlns:ns1="urn://Ocpp/Cp/2012/06/">
  ▾ <env:Header>
    ▾ <ns1:chargeBoxIdentity
          env:mustUnderstand="true">
          cargador1
          </ns1:chargeBoxIdentity>
      </env:Header>
  ▾ <env:Body>
    ▾ <ns1:reserveNowRequest>
      ▾ <ns1:connectorId>
            [redacted]
            </ns1:connectorId>
      ▾ <ns1:expiryDate>
            2019-07-05T13:43:45Z
            </ns1:expiryDate>
      ▾ <ns1:idTag>
            [redacted]
            </ns1:idTag>
      ▾ <ns1:reservationId>
            1
            </ns1:reservationId>
        </ns1:reserveNowRequest>
      </env:Body>
    </env:Envelope>
```

Figura 5.3 – Trama de solicitud de reserva del cargador – detalle SOAP-XML.

Accediendo a la información *XML* completa ya se pueden diferenciar las partes correspondientes al sobre y al cuerpo del mensaje del protocolo *SOAP*.
En la parte del sobre, elemento raíz del mensaje, se indica la dirección en la que se definen todos los elementos por defecto: *http://www.w3.org/2003/05/soap-envelope*, identificando al conjunto como mensaje *SOAP*, y donde también se distingue el protocolo de aplicación que se utiliza, *OCPP*. Asimismo, aparece en la cabecera el nombre asignado al *Punto de Recarga* para la realización de este ejemplo, *cargador1*.
En el cuerpo del mensaje aparece la solicitud de reserva, *reserveNowRequest*, *PDU* enviada desde el servidor, como se había indicado anteriormente [Tabla 4.2], para retener cierto conector del cargador para cualquier otro usuario. En el caso de un *PdR* instalado para uso del público, el conector se mantendrá bloqueado hasta que el usuario que haya realizado la reserva llegue para utilizarlo, para el resto de los usuarios será inaccesible hasta pasado un tiempo establecido.

En este momento, tras la solicitud de reserva recibida, el *Punto de Recarga* debe modificar su disponibilidad y enviar una actualización de su estado al servidor. Esto es necesario ya que, en el mapa y listado de cargadores en la aplicación móvil, el cargador que acaba de ser reservado debería aparecer como ocupado, evitando así que otros usuarios puedan utilizarlo.

```
93.156.191.36   8.505559    192.168.0.110   HTTP/XML   1329 POST /msr/Servicio/15/ HTTP/1.1
```

Figura 5.4 – Trama de solicitud de actualización de estado del cargador (1).

La trama superior corresponde a la solicitud de actualización de estado enviada desde el cargador que acaba de ser reservado hacia el servidor.

```
Hypertext Transfer Protocol
▾ POST /msr/Servicio/15/ HTTP/1.1\r\n
  ▾ [Expert Info (Chat/Sequence): POST /msr/Servicio/15/ HTTP/1.1\r\n]
      [POST /msr/Servicio/15/ HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: POST
    Request URI: /msr/Servicio/15/
    Request Version: HTTP/1.1
  Host: [redacted]\r\n
  User-Agent: gSOAP/2.8\r\n
  Content-Type: application/soap+xml; charset=utf-8; action="/StatusNotification"\r\n
▾ Content-Length: 997\r\n
    [Content length: 997]
  Connection: close\r\n
  Accept-Encoding: gzip, deflate\r\n
  SOAPAction: "/StatusNotification"\r\n
  \r\n
  [Full request URI: http://[redacted]/msr/Servicio/15/]
  [HTTP request 1/1]
  [Response in frame: 1149]
  File Data: 997 bytes
```

Figura 5.5 – Trama de solicitud de actualización de estado del cargador (1) – detalle HTTP.

En la petición *POST* anterior, el cargador envía una *PDU* de *StatusNotification* [Tabla 4.1] al servidor.

Figura 5.6 – Trama de solicitud de actualización de estado del cargador (1) – detalle SOAP-XML.

En la imagen superior [Figura 5.6] se indica el contenido del mensaje *SOAP* para la actualización del estado del cargador. Al igual que en la previa [Figura 5.3], en el sobre se indica la dirección de definición de los elementos, incluyendo también la de codificación, definición de instancias, esquema y direccionamiento. En la cabecera del mensaje aparece el *ID* de la solicitud, también las direcciones *IP* reales del cargador y del servidor, las cuales se han borrado, y la operación a realizar, un *StatusNotification* que envía el cargador denominado *cargador1.* Por último, en el cuerpo, se indica el *ID* del *PdR* que envía la solicitud y su estado actual, que ahora aparece como *Reserved.*

Para que la actualización se lleve a cabo, el servidor debe responder positivamente:

```
192.168.0.110  8.873086  93.156.191.36   HTTP/XML    452 HTTP/1.1 200 OK
```

Figura 5.7 – Trama de respuesta a la solicitud de actualización de estado del cargador (1).

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
        [HTTP/1.1 200 OK\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
  Content-Type: application/soap+xml; charset=utf-8\r\n
  SERVER: \r\n
  X-POWERED-BY: \r\n
  Date: Fri, 05 Jul 2019 09:43:46 GMT\r\n
  Connection: close\r\n
  Content-Length: 213\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.367527000 seconds]
  [Request in frame: 1056]
  [Request URI: http://[redacted]/msr/Servicio/15/]
  File Data: 213 bytes
```

Figura 5.8 – Trama de respuesta a la solicitud de actualización de estado del cargador (1) – detalle HTTP.

Como se puede observar en la imagen superior [Figura 5.8], el servidor acepta la solicitud de modificación de estado del cargador enviando el código de estado *200*, que indica que el procedimiento se ha llevado a cabo correctamente.

```
eXtensible Markup Language
  \357\273\277
  <?xml
      version="1.0"
      encoding="UTF-8"
      ?>
  <env:Envelope
      xmlns:env="http://www.w3.org/2003/05/soap-envelope"
      xmlns:ns1="urn://Ocpp/Cs/2012/06/">
    <env:Body>
        <ns1:statusNotificationResponse/>
        </env:Body>
      </env:Envelope>
```

Figura 5.9 – Trama de respuesta a la solicitud de actualización de estado del cargador (1) – detalle SOAP-XML.

El mensaje en formato *SOAP* simplemente constaría del sobre indicando el formato del mismo y el protocolo *OCPP* que se utiliza; y del cuerpo del mensaje señalando que se trata de una *statusNotificationResponse*, respuesta a la solicitud anterior del *PdR*.

Si se estuviera reservando el cargador desde la aplicación móvil de *EDP MoveOn*, al recibir el servidor la *StatusNotificationRequest* y responder a esta de manera positiva, el *PdR* implicado aparecería en color azul – activo, pero siendo utilizado por otro cliente en ese momento.

El siguiente paso corresponde a la recarga del vehículo una vez el cliente hubiese llegado a la localización de este.

5.2 Recarga

Para comenzar el proceso de recarga es necesario que el usuario conecte el vehículo al conector previamente reservado, después, mediante la aplicación móvil debe iniciarla presionando en “Recargar”.
En el entorno en que se ha realizado la aplicación práctica que se expone, el inicio de la recarga se realiza mediante la lectura de la tarjeta asociada a la persona del equipo de gestión que solicitó previamente la reserva a través del servidor.

```
93.156.191.36   33.209282   192.168.0.110  HTTP/XML   1401 POST /msr/Servicio/15/ HTTP/1.1
```

Figura 5.10 – Trama de solicitud de inicio de recarga.

Como se ve en la trama anterior, el *Punto de Recarga* envía hacia el servidor la solicitud una vez se ha leído la tarjeta del usuario; está listo para recibir confirmación y comenzar la recarga.

A continuación, se muestra la especificación de la comunicación *HTTP*:

```
Hypertext Transfer Protocol
▾ POST /msr/Servicio/15/ HTTP/1.1\r\n
  ▾ [Expert Info (Chat/Sequence): POST /msr/Servicio/15/ HTTP/1.1\r\n]
      [POST /msr/Servicio/15/ HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: POST
    Request URI: /msr/Servicio/15/
    Request Version: HTTP/1.1
  Host: [redacted]\r\n
  User-Agent: gSOAP/2.8\r\n
  Content-Type: application/soap+xml; charset=utf-8; action="/StartTransaction"\r\n
▾ Content-Length: 1072\r\n
    [Content length: 1072]
  Connection: close\r\n
  Accept-Encoding: gzip, deflate\r\n
  SOAPAction: "/StartTransaction"\r\n
  \r\n
  [Full request URI: http://[redacted]/msr/Servicio/15/]
  [HTTP request 1/1]
  [Response in frame: 1984]
  File Data: 1072 bytes
```

Figura 5.11 – Trama de solicitud de inicio de recarga – detalle HTTP.

Mediante una solicitud *POST* en *HTTP 1.1*, el cargador envía la solicitud *StartTransaction*.

Figura 5.12 – Trama de solicitud de inicio de recarga – detalle SOAP-XML.

En el sobre de la solicitud de recarga se incluyen los mismos campos que en el envío de solicitud de actualización del estado [Figura 5.6], ya que ambas peticiones se envían desde el cargador. De nuevo se ocultan las direcciones *IP* de los elementos que intervienen, así como el *ID* del conector que se utiliza, el mismo que se había reservado previamente. Para probar que esta solicitud tiene su origen en el mismo cargador cuyo conector se había reservado, se muestra el campo *reservationId* del cuerpo del mensaje, cuyo valor, *1*, coincide con el valor en el campo *reservationId* de la solicitud de reserva [Figura 5.3].
Antes de que comience el intercambio de energía entre la red eléctrica y el *PdR* para recargar el vehículo, el servidor debe recibir y aceptar la solicitud.

```
192.168.0.110   35.385638   93.156.191.36   HTTP/XML    588 HTTP/1.1 200 OK
```

Figura 5.13 – Trama de recarga aceptada.

El servidor indica que ha recibido la petición del cargador, y comunica el éxito en la operación mediante el código de estado *200* en *HTTP 1.1*:

```
Hypertext Transfer Protocol
▾ HTTP/1.1 200 OK\r\n
  ▾ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Content-Type: application/soap+xml; charset=utf-8\r\n
  SERVER: \r\n
  X-POWERED-BY: \r\n
  Date: Fri, 05 Jul 2019 09:44:13 GMT\r\n
  Connection: close\r\n
▾ Content-Length: 349\r\n
    [Content length: 349]
  \r\n
  [HTTP response 1/1]
  [Time since request: 2.176356000 seconds]
  [Request in frame: 1826]
  [Request URI: http://[redacted]/msr/Servicio/15/]
  File Data: 349 bytes
```

Figura 5.14 – Trama de recarga aceptada – detalle HTTP.

```
eXtensible Markup Language
  \357\273\277
▾ <?xml
    version="1.0"
    encoding="UTF-8"
    ?>
▾ <env:Envelope
    xmlns:env="http://www.w3.org/2003/05/soap-envelope"
    xmlns:ns1="urn://Ocpp/Cs/2012/06/">
  ▾ <env:Body>
    ▾ <ns1:startTransactionResponse>
      ▾ <ns1:transactionId>
          [redacted]
          </ns1:transactionId>
      ▾ <ns1:idTagInfo>
        ▾ <ns1:status>
            Accepted
            </ns1:status>
          </ns1:idTagInfo>
        </ns1:startTransactionResponse>
      </env:Body>
    </env:Envelope>
```

Figura 5.15 – Trama de recarga aceptada – detalle SOAP-XML.

En la parte de la trama relativa al contenido *SOAP – XML* se puede encontrar en el sobre la misma dirección *SOAP* y protocolo *OCPP* que en la trama de solicitud de reserva del cargador [Figura 5.3]. En el contenido del cuerpo *SOAP* se indica que la solicitud transferida es una *startTransactionResponse*, respuesta asociada a la previa petición del cargador para iniciar la recarga [Figura 5.9] y cuyo estado aparece como *Accepted*, por lo que la solicitud ha sido aprobada y hay permiso para comenzar la recarga del vehículo.

Antes de comenzar, de nuevo el *Punto de Recarga* debe actualizar su estado e informar del mismo al servidor central:

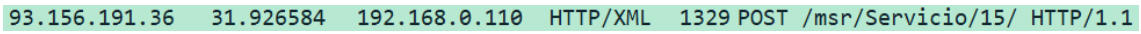

Figura 5.16 – Trama de solicitud de actualización de estado del cargador (2).

Figura 5.17 – Trama de solicitud de actualización de estado del cargador (2) – detalle HTTP.

Para informar de la modificación de su estado, el cargador envía de nuevo una solicitud *StatusNotification* con destino el servidor. Una vez el servidor reciba la notificación, el estado del cargador pasará a ser “ocupado”.
Realizando estas operaciones a través de la aplicación móvil, en la pantalla de visualización de los distintos *Puntos de Recarga* que existen, el *PdR* involucrado seguirá apareciendo en azul – activo, pero en uso por otro cliente; sólo se puede visualizar el cambio de estado “real”, de “reservado” a “ocupado” observando los detalles del sobre *SOAP*:

```
eXtensible Markup Language
<?xml
    version="1.0"
    encoding="UTF-8"
    ?>
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
    xmlns:SOAP-ENC="http://www.w3.org/2003/05/soap-encoding"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:wsa5="http://www.w3.org/2005/08/addressing"
    xmlns:cp="urn://Ocpp/Cp/2012/06/"
    xmlns:cs="urn://Ocpp/Cs/2012/06/">
  <SOAP-ENV:Header>
    <wsa5:MessageID>
        [redacted]
      </wsa5:MessageID>
    <wsa5:From>
      <wsa5:Address>
          http://[redacted]
          </wsa5:Address>
        </wsa5:From>
    <wsa5:To
        SOAP-ENV:mustUnderstand="true">
        http://[redacted]/msr/Servicio/15/
        </wsa5:To>
    <wsa5:Action
        SOAP-ENV:mustUnderstand="true">
        /StatusNotification
        </wsa5:Action>
    <cs:chargeBoxIdentity>
        cargador1
        </cs:chargeBoxIdentity>
      </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cs:statusNotificationRequest>
      <cs:connectorId>
          [redacted]
          </cs:connectorId>
      <cs:status>
          Occupied
          </cs:status>
      <cs:errorCode>
          NoError
          </cs:errorCode>
        </cs:statusNotificationRequest>
      </SOAP-ENV:Body>
    </SOAP-ENV:Envelope>
```

Figura 5.18 – Trama de solicitud de actualización de estado del cargador (2) – detalle SOAP-XML.

En la imagen superior puede verse la solicitud de actualización de estado del cargador en *XML*; con los mismos datos en el sobre y la cabecera que en ejemplos anteriores [Figuras 5.6 y 5.9], aunque con un *MessageID* distinto.
En el cuerpo del mensaje se encuentra la información interesante, el campo *status* indica *Occupied*.

A partir del momento en que el servidor reciba la petición de modificación de disponibilidad y la acepte, se comienza la transacción de energía hacia el vehículo:

```
192.168.0.110   32.520881  93.156.191.36    HTTP/XML      452 HTTP/1.1 200 OK
```

Figura 5.19 – Trama de respuesta a la solicitud de actualización de estado del cargador (2).

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
        [HTTP/1.1 200 OK\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
  Content-Type: application/soap+xml; charset=utf-8\r\n
  SERVER: \r\n
  X-POWERED-BY: \r\n
  Date: Fri, 05 Jul 2019 09:44:10 GMT\r\n
  Connection: close\r\n
  Content-Length: 213\r\n
      [Content length: 213]
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.594297000 seconds]
  [Request in frame: 1680]
  [Request URI: http://[redacted]/msr/Servicio/15/]
  File Data: 213 bytes
```

Figura 5.20 – Trama de respuesta a la solicitud de actualización de estado del cargador (2) – detalle HTTP.

```
eXtensible Markup Language
    \357\273\277
  <?xml
      version="1.0"
      encoding="UTF-8"
      ?>
  <env:Envelope
      xmlns:env="http://www.w3.org/2003/05/soap-envelope"
      xmlns:ns1="urn://Ocpp/Cs/2012/06/">
    <env:Body>
        <ns1:statusNotificationResponse/>
        </env:Body>
      </env:Envelope>
```

Figura 5.21 – Trama de respuesta a la solicitud de actualización de estado del cargador (2) – detalle SOAP-XML.

De nuevo el servidor responde con un *200: OK*, por lo que se ha aprobado la transmisión y se comienza a realizar la recarga del *EV*.

5.3 Fin de la recarga

Para solicitar la terminación de la recarga es necesario volver a leer la tarjeta de fidelización del usuario del equipo de gestión que la ha iniciado.
Si se hubiera requerido el inicio de la recarga a través de la aplicación *EDP MoveOn*, se podría detener mediante la misma o utilizando la tarjeta, ya que esta está asociada al usuario que en ese momento está utilizando el *PdR* por lo que las credenciales de *ID* serían aceptadas por el servidor.

Figura 5.22 – Trama de solicitud de finalización de recarga.

Figura 5.23 – Trama de solicitud de finalización de recarga – detalle HTTP.

Para detener la recarga el *PdR* envía una *PDU StopTransaction* [Tabla 4.1] al servidor.

A continuación, se detalla el mensaje transmitido en formato *XML* en el protocolo *SOAP*:

```
eXtensible Markup Language
<?xml
    version="1.0"
    encoding="UTF-8"
    ?>
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
    xmlns:SOAP-ENC="http://www.w3.org/2003/05/soap-encoding"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:wsa5="http://www.w3.org/2005/08/addressing"
    xmlns:cp="urn://Ocpp/Cp/2012/06/"
    xmlns:cs="urn://Ocpp/Cs/2012/06/">
  <SOAP-ENV:Header>
    <wsa5:MessageID>
        [redacted]
        </wsa5:MessageID>
    <wsa5:From>
      <wsa5:Address>
          [redacted]
          </wsa5:Address>
        </wsa5:From>
    <wsa5:To
        SOAP-ENV:mustUnderstand="true">
        http://[redacted]/msr/Servicio/15/
        </wsa5:To>
    <wsa5:Action
        SOAP-ENV:mustUnderstand="true">
        /StopTransaction
        </wsa5:Action>
    <cs:chargeBoxIdentity>
        cargador1
        </cs:chargeBoxIdentity>
      </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cs:stopTransactionRequest>
      <cs:transactionId>
          [redacted]
          </cs:transactionId>
      <cs:idTag>
          [redacted]
          </cs:idTag>
      <cs:timestamp>
          2019-07-05T11:44:17Z
          </cs:timestamp>
      <cs:meterStop>
          0
          </cs:meterStop>
        </cs:stopTransactionRequest>
      </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

Figura 5.24 – Trama de solicitud de finalización de recarga – detalle SOAP-XML.

Para detener la recarga, el sobre del mensaje es igual que en la solicitud de inicio [Figura 5.12]. En la cabecera aparecen de nuevo ocultados los *ID* del mensaje transmitido y las direcciones *IP* de los elementos que participan, además de la acción que se insta a llevar a cabo: *Action /StopTransaction*.
En el cuerpo se indica la *PDU stopTransactionRequest* [Tabla 4.1].

Invariablemente, para que la operación se lleve a cabo debe ser aceptada por el servidor.

```
192.168.0.110  42.522433   93.156.191.36   HTTP/XML     542 HTTP/1.1 200 OK
```

Figura 5.25 – Trama de respuesta a la solicitud de finalización de recarga.

```
Hypertext Transfer Protocol
▾ HTTP/1.1 200 OK\r\n
  ▾ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Content-Type: application/soap+xml; charset=utf-8\r\n
  SERVER: \r\n
  X-POWERED-BY: \r\n
  Date: Fri, 05 Jul 2019 09:44:20 GMT\r\n
  Connection: close\r\n
▾ Content-Length: 303\r\n
    [Content length: 303]
  \r\n
  [HTTP response 1/1]
  [Time since request: 1.512105000 seconds]
  [Request in frame: 2050]
  [Request URI: http://[redacted]/msr/Servicio/15/]
  File Data: 303 bytes
```

Figura 5.26 – Trama de respuesta a la solicitud de finalización de recarga – detalle HTTP.

```
eXtensible Markup Language
  \357\273\277
▾ <?xml
    version="1.0"
    encoding="UTF-8"
    ?>
▾ <env:Envelope
    xmlns:env="http://www.w3.org/2003/05/soap-envelope"
    xmlns:ns1="urn://Ocpp/Cs/2012/06/">
  ▾ <env:Body>
    ▾ <ns1:stopTransactionResponse>
      ▾ <ns1:idTagInfo>
        ▾ <ns1:status>
            Accepted
            </ns1:status>
          </ns1:idTagInfo>
        </ns1:stopTransactionResponse>
      </env:Body>
    </env:Envelope>
```

Figura 5.27 – Trama de respuesta a la solicitud de finalización de recarga – detalle SOAP-XML.

El servidor comunica que acepta la finalización de la recarga enviando un *stopTransactionResponse* con código de estado *HTTP 1.1 200: OK*.

Al recibir el permiso para detener la recarga, automáticamente el *Punto de Recarga* envía una nueva solicitud al servidor, esta vez un *StatusNotification*, pues debe modificar su estado a “disponible”.

```
93.156.191.36  42.629827   192.168.0.110   HTTP/XML   1330 POST /msr/Servicio/15/ HTTP/1.1
```

Figura 5.28 – Trama de solicitud de actualización de estado del cargador (3).

```
Hypertext Transfer Protocol
  POST /msr/Servicio/15/ HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): POST /msr/Servicio/15/ HTTP/1.1\r\n]
        [POST /msr/Servicio/15/ HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: POST
      Request URI: /msr/Servicio/15/
      Request Version: HTTP/1.1
  Host: [redacted]\r\n
  User-Agent: gSOAP/2.8\r\n
  Content-Type: application/soap+xml; charset=utf-8; action="/StatusNotification"\r\n
  Content-Length: 998\r\n
      [Content length: 998]
  Connection: close\r\n
  Accept-Encoding: gzip, deflate\r\n
  SOAPAction: "/StatusNotification"\r\n
  \r\n
  [Full request URI: http://[redacted]/msr/Servicio/15/]
  [HTTP request 1/1]
  [Response in frame: 2348]
  File Data: 998 bytes
```

Figura 5.29 – Trama de solicitud de actualización de estado del cargador (3) – detalle HTTP.

El cargador transmite la petición *StatusNotification* [Tabla 4.1] al servidor para actualizar de nuevo su estado. Al haber sido aceptada la solicitud de finalización de recarga del *EV*, en las distintas ventanas de la aplicación en las que se muestra el conjunto de *PdR* instalados, el cargador que se estaría utilizando debería aparecer en color verde – activo y disponible para su uso.

En la imagen siguiente se indica el mensaje *SOAP* correspondiente a esta solicitud, donde se puede ver que ahora el estado del *Punto de Recarga cargador1* ha pasado a “disponible”.

```
eXtensible Markup Language
▼ <?xml
      version="1.0"
      encoding="UTF-8"
      ?>
▼ <SOAP-ENV:Envelope
      xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
      xmlns:SOAP-ENC="http://www.w3.org/2003/05/soap-encoding"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:wsa5="http://www.w3.org/2005/08/addressing"
      xmlns:cp="urn://Ocpp/Cp/2012/06/"
      xmlns:cs="urn://Ocpp/Cs/2012/06/">
   ▼ <SOAP-ENV:Header>
      ▼ <wsa5:MessageID>
            [redacted]
            </wsa5:MessageID>
      ▼ <wsa5:From>
         ▼ <wsa5:Address>
               http://[redacted]
               </wsa5:Address>
            </wsa5:From>
      ▼ <wsa5:To
            SOAP-ENV:mustUnderstand="true">
            http://[redacted]/msr/Servicio/15/
            </wsa5:To>
      ▼ <wsa5:Action
            SOAP-ENV:mustUnderstand="true">
            /StatusNotification
            </wsa5:Action>
      ▼ <cs:chargeBoxIdentity>
            cargador1
            </cs:chargeBoxIdentity>
         </SOAP-ENV:Header>
   ▼ <SOAP-ENV:Body>
      ▼ <cs:statusNotificationRequest>
         ▼ <cs:connectorId>
               [redacted]
               </cs:connectorId>
         ▼ <cs:status>
               Available
               </cs:status>
         ▼ <cs:errorCode>
               NoError
               </cs:errorCode>
            </cs:statusNotificationRequest>
         </SOAP-ENV:Body>
      </SOAP-ENV:Envelope>
```

Figura 5.30 – Trama de solicitud de actualización de estado del cargador (3) – detalle SOAP-XML.

En el cuerpo del mensaje, el cargador indica su estado: *cs:status Available.*

El servidor acepta la petición:

```
192.168.0.110  42.943626   93.156.191.36   HTTP/XML     452 HTTP/1.1 200 OK
```

Figura 5.31 – Trama de respuesta a la solicitud de actualización de estado del cargador (3).

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
        [HTTP/1.1 200 OK\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
  Content-Type: application/soap+xml; charset=utf-8\r\n
  SERVER: \r\n
  X-POWERED-BY: \r\n
  Date: Fri, 05 Jul 2019 09:44:20 GMT\r\n
  Connection: close\r\n
  Content-Length: 213\r\n
      [Content length: 213]
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.313799000 seconds]
  [Request in frame: 2289]
  [Request URI: http://[redacted]/msr/Servicio/15/]
  File Data: 213 bytes
```

Figura 5.32 – Trama de respuesta a la solicitud de actualización de estado del cargador (3) – detalle HTTP.

```
eXtensible Markup Language
    \357\273\277
  <?xml
      version="1.0"
      encoding="UTF-8"
      ?>
  <env:Envelope
      xmlns:env="http://www.w3.org/2003/05/soap-envelope"
      xmlns:ns1="urn://Ocpp/Cs/2012/06/">
    <env:Body>
        <ns1:statusNotificationResponse/>
        </env:Body>
      </env:Envelope>
```

Figura 5.33 – Trama de respuesta a la solicitud de actualización de estado del cargador (3) – detalle SOAP-XML.

Tras haber recibido la respuesta positiva anterior [Figuras 5.31, 5.32 y 5.33], el *PdR cargador1* termina la recarga y queda liberado, por lo tanto, libre para el siguiente usuario.

Utilizando la plataforma para realizar los accesos y peticiones, el usuario tan sólo habría interactuado para enviar “reserva”, “recarga” y “detención” a través de su dispositivo móvil, lo cual desencadena en las transacciones que se han ido comentando a lo largo de este capítulo. En cuanto a la conectividad con el vehículo, el coche debe ser enchufado antes de enviar la solicitud de recarga y sólo al conector que se había reservado con anterioridad; sólo podrá ser soltado una vez el *PdR* haya recibido una respuesta positiva a su solicitud *StopTransaction*.
En todo momento durante la recarga, el conector se encuentra bloqueado.

5.4 Otros ejemplos

A continuación, se añaden otras funcionalidades enviadas desde el mismo cargador hacia el servidor que se han tomado en una segunda captura. Esta vez no se citan solicitudes iniciadas por el usuario, si no que se incluyen algunas de las que se podrían catalogar como de disponibilidad, medidas y aprovisionamiento dentro de los bloques funcionales de *OCPP 2.0*, aunque se sigue utilizando la versión *1.6*.

5.4.1 Heartbeat

La *PDU Heartbeat* es enviada desde el *Punto de Recarga* hasta el servidor, funciona como una notificación para verificar del estado del cargador cada cierto período de tiempo [Tabla 4.1]. En el caso de la plataforma *EDP MoveOn*, cada uno de los cargadores instalados para el público envía un *Heartbeat* al servidor central cada quince minutos, de esta manera, aunque en ese período de tiempo ningún cliente lo haya utilizado, el servidor lo mantiene en su lista de cargadores disponibles y por tanto conservando su estado verde – activo y disponible para su uso en la aplicación móvil y la plataforma web. Si en el período de tiempo establecido, el servidor no recibe el mensaje *Heartbeat*, automáticamente el estado de ese cargador es modificado a “no disponible”.

```
93.156.191.36   66.096332   192.168.0.110   HTTP/XML   1204 POST /msr/Servicio/15/ HTTP/1.1
```

Figura 5.34 – Trama Heartbeat.

```
Hypertext Transfer Protocol
  POST /msr/Servicio/15/ HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): POST /msr/Servicio/15/ HTTP/1.1\r\n]
        [POST /msr/Servicio/15/ HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: POST
      Request URI: /msr/Servicio/15/
      Request Version: HTTP/1.1
    Host: [redacted]\r\n
    User-Agent: gSOAP/2.8\r\n
    Content-Type: application/soap+xml; charset=utf-8; action="/Heartbeat"\r\n
  Content-Length: 890\r\n
      [Content length: 890]
    Connection: close\r\n
    Accept-Encoding: gzip, deflate\r\n
    SOAPAction: "/Heartbeat"\r\n
    \r\n
    [Full request URI: http://[redacted]/msr/Servicio/15/]
    [HTTP request 1/1]
    [Response in frame: 978]
    File Data: 890 bytes
```

Figura 5.35 – Trama Heartbeat – detalle HTTP.

Es el *Punto de Recarga* quien envía la solicitud hacia el servidor, mediante una petición *POST* envía los datos de su estado actual en el *Heartbeat*.

```
eXtensible Markup Language
<?xml
    version="1.0"
    encoding="UTF-8"
    ?>
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
    xmlns:SOAP-ENC="http://www.w3.org/2003/05/soap-encoding"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:wsa5="http://www.w3.org/2005/08/addressing"
    xmlns:cp="urn://Ocpp/Cp/2012/06/"
    xmlns:cs="urn://Ocpp/Cs/2012/06/">
  <SOAP-ENV:Header>
    <wsa5:MessageID>
      [redacted]
      </wsa5:MessageID>
    <wsa5:From>
      <wsa5:Address>
        http://[redacted]
        </wsa5:Address>
      </wsa5:From>
    <wsa5:To
      SOAP-ENV:mustUnderstand="true">
      http://[redacted]/msr/Servicio/15/
      </wsa5:To>
    <wsa5:Action
      SOAP-ENV:mustUnderstand="true">
      /Heartbeat
      </wsa5:Action>
    <cs:chargeBoxIdentity>
      [redacted]
      </cs:chargeBoxIdentity>
    </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cs:heartbeatRequest>
      </cs:heartbeatRequest>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

Figura 5.36 – Trama Heartbeat – detalle SOAP-XML.

Al inicio del *XML* se incluye información respectiva al protocolo, versión y codificación de los datos. En la parte referente al mensaje *SOAP* se indica en el sobre la definición de todos los conceptos que intervienen y el protocolo *OCPP* usado en la capa de aplicación. La cabecera contiene los identificadores del mensaje y los elementos de inicio y final de la comunicación, además de información acerca de la acción que se desea llevar a cabo: *Heartbeat*.
En el cuerpo del mensaje es donde aparece la solicitud: *HeartbeatRequest.*

Para que el envío de la trama anterior se haga efectivo por completo, el servidor debe responder positivamente al recibimiento de la misma.

```
192.168.0.110   68.181052   93.156.191.36    HTTP/XML     521 HTTP/1.1 200 OK
```

Figura 5.37 – Trama de respuesta al Heartbeat.

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
        [HTTP/1.1 200 OK\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Content-Type: application/soap+xml; charset=utf-8\r\n
    SERVER: \r\n
    X-POWERED-BY: \r\n
    Date: Fri, 05 Jul 2019 09:35:15 GMT\r\n
    Connection: close\r\n
  Content-Length: 282\r\n
      [Content length: 282]
    \r\n
    [HTTP response 1/1]
    [Time since request: 2.084720000 seconds]
    [Request in frame: 800]
    [Request URI: http://[redacted]/msr/Servicio/15/]
    File Data: 282 bytes
```

Figura 5.38 – Trama de respuesta al Heartbeat – detalle HTTP.

```
eXtensible Markup Language
    \357\273\277
  <?xml
      version="1.0"
      encoding="UTF-8"
      ?>
  <env:Envelope
      xmlns:env="http://www.w3.org/2003/05/soap-envelope"
      xmlns:ns1="urn://Ocpp/Cs/2012/06/">
    <env:Body>
      <ns1:heartbeatResponse>
        <ns1:currentTime>
            2019-07-05T11:35:13Z
            </ns1:currentTime>
          </ns1:heartbeatResponse>
        </env:Body>
      </env:Envelope>
```

Figura 5.39 – Trama de respuesta al Heartbeat – detalle SOAP-XML.

Con la respuesta positiva del servidor, el estado del cargador queda como “disponible”, por tanto, visible y útil para los clientes de *EDP MoveOn*. En un período de quince minutos esta tarea será realizada de la misma manera.

5.4.2 BootNotification

La instrucción *BootNotification* contiene información sobre la configuración del cargador para que el servidor pueda verificarlo [Tabla 4.1]. Esta solicitud es enviada por el *PdR* tras el primer uso después de la instalación o cada vez que es reiniciado, y para que se finalice correctamente es necesario que entre el envío de la *BootNotificationRequest* y la pertinente respuesta del servidor no se envíe ningún otro mensaje entre ellos.
En el momento en que el cargador recibe la respuesta positiva del servidor, reajusta su reloj para comenzar desde ese momento a enviar en intervalos de quince minutos el *Heartbeat*.
Si el servidor rechaza al *Punto de Recarga,* este no podrá enviar ningún tipo de mensaje *OCPP* hasta que se haya reintentado la solicitud y esta sea aceptada.

Figura 5.40 – Trama BootNotification.

Figura 5.41 – Trama BootNotification – detalle HTTP.

```
eXtensible Markup Language
<?xml
     version="1.0"
     encoding="UTF-8"
     ?>
<SOAP-ENV:Envelope
     xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
     xmlns:SOAP-ENC="http://www.w3.org/2003/05/soap-encoding"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:wsa5="http://www.w3.org/2005/08/addressing"
     xmlns:cp="urn://Ocpp/Cp/2012/06/"
     xmlns:cs="urn://Ocpp/Cs/2012/06/">
  <SOAP-ENV:Header>
    <wsa5:MessageID>
         [redacted]
         </wsa5:MessageID>
    <wsa5:From>
      <wsa5:Address>
           http://[redacted]
           </wsa5:Address>
         </wsa5:From>
    <wsa5:To
         SOAP-ENV:mustUnderstand="true">
         http://[redacted]'msr/Servicio/15/
         </wsa5:To>
    <wsa5:Action
         SOAP-ENV:mustUnderstand="true">
         /BootNotification
         </wsa5:Action>
    <cs:chargeBoxIdentity>
         cargador1
         </cs:chargeBoxIdentity>
       </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cs:bootNotificationRequest>
      <cs:chargePointVendor>
           INGETEAM ENERGY
           </cs:chargePointVendor>
      <cs:chargePointModel>
           M1M32AM3M32A
           </cs:chargePointModel>
      <cs:chargePointSerialNumber>
           [redacted]
           </cs:chargePointSerialNumber>
      <cs:chargeBoxSerialNumber>
           [redacted]
           </cs:chargeBoxSerialNumber>
      <cs:firmwareVersion>
           gbcf55c3-ABA1005_O b32
           </cs:firmwareVersion>
      <cs:iccid>
           </cs:iccid>
      <cs:imsi>
           </cs:imsi>
      <cs:meterType>
           meterType
           </cs:meterType>
      <cs:meterSerialNumber>
           meterSerialNumber
           </cs:meterSerialNumber>
         </cs:bootNotificationRequest>
       </SOAP-ENV:Body>
     </SOAP-ENV:Envelope>
```

Figura 5.42 – Trama BootNotification – detalle SOAP-XML.

El cargador envía su información de identificación para ser verificado por el servidor: comerciante, modelo y números de serie, y su versión de firmware.

```
192.168.0.110   72.701386   93.156.191.36    HTTP/XML      617 HTTP/1.1 200 OK
```

Figura 5.43 – Trama de respuesta al BootNotification.

```
Hypertext Transfer Protocol
▾ HTTP/1.1 200 OK\r\n
  ▾ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Content-Type: application/soap+xml; charset=utf-8\r\n
  SERVER: \r\n
  X-POWERED-BY: \r\n
  Date: Fri, 05 Jul 2019 09:35:19 GMT\r\n
  Connection: close\r\n
▾ Content-Length: 378\r\n
    [Content length: 378]
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.223163000 seconds]
  [Request in frame: 1071]
  [Request URI: http://[redacted]/msr/Servicio/15/]
  File Data: 378 bytes
```

Figura 5.44 – Trama de respuesta al BootNotification – detalle HTTP.

```
eXtensible Markup Language
  \357\273\277
▾ <?xml
    version="1.0"
    encoding="UTF-8"
    ?>
▾ <env:Envelope
    xmlns:env="http://www.w3.org/2003/05/soap-envelope"
    xmlns:ns1="urn://Ocpp/Cs/2012/06/">
  ▾ <env:Body>
    ▾ <ns1:bootNotificationResponse>
      ▾ <ns1:status>
          Accepted
          </ns1:status>
      ▾ <ns1:currentTime>
          2019-07-05T11:35:19Z
          </ns1:currentTime>
      ▾ <ns1:heartbeatInterval>
          60
          </ns1:heartbeatInterval>
        </ns1:bootNotificationResponse>
      </env:Body>
    </env:Envelope>
```

Figura 5.45 – Trama de respuesta al BootNotification – detalle SOAP-XML.

Como se observa en las imágenes referidas a la respuesta del servidor [Figuras 4.43, 5.44 y 5.45], el servidor ha recibido la información por parte del cargador y lo ha validado, ya que en el cuerpo del mensaje *SOAP* el campo *status* aparece como *Accepted;* el *Punto de Recarga* tiene permisos para comenzar a enviar transacciones en *OCPP*.

5.4.3 MeterValues

El mensaje *MeterValues* es enviado por el *Punto de Recarga* con información relativa a ciertas medidas que haya ido tomando con el medidor de electricidad, proporcionando información sobre valores mesurables como pueda ser la potencia, voltaje o intensidad de una toma.
La decisión del envío de este mensaje queda en manos del *PdR* sin un intervalo de tiempo establecido, como así era en el ejemplo anterior. Cada mensaje de este tipo debe contener: información acerca del conector del que se han tomado las medidas, si este campo no se indica, la medida se refiere al cargador en general, la *transactionID* de la transacción a la que los valores enviados se refieren, campo que será omitido en caso de que no haya ninguna operación en proceso, y uno o más campos referidos a las medidas que se han tomado.
En el siguiente ejemplo, dado que no se realizaba ninguna operación al mismo tiempo, el campo *transactionID* no aparece, y la medida que se ha tomado hace referencia al *Punto de Recarga* en su conjunto.

```
93.156.191.36   113.153448   192.168.0.110    HTTP/XML    1473 POST /msr/Servicio/15/ HTTP/1.1
```

Figura 5.46 – Trama MeterValues.

```
Hypertext Transfer Protocol
  POST /msr/Servicio/15/ HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): POST /msr/Servicio/15/ HTTP/1.1\r\n]
        [POST /msr/Servicio/15/ HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: POST
      Request URI: /msr/Servicio/15/
      Request Version: HTTP/1.1
  Host: [redacted]\r\n
  User-Agent: gSOAP/2.8\r\n
  Content-Type: application/soap+xml; charset=utf-8; action="/MeterValues"\r\n
  Content-Length: 1154\r\n
      [Content length: 1154]
  Connection: close\r\n
  Accept-Encoding: gzip, deflate\r\n
  SOAPAction: "/MeterValues"\r\n
  \r\n
  [Full request URI: http://[redacted]/msr/Servicio/15/]
  [HTTP request 1/1]
  [Response in frame: 2845]
  File Data: 1154 bytes
```

Figura 5.47 – Trama MeterValues – detalle HTTP.

Figura 5.48 – Trama MeterValues – detalle SOAP-XML.

El *PdR cargador1* envía la trama *MeterValues* al servidor; en el fragmento que incluye los valores, *cs:values*, se indican la medida y sus unidades, la toma donde se ha realizado y su formato y contexto. El *Punto de Recarga* disponible en el centro de gestión en el que se han realizado las capturas tiene un único conector, en el cual se realizan las medidas, lo que viene indicado en el campo *cs:value location="Outlet".* El campo *cs:value measurand="Energy.Active.Import.Register"* muestra que se ha medido la energía consumida, valor que se muestra en Wh, como indica el campo *cs:value unit="Wh".* El formato *cs:value format="Raw"* significa que los datos obtenidos al realizar las medidas se representan en la forma definida por defecto, valores numéricos simples. Por último, el campo *cs:value context="Sample.Periodic"* muestra que las medidas se realizarán de forma periódica según cierto intervalo de tiempo en segundos.

Para terminar, se incluyen las imágenes referentes al correcto recibimiento de la información en el servidor y su respuesta positiva:

```
192.168.0.110    113.279792    93.156.191.36      HTTP/XML        445 HTTP/1.1 200 OK
```

Figura 5.49 – Trama de respuesta al MeterValues.

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
        [HTTP/1.1 200 OK\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
  Content-Type: application/soap+xml; charset=utf-8\r\n
  SERVER: \r\n
  X-POWERED-BY: \r\n
  Date: Fri, 05 Jul 2019 09:36:00 GMT\r\n
  Connection: close\r\n
  Content-Length: 206\r\n
      [Content length: 206]
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.126344000 seconds]
  [Request in frame: 2816]
  [Request URI: http://[redacted]/msr/Servicio/15/]
  File Data: 206 bytes
```

Figura 5.50 – Trama de respuesta al MeterValues – detalle HTTP.

```
eXtensible Markup Language
    \357\273\277
  <?xml
      version="1.0"
      encoding="UTF-8"
      ?>
  <env:Envelope
      xmlns:env="http://www.w3.org/2003/05/soap-envelope"
      xmlns:ns1="urn://Ocpp/Cs/2012/06/">
    <env:Body>
        <ns1:meterValuesResponse/>
        </env:Body>
      </env:Envelope>
```

Figura 5.51 – Trama de respuesta al MeterValues – detalle SOAP-XML.

Capítulo 6: Protocolo OICP

A continuación, se expone el protocolo *OICP*. Se presentará una estructura al igual que en el capítulo cuatro del presente documento para el protocolo *OCPP*, comenzando por una introducción de sus orígenes y pasando después a tratar la versión *OICP 2.2* en la plataforma de interoperabilidad *Hubject*.
Además, a partir del ámbito de utilización del protocolo, se introducirá brevemente el protocolo *OCPI (Open Charge Point Interface)* versión *2.1.1*, versión de formato abierto con la misma utilidad.

6.1 Historia – Hubject

Hubject es una plataforma *eRoaming* que surge en Alemania en el año 2012, formada por compañías del país como *BMW*, *Bosch*, *Daimler*, *EnBW*, *RWE* y *Siemens*. Su principal objetivo ha sido la instalación de una red de *Puntos de Recarga* en Europa, con la intención de establecer una red de itinerancia en la que cualquier usuario de *EV* pueda recargar su vehículo. Estos *PdR* actúan bajo la misma programación, existe un único tipo de contrato sin importar el país, el operador o el tipo de cargador; de esta forma se elimina la necesidad de que los usuarios de *EV* tengan que darse de alta en diferentes aplicaciones dependiendo de su localización.

Figura 6.1 – Logo Hubject.

Los primeros cargadores del sistema fueron instalados en Alemania, Bélgica, Holanda, Luxemburgo y Austria; y todos poseen un código *QR* mediante el cual el usuario puede acceder a las características de los mismos.
Actualmente cuenta con más de 300 socios y es la plataforma internacional más grande en el mercado *B2B*, con cerca de 140.000 *Puntos de Recarga* interconectados y repartidos en tres continentes.

Para la comunicación entre terminales y la integración del sistema surge un nuevo protocolo, diseñado por *Hubject* especialmente para esta tarea, *OICP*.

6.2 Funcionamiento

El propósito de la plataforma *B2B* de *Hubject* es hacer posible el mercado de la movilidad eléctrica proporcionando una puerta de enlace transaccional y de información para todo tipo de empresas que puedan intervenir, como proveedores de infraestructura de carga, proveedores de servicios de movilidad o fabricantes de vehículos.
El diagrama de comunicación es el siguiente:

Figura 6.2 – Esquema de comunicación en Hubject [9].

Como se puede ver en la imagen, existe un contrato entre el usuario de *EV* y su proveedor de servicio de movilidad. Estos proveedores se entienden como compañías cuyos clientes son simplemente los usuarios de los *EV*, no poseen infraestructura de recarga propia, si no que sirven a los usuarios como plataforma para encontrar *Puntos de Recarga* cercanos. Estos *PdR* estarán administrados por otras compañías, e instalados según su normativa, precios etc. La labor de *Hubject* reside básicamente en operar como plataforma intermedia y así hacer posible la comunicación entre los *CPO* y los *EMP (Electric Mobility Provider);* es una plataforma para la comunicación entre plataformas. De la misma manera que entre el cliente y el *EMP* existe un contrato, también lo hay entre este y el *CPO,* un contrato *eRoaming*, en el que se detallarían las pautas de participación de ambos dentro de *Hubject*.

El proceso de utilización del sistema es sencillo: los clientes que tienen un contrato con un *EMP* tienen la posibilidad de recargar su vehículo utilizando la infraestructura dispuesta por los *CPO* conectándose a la plataforma de *eRoaming HBS (Hubject Brokering System)*, la cual funciona como un mercado abierto para la movilidad.

Entre las tareas principales de la plataforma *Hubject* se incluye:

- Asegurar la interoperabilidad de la infraestructura pública mediante la promoción de estándares aceptados dentro de la red y la apertura de interfaces de usuario de negocios a la plataforma.
- Simplificación de los procedimientos de autenticación y autorización a través de una conexión confiable, así como la custodia de datos confidenciales.
- Automatización de las relaciones comerciales basadas en contratos entre proveedores de energía, fabricantes de automóviles o proveedores de servicios de infraestructura, así como otras partes comerciales dentro del mercado de la movilidad.
- Servicios de información *B2B* para la realización de servicios avanzados dentro de las áreas de energía, gestión, gestión del tráfico, reservas de vehículos, carga inteligente o uso compartido de vehículos.

Por lo tanto, los dos principales grupos asociados son por un lado los *EMP* que desean proporcionar acceso a sus clientes a la infraestructura de recarga, y por otro lado los propios operadores de esos *PdR*, los *CPO*, que desean aumentar la tasa de utilización de sus cargadores.
A continuación, se muestra un diagrama que ayuda a comprender el intercambio de mensajes en una solicitud de recarga:

Figura 6.3 – Esquema de comunicación del sistema en 'Recarga tipo 2'. Acceso usuario (extremo izquierdo), servidor central EDP (izquierda), conexión a través de la plataforma Hubject con el operador del PdR (derecha), Punto de Recarga (extremo derecho).

Tomando como ejemplo las anteriormente denominadas 'Recargas tipo 2' y a *EDP* como *EMP*, el funcionamiento sería el siguiente: el cliente *EDP* envía una solicitud de recarga y comienzan las comunicaciones entre los dispositivos. Ahora la solicitud del cliente se transmite al servidor de su proveedor de energía, en este caso el servidor central de *EDP*, y de este al servidor del operador del *Punto de Recarga* en el que se solicita realizar esta. Esta tarea es la que recae sobre la plataforma *Hubject,* utilizando el protocolo *OICP*. Por último, el operador transmite la solicitud a su cargador, que responde a esta y, si es positivamente, se procede a la recarga del vehículo.
De la misma manera podría encontrarse a *EDP* en el papel de *CPO* en lugar de *EMP*, si es un cliente extranjero, con otra compañía como *EMP,* quien desea utilizar la infraestructura de *EDP MoveOn*.

6.3 Versión OICP 2.2

Esta versión data de marzo de 2018 [9]. Una de las particularidades más notables de la versión *OICP 2.2*, a diferencia de las que le preceden, es que ya no permite la utilización del protocolo *SOAP* como protocolo de transporte, por lo que las transferencias se basan en *REST*. Esto obliga a los *CPO* interesados en participar a implementar *APIs* que soporten *REST* para sus *Puntos de Recarga,* para poder asegurar una correcta comunicación a través de la plataforma *HBS.*

Dado que un gran número de *Puntos de Recarga* instalados se basan en *SOAP*, como puede ser el caso de algunos *PdR* de *EDP*, *OICP 2.2* tiene capacidad para tramitar errores en las solicitudes que no lleguen en formato *REST*. Si se recibe una petición *SOAP*, el servidor que la ha enviado recibirá una respuesta por parte de *Hubject*, en esa misma configuración, indicando la violación del formato estándar.
La mayoría de los mensajes de respuesta contienen un campo de código de estado, que sirve para devolver detalles sobre la condición actual de un proceso, por ejemplo: si una solicitud de autorización falla, el proveedor al que se le ha solicitado puede especificar en la respuesta por qué la petición no puede ser validada; no es necesario analizar el estado funcional del sistema en general.
Hubject asigna una *session ID* a cada sesión después del recibimiento de una solicitud de autorización. Este campo se incluye en la respuesta a la petición inicial, y sirve para poder relacionar las operaciones individuales llevadas a cabo por los *PdR* con la sesión correcta.
Debido a que la función principal de la plataforma es dar servicio a usuarios de *EV* cuando no se encuentran en su país de origen, la mayoría de las operaciones incluyen los campos de identidad *ProviderID* y *OperatorID*. El primero consta de un código para identificar el país y una cadena de tres dígitos, el segundo es una composición de un código identificativo del país y una cadena de tres a seis dígitos. Ambos sirven para reconocer internacionalmente el rol del *CPO* o *EMP* en concreto en cada sesión. Estas identificaciones son únicas y utilizadas por *Hubject* para saber el papel de cada interlocutor en cada transacción.

6.3.1 Recarga en OICP 2.2

A continuación, se indican los principales mensajes que se intercambian entre el *CPO*, *Hubject* y el *EMP* para la recarga de un vehículo.

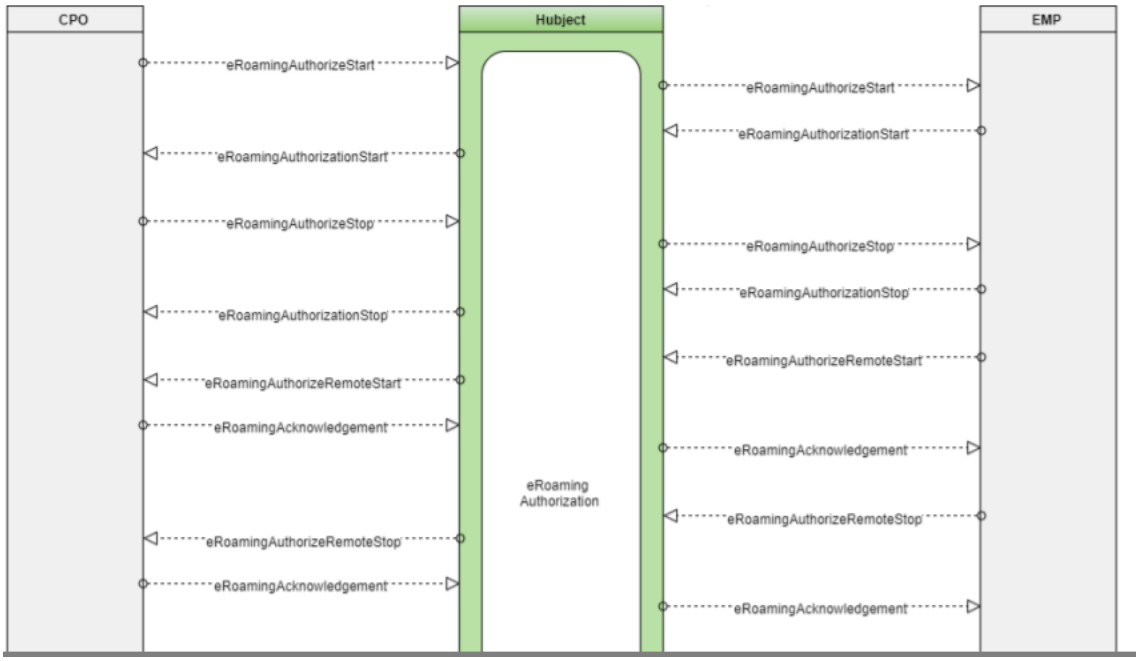

Figura 6.4 – Intercambio de mensajes en una solicitud de recarga – Hubject [9].

Para iniciar, se aclara el intercambio de mensajes tomando como ejemplo una recarga común.
Un cliente de una plataforma *EMP* desea recargar su vehículo utilizando la infraestructura instalada por un *CPO*. El cliente debe autenticarse ante el *PdR* para poder utilizar el sistema, de modo que el cargador envía un *eRoamingAthorizeStart* a *Hubject* para verificar y reconocer al cliente. Esta petición debe incluir el *ID* de la operación a realizar, además de otros datos de identificación. En este momento, *Hubject* genera la *session ID* para identificar todos los mensajes del proceso de carga que se va a llevar a cabo. Antes de reenviar la solicitud al *EMP*, la plataforma *HBS* chequea su base de datos e intenta identificar al usuario, por lo que la respuesta a la petición de autorización del *CPO* puede ser enviada sin necesidad de comunicación con el *EMP*. En caso de que esto no sea posible, en base a la información de identificación añadida en la petición, *Hubject* se comunicará con el *EMP* pertinente. Este responderá a la petición inicial con un *eRoamingAuthorizationStart* que enviará a *Hubject* y este a su vez al *CPO* para que sea comunicado al *Punto de Recarga*. El cliente ha sido autorizado y puede proceder a la recarga de su *EV*. Para detener la recarga, el intercambio de mensajes es el mismo: el *eRoamingAuthorizeStop* enviado para detener la recarga debe contener el *session ID* creado por *Hubject* tras la primera solicitud. La solicitud para terminar la recarga, al igual que se ha comentado antes, puede ser directamente gestionada por *Hubject* si no hubo necesidad de comunicarse con el *EMP* para autorizar el cliente, y, de hecho, debe ser así si el *EMP* no ha intervenido.

Otro proceso de recarga puede darse de la siguiente manera: un usuario quiere recargar su vehículo en un *PdR* e informa de su intención a su *EMP* a través de una aplicación móvil. Al conocer la intención de su cliente, el *EMP* inicia el proceso de recarga enviando

a *Hubject* un mensaje *eRoamingAuthorizeRemoteStart,* donde se incluye información del proveedor de movilidad, así como del vehículo. *Hubject* reenvía la solicitud al *CPO* del cargador a utilizar, que responde con un *eRoamingAckowledgement* para permitir la operación.
Para terminar la recarga, el usuario vuelve a informar al *EMP* utilizando la aplicación, enviando este un mensaje *eRoamingAuthorizeRemoteStop*, el cual se vuelve a responder con un *eRoamingAckowledgement* por parte del *CPO*.

6.4 Protocolo OCPI

A partir del éxito de la plataforma *Hubject* y de su protocolo *OICP*, aparece el protocolo *OCPI,* protocolo que trabaja de la misma manera [10] que el anterior, pero siendo la versión abierta independiente del mismo.
Los primeros borradores, *OCPI 0.3* y *OCPI 0.4*, aparecen dos años más tarde que la primera versión del *OICP*. La primera versión lanzada al público es el *OCPI 2.0,* surgida en diciembre del año 2015; desde enero de 2017 se trabaja con la última versión, *OCPI 2.1.1*.

OCPI permite una configuración de itinerancia de vehículos eléctricos automatizada entre *CPOs* y *EMPs*. Soporta mensajes de autorización, intercambios de información sobre los cargadores, de registros con detalles de las recargas, envío de mensajes de forma remota o transacciones “Smart Charging”. Con esto, ofrece una solución escalable para el *roaming* entre redes, evitando los costes y limitaciones que puedan aparecer con la participación de intermediarios.

Sus principales funcionalidades son:

- Ofrece un sistema de *roaming* bilateral.
- Información sobre la ubicación, disponibilidad y precio en tiempo real.
- Una manera uniforme de intercambio de datos antes, durante y después de la operación.
- Soporte móvil remoto para acceder a cualquier estación de recarga sin necesidad de realizar un registro previo.

Figura 6.5 – Logo OICP.

Actualmente, un gran número de empresas internacionales apoyan la iniciativa de este protocolo, entre ellas *ElaadNL*, compañía creadora del protocolo *OCPP*.

El protocolo *OCPI* está basado, como los comentados hasta ahora, en *HTTP*, y utiliza el formato *JSON*. Se basa en una arquitectura *REST* sobre webServices cuando es posible.

6.4.1 Ejemplo de aplicación

Para finalizar, se presenta un ejemplo de solicitud para conocer qué *Puntos de Recarga* instalados por *EDP* serían capaces de soportar comunicaciones *OCPI* para recargar. Aunque hoy en día estas no sean las recargas más solicitadas, *EDP MoveOn* sigue evolucionando y actualizando sus cargadores para que formen parte de nuevos proyectos internacionales que van surgiendo.

La imagen siguiente corresponde a la respuesta del servidor central de *EDP*, en la que, en la organización de árbol, cada objeto principal corresponde a un *Punto de Recarga* de *EDP*.

Figura 6.6 – LocationRequest en OCPI.

Se toma para el ejemplo el objeto 31, el *Punto de Recarga* 31. Al desplegarlo, aparecen primero sus datos de identificación: número *ID*, tipo de instalación, nombre y ubicación. Se trata de un cargador instalado en la calle, en Posada de Llanes (Asturias). Además de estos datos, aparecen más objetos para desplegar: sus coordenadas de localización, información acerca del estado del cargador, la zona horaria en la que trabaja y la fecha de la última actualización de su software.
Se observa que el cargador se encuentra fuera de servicio, que se puede acceder a la recarga leyendo la tarjeta *RFID* del usuario, está configurado dentro de la zona horaria de Europa central (Madrid) y su última actualización se realizó el 3 de abril de 2019. Accediendo al objeto que describe los conectores, se aprecia que el cargador dispone de tres, siendo uno de ellos de carga rápida, un CHAdeMO que trabaja con corriente continua.

Capítulo 7: Conclusiones y líneas futuras

Durante el desarrollo del presente documento, queda constatado que el concepto de movilidad sostenible y la amplia evolución y empleo de vehículos eléctricos que deriva de ella es una realidad.

Es probable que la sociedad no sea aún totalmente consciente del rápido proceso de desarrollo que este sector está sufriendo, que en tan sólo diez años la tecnología haya influido tanto en el automovilismo e incluso esta sea la base de la fabricación de los modelos de transporte futuros.
Partiendo de simples borradores y documentos de prueba, con vehículos con autonomía limitada y escasas características de conectividad, se ha llegado a la estandarización de métodos y protocolos internacionalmente aceptados para garantizar una homogeneidad que permita la evolución de la *eMobility*.
Basándose en los protocolos en los que hoy en día se basa Internet, se han elaborado sistemas de comunicación y transferencia de datos capaces de transmitir de forma segura información que pasa por *smartphones*, cargadores, vehículos y plataformas en segundos, y que consiguen recargar vehículos en minutos.

En puntos de este trabajo se mencionan protocolos cuyas últimas versiones datan de hace escasos meses de este mismo año, y que se encuentran en constante crecimiento para adaptarse a las necesidades que los, cada vez más, usuarios de *EV* solicitan, al mismo tiempo que se transforman a la par que la ingeniería electrónica; por lo que se puede asegurar que el vehículo eléctrico será el vehículo del mañana.

Bibliografía

[1]
Introducción – Energías de Portugal
https://espana.edp.com/es/node/32538
https://espana.edp.com/es/node/30217

[2]
Infraestructura – Tipos de cargadores
https://edpmoveon.com/home/index.php

[3]
"Understanding OCPP", OCPP Fact Sheet. Charge Point, Campbell, California (2013)

[4]
Buve F., *"Open Charge Point Protocol 1.5"*, OCPP 1.5 Specification. Holanda (2012)

[5]
Rademakers P., de Leeuw R., Lamers R., Giaume C., Warnier O., "Open Charge Point Protocol SOAP 1.6", OCPP-S 1.6 Specification. Open Charge Alliance, Holanda (2015)

[6]
de Leeuw, R., Lamers R., McMahon B., Muhlenberg L., Rademakers P., Tcaciuc S., van Zuuren K., *"Open Charge Point Protocol 1.6"*, OCPP 1.6 Specification. Open Charge Alliance, Holanda (2015)

[7]
Rademakers P., de Leeuw R., Lamers R., *"Open Charge Point Protocol JSON 1.6"*, OCPP-J 1.6 Specification. Open Charge Alliance, Holanda (2015)

[8]
"Open Charge Point Protocol 2.0", OCA OCPP 2.0 Specification. Open Charge Alliance, Holanda (2018)

[9]
"Open InterCharge Protocol for Charge Point Operators version 2.2", Hubject GmbH, Alemania (2018)

[10]
"Open Charge Point Interface 2.1.1, document version: 2.1.1-RC1", Netherlands Knowledge Platform, Países Bajos (2017)