

<b>SD</b>	<b>Sistemas Distribuidos</b>
<b>25/26</b>	Práctica no guiada: Sockets, Streaming de Eventos, Colas y modularidad.
	<b>EVCharging Network</b>

## Preámbulo

El objetivo de esta práctica es que los estudiantes extiendan y afiancen sus conocimientos sobre el uso de la tecnología de comunicación básica sockets, estudiada durante las sesiones de teoría.

Los sockets son la base de las comunicaciones entre computadores y suponen el punto de enlace básico entre nuestra aplicación e Internet, utilizando muy pocos recursos de red. Como contrapartida, las aplicaciones que utilizan sockets como mecanismo de comunicación deben interpretar los mensajes que intercambian entre sí, es decir, deben establecer un protocolo o acuerdo de comunicación entre ellos. Esta necesidad implica un fuerte acoplamiento entre las aplicaciones distribuidas que utilizan sockets como mecanismo de comunicación, ya que todas las partes deben conocer de antemano la estructura de los mensajes que serán intercambiados y codificarlo en su programación.

Por otro lado, otras tecnologías y arquitecturas actuales orientadas al streaming de eventos en tiempo real, basadas en sockets pero de más alto nivel en términos de construcción y desarrollo, permiten la implementación de soluciones menos acopladas y más resilientes que estos.

Esta práctica establece el marco inicial de un desarrollo que se ampliará durante el cuatrimestre y que permitirá al estudiante crear una aplicación similar a las que se desarrollan hoy en día en los entornos empresariales, poniendo el foco en el uso e integración de distintos paradigmas de comunicación susceptibles de ser utilizados.

## Movilidad eléctrica

En la actualidad, la movilidad eléctrica se presenta como una de las principales soluciones para reducir las emisiones contaminantes y avanzar hacia un modelo de transporte más sostenible. El crecimiento del parque de vehículos eléctricos en los últimos años ha impulsado la necesidad de desplegar una red cada vez más amplia y eficiente de puntos de recarga, que permita garantizar la autonomía de los usuarios y facilitar su uso cotidiano.

Sin embargo, la simple instalación de puntos de recarga no resulta suficiente. Su gestión adecuada es un aspecto crítico para asegurar la disponibilidad, fiabilidad y eficiencia del servicio. Factores como la localización estratégica de los cargadores, la gestión de la potencia suministrada, la monitorización del estado de cada estación y la detección temprana de incidencias técnicas resultan esenciales para optimizar la experiencia del usuario y la sostenibilidad económica de la red.

Además, el control y la monitorización de estos puntos permiten integrar herramientas de planificación energética, detectar patrones de consumo, evitar sobrecargas en la red eléctrica y facilitar la integración con fuentes de energía renovable. Una supervisión eficaz no solo garantiza el correcto funcionamiento de los equipos, sino que también aporta datos valiosos para la toma de decisiones, tanto desde la perspectiva técnica como desde la económica y ambiental.

En este contexto, la práctica que se propone busca acercar al estudiante al reto de implementar los conceptos teóricos de la asignatura de sistemas distribuidos en una problemática real, poniendo en valor la importancia de gestionar y supervisar una red de recarga de manera integral, resiliente, escalable y segura. Se trata de comprender cómo las distintas tecnologías que se irán estudiando, su interconexión y la planificación se convierten en pilares fundamentales para avanzar hacia un sistema de transporte más eficiente, accesible y respetuoso con el medioambiente.

## Especificación

El objetivo de la práctica es desarrollar un sistema distribuido que implemente una simulación de una solución para la gestión de una red de puntos de recarga de vehículo eléctrico.

## Descripción funcional

Los estudiantes deberán implementar una solución de control online en tiempo real denominada “EVCharging”.

### Alcance

La solución tendrá como objetivo realizar un sistema que simula la gestión de una red de postes de recarga (en adelante también “Charging Point” o CP) de vehículos eléctricos y su uso tanto por los dueños de la red como por los usuarios finales que acceden a las recargas de sus vehículos.



*Imagen 1: Estaciones de recarga de la empresa Zunder*

Nuestra simulación consistirá en, partiendo de un mapa de una ciudad o país ficticio, dispondremos de una serie de puntos de recarga de vehículos eléctricos en los que los usuarios (conductores), provistos de una aplicación podrán hacer uso de los mismos para recargar sus vehículos.

Todos los puntos de recarga son gestionados desde una central que habilitará, operará y monitorizará los mismos en caso de que se produzcan incidencias en los mismos.

Para la realización de todo el sistema, nos basaremos en una serie de componentes distribuidos que estarán interconectados y que conformarán la solución de manera resiliente, escalable y segura.

## CENTRAL. Sistema de monitorización de la red

Dispondremos de un cuadro de mandos o panel de monitorización donde se podrá observar, el detalle de funcionamiento en tiempo real de toda el sistema incluyendo los puntos de recarga, las peticiones de suministro y el estado del sistema central.

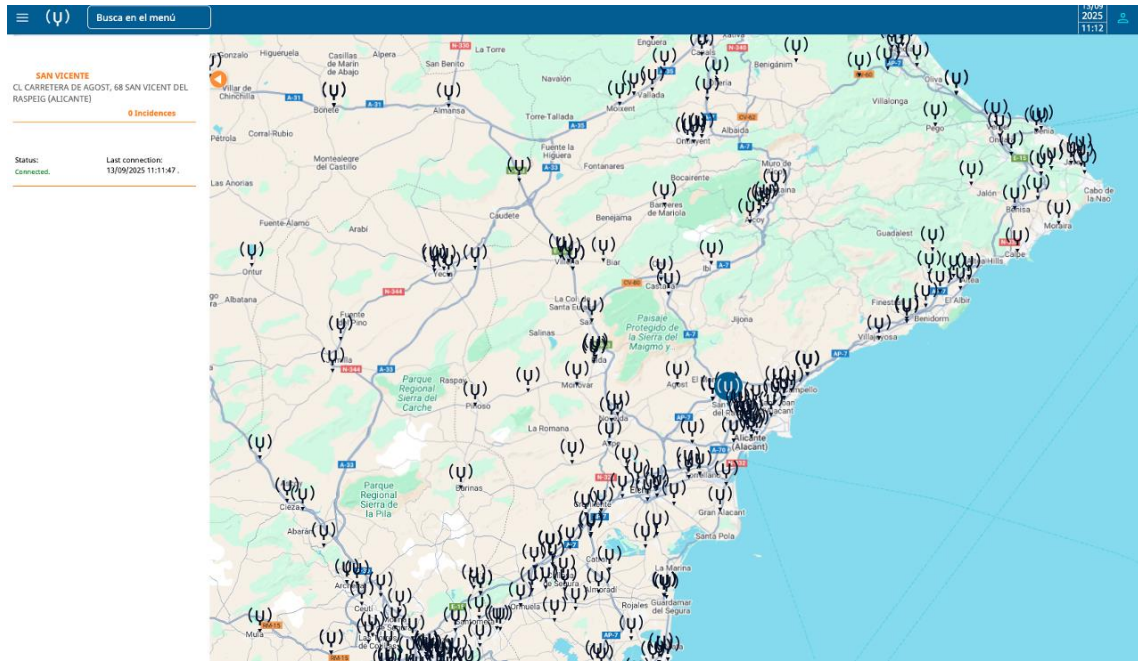


Imagen 2: Ejemplo de sistema de monitorización de estaciones

En el panel que muestre la información de detalle de consumo la información a visualizar será la siguiente:

- ID del punto de recarga.
- Ubicación del punto de recarga.
- Precio al que tiene el suministro en €/Kwh.
- Estado:
  - **Activado (disponible)** : El CP está funcionando correctamente y a la espera de una solicitud de recarga. Se mostrará mediante un fondo de color VERDE.
  - **Parado (fuera de servicio)**: El CP está en correcto funcionamiento pero no puede usarse deliberadamente por orden de la CENTRAL. Se mostrará mediante un fondo de color NARANJA y con la leyenda “Out of Order”.
  - **Suministrando**: El punto de recarga está funcionando correctamente y está, en ese momento, suministrando energía a un determinado vehículo. Se mostrará mediante un fondo de color VERDE y con los siguientes datos impresos:
    - Consumo en Kw en tiempo real.
    - Importe en € en tiempo real.
    - Id del conductor al que se está suministrando.<sup>1</sup>

<sup>1</sup> En la práctica profesional existen sistema de identificación de reconocimiento de matrícula (Automatic Number Plate Recognition ANPR). Sin embargo, esta identificación ni ninguna otra de carácter personal deben usarse para mostrarla en pantalla. La normativa GDPR prohíbe expresamente su uso.

- **Averiado** : El punto de recarga está conectado al sistema central pero presenta una avería y no es posible su uso. Se mostrará mediante un color ROJO de fondo.
- **Desconectado**: El punto de recarga no está conectado al sistema central (por avería o por fallo de red o por cualquier otro motivo) y no es posible monitorizar su uso. Se mostrará mediante un color GRIS de fondo.

*** SD EV CHARGING SOLUTION. MONITORIZATION PANEL ***				
ALC1	ALC3	MAD2	MAD3	MAD1
C/Italia 5 0,54€/kWh	Gran Vía 2 0,54€/kWh	C/Serrano 10 0,6€/kWh Driver 5 10,3kWh 6,18 €	C/Pez 23 0,48€/kWh	C/Arguelles 0,4€/kWh
SEV3	SEV2	VAL8	VAL1	COR1
C/Sevilla 4 0,54€/kWh Driver 24 50kWh 27 €	Maestranza 0,4€/kWh	Museo Arts 0,5€/kWh	San Javier 0,48€/kWh	Mezquita 0,4€/kWh Driver 234 20kWh 8 €
*** ON_GOING DRIVERS REQUESTS ***				
DATE	START TIME	User ID	CP	
12/9/25	10:58	5	MAD2	
12/9/25	9:00	23	SEV3	
12/9/25	9:05	234	COR1	
*** APPLICATION MESSAGES ***				
ALC3 out of order				
SEV2 out of order				
CENTRAL system status OK				

Figura 1: Ejemplo de panel de control del sistema central

## CONDUCTORES

Los usuarios (conductores de los vehículos) del servicio dispondrán de una aplicación que permitirá recargar su vehículo en un punto determinado. Así, dicha aplicación enviará a la central la petición de un servicio y esta le responderá autorizándolo.

El sistema central, como se verá en posteriores apartados, deberá velar porque el punto de recarga está disponible antes de autorizar el suministro.

## PUNTOS DE RECARGA

Puntos de recarga en los cuales los conductores pueden recargar sus vehículos.

Los estados en los que puede entrar un CP son los expresados anteriormente en el panel de control de la central.

Los puntos de recarga permiten las siguientes funcionalidades:

- Registrarse en la central: Enviarán para ello su Id y ubicación a la central.
- Activarse o pararse.
- Suministrar: proporcionar un servicio de recarga de dos formas:

- Manualmente mediante una opción en el propio punto.
- A través de una petición proveniente de la aplicación del conductor (vía central como se verá más adelante).

Estos puntos de recarga disponen a su vez de un módulo independiente que permite monitorizar localmente el hardware y software local para controlar su buen funcionamiento o posible avería.

## Mecánica de la solución

El sistema comienza mediante la ejecución del módulo central (en adelante “CENTRAL”) que permanecerá en ejecución sin límite de tiempo sin apagarse en ningún momento (salvo que una excepción o bloqueo súbito provoque su apagado no controlado).

Tendrá lugar entonces la siguiente secuencia:

- 1- Ante cualquier ejecución o reinicio, CENTRAL comprobará (en su BD) si ya tiene puntos de recarga disponibles registrados (con su ubicación) y los mostrará en su panel de monitorización y control en su estado correspondiente. **IMPORTANTE:** hasta que un punto de recarga no conecte con CENTRAL esta no podrá conocer el estado real del punto. En ese caso, lo mostrará con el estado **DESCONECTADO**.
- 2- CENTRAL siempre estará a la espera de:
  - a. Recibir peticiones de registro y alta de un nuevo punto de recarga.
  - b. Recibir peticiones de autorización de un suministro.
- 3- Los conductores, desde su aplicación o desde un menú en el propio CP, solicitarán un suministro en cualquier punto de recarga. Con el objetivo académico de automatizar las pruebas del sistema, además de poder solicitar un servicio manualmente, la aplicación del conductor también podrá leer los servicios de recarga a solicitar desde un archivo con el siguiente formato:  
    <ID\_CP>  
    <ID\_CP>  
    ...
- 4- CENTRAL procederá a realizar las comprobaciones oportunas para validar que el punto de recarga esté disponible y, en su caso, solicitará autorización al punto de recarga para que proceda al suministro. Todo el proceso requerirá de la notificación al conductor de los pasos que van sucediendo hasta autorizar o denegar el suministro. Dichos mensajes **se deben mostrar claramente en pantalla, tanto en la aplicación del cliente como de CENTRAL**.
- 5- Los puntos de recarga, **una vez se han registrado y conectado a la central**, estarán en estado de reposo a la espera de que un conductor solicite, **bien en el propio interfaz del punto de recarga o a través de su aplicación, un suministro**.
- 6- Realizadas todas las comprobaciones por la CENTRAL y enviada la notificación tanto al CP como a la aplicación del conductor (que lo verá en pantalla) para que procedan al suministro, el conductor enchufará su vehículo al CP.
- 7- Para simular este acto en el cual un conductor enchufa su vehículo a un CP, el CP dispondrá de una opción de menú. Al ejecutar esta opción se entenderá que la conexión ha sido exitosa y empezará el suministro.
- 8- Durante el suministro, el punto de recarga estará enviando información constante (**cada segundo**) a CENTRAL, la cual mostrará en su panel de monitorización el consumo e importes que va teniendo lugar en cada CP tal y como se indicó en

apartados anteriores. La aplicación del conductor igualmente mostrará esos estados del CP que le está suministrando.

- 9- Finalizado el suministro mediante una opción de menú en el CP que simula el desenchufado del vehículo del CP, este lo notificará a CENTRAL quien a su vez enviará el “ticket” final al conductor<sup>2</sup>. El CP volverá al estado de reposo a la espera de una nueva petición de suministro.
- 10- Durante todo el funcionamiento del sistema, el punto de recarga, desde su módulo de monitorización estará comprobando el estado de salud del punto de recarga. En caso de avería, notificará a CENTRAL dicha situación. En caso de que la contingencia se resuelva, el monitor enviarán un nuevo mensaje a CENTRAL y el punto de recarga volverá a estar disponible. Si la avería se produjera durante un suministro, este debe finalizar inmediatamente informando a todos los módulos y aplicaciones de dicha situación. **Toda la información tiene que ser claramente visible en pantalla tanto de CENTRAL como del monitor del punto de recarga.**
- 11- CENTRAL recibirá todos los estados y suministros de todos los CPs. Los conductores solo recibirán los mensajes del CP específico que les esté proporcionando un servicio pero también podrán ver todos los CPs que estén disponibles para suministrar.
- 12- Cuando un suministro concluya, si dicho conductor precisa de otro servicio (tiene más registros en su fichero) **el sistema esperará 4 segundos y procederá a solicitar un nuevo servicio.**
- 13- CENTRAL permanecerá siempre en funcionamiento a la espera de nuevas peticiones de suministros. Adicionalmente dispondrá de opciones para, de forma arbitraria, poder enviar a uno o todos los CPs cualquiera de estas opciones:
  - a. Parar: El CP finalizará cualquier suministro si estuviera en ejecución y pondrá su estado en ROJO mostrando el mensaje de “Fuera de Servicio”.
  - b. Reanudar: El CP volverá a estado ACTIVADO y pondrá su color en VERDE.

La acción se va desarrollando de forma asíncrona, es decir, sin “turnos”, de manera que cada conductor, CENTRAL y cada CP envían mensajes o peticiones de suministros en cualquier momento.

---

<sup>2</sup> En un suministro de un sistema real se habría incorporado todo un flujo de identificación y autorización o denegación del sistema de pago mediante integración con pasarela de pago. Los mecanismos de interconexión con este sistema se han ido realizando a lo largo de los años tanto en sistemas basados en sockets como en llamadas a API's (tecnología que veremos en la segunda parte de la práctica). A efectos de esta práctica, dado el plazo limitado disponible para su desarrollo, omitiremos este módulo.



## Diseño técnico

Se propone al estudiante implementar un sistema distribuido compuesto, al menos, de los siguientes componentes software:

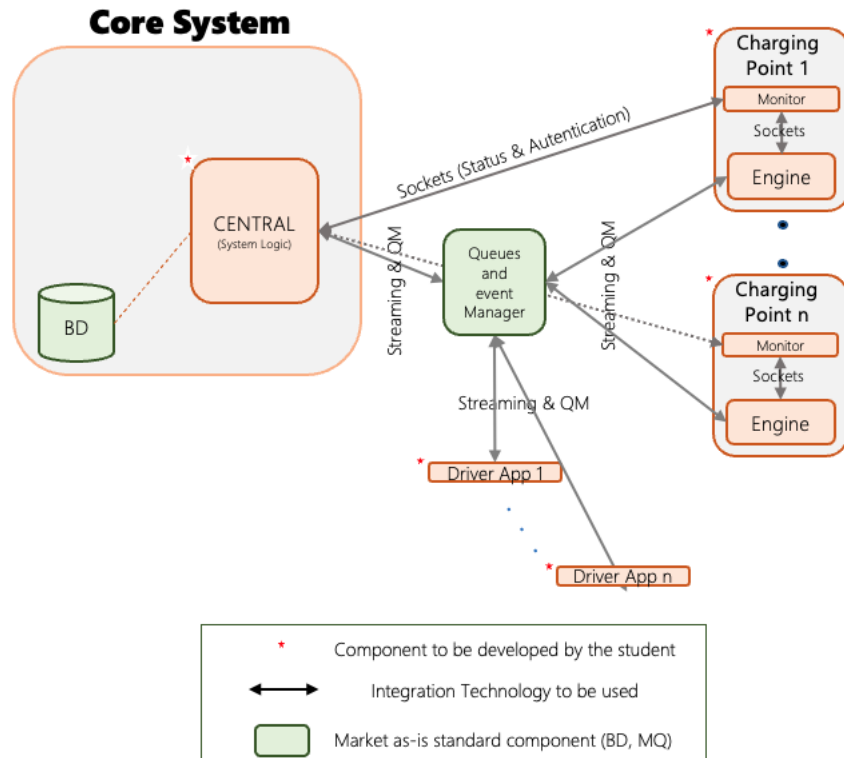


Figura 2. Arquitectura conceptual del Sistema software, interconexiones entre los componentes y tipos de interfaces. Release 1

Los componentes software que el estudiante ha de desarrollar son los siguientes:

- **Core System:** Módulos centrales de la solución.
  - **CENTRAL:** Módulo que representa la central de control de toda la solución. Implementa la lógica y gobierno de todo el sistema.
- **Charging Point (CP):** Módulos que simulan los puntos de recarga
  - Engine: Módulo que recibe la información de los sensores y se conecta al sistema central.
  - Monitor: Módulo que monitoriza la salud de todo el punto de recarga y que reporta a la CENTRAL cualquier avería de este. Sirve igualmente para autenticar y registrar a los CP en la central cuando sea oportuno.
- **Driver:** Aplicación que usan los consumidores para usar los puntos de recarga.

Los componentes pueden ser desarrollados en el lenguaje de preferencia del estudiante: Java, C/C++, . NET, Python, etc. asumiendo el estudiante la responsabilidad de su conocimiento y forma de desplegarlo en el laboratorio.

El subsistema de gestión de streaming de eventos será implementado mediante la tecnología KAFKA cuyo funcionamiento se explicará en las sesiones de prácticas.

A continuación, se especifica más detalladamente cada componente.

## CORE

Contiene todos los módulos que implementan la estructura principal del sistema.

### CENTRAL:

Se trata de la aplicación que implementará la lógica fundamental del sistema. El nombre de la aplicación será **obligatoriamente “EV\_Central”**. Para su ejecución recibirá por la línea de parámetros al menos los siguientes argumentos:

- Puerto de escucha
- IP y puerto del Broker/Bootstrap-server del gestor de colas.
- IP y puerto de la BBDD (solo en caso de que sea necesario)

Al arrancar la aplicación quedará a la escucha en el puerto establecido e implementará todas las funcionalidades necesarias para ejecutar la mecánica de la solución expresada en el apartado anterior **incluidas las indicadas en el punto 13** del apartado “Mecánica de la solución”.

### Base de Datos:

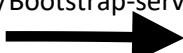
Contendrá, los datos de los conductores, CPs (Id, estado,...) así cualquier otra información que los estudiantes consideren necesaria para la implementación del sistema.

Los estudiantes podrán decidir el motor de BD a implementar recomendando los profesores los siguientes: SQLite, MySQL, SQLServer o MongoDB. **Igualmente será posible usar simples ficheros para la implementación.**

## Charging Point (CP)

### Engine:

Se trata de la aplicación que implementará la lógica principal de todo el sistema. El nombre de la aplicación será **obligatoriamente “EV\_CP\_E”**. Para su ejecución recibirá por la línea de parámetros al menos los siguientes argumentos:

- IP y puerto del Broker/Bootstrap-server del gestor de colas.
- IP y Puerto del EV\_M.  Es EV\_CP\_M

Al arrancar la aplicación, esta se conectará al EC\_Central para autenticarse y validar que el vehículo está operativo y preparado para prestar servicios. La aplicación permanecerá a la espera hasta recibir una solicitud de servicio procedente de la central. **El conductor quedará asignado con el ID recibido en la línea de parámetros y cuya validez deberá ser confirmada por EC\_Central en el proceso de autenticación.**

### Monitor:

Aplicación que simula un módulo de gestión de observación de todo el CP (hardware y software) velando por la seguridad y correcto funcionamiento de este sin incidencias. El nombre

de la aplicación será **obligatoriamente “EC\_CP\_M”**. Para su ejecución recibirá por la línea de parámetros al menos los siguientes argumentos:

- IP y puerto del EV\_CP\_E
- IP y puerto del EV\_Central
- ID del CP: Identificador único con que ese CP está dado de alta en la red.

Al arrancar la aplicación esta se conectará al Engine del CP al que pertenece y empezará a **enviarle cada segundo un mensaje de comprobación de estado de salud**. Si no se recibe respuesta desde el EV\_CP\_E o se recibe una respuesta tipo KO, EV\_CP\_M enviará a EV\_Central un mensaje de avería como se ha indicado en anteriores apartados. Para simular dichas incidencias, la aplicación EV\_CP\_E deberá permitir que, en tiempo de ejecución, se pulse una tecla para reportar un KO al monitor.

## DRIVERS

Aplicación que disponen los usuarios del sistema para solicitar un suministro. El nombre de la aplicación será **obligatoriamente “EV\_Driver”**. Para su ejecución recibirá por la línea de parámetros al menos los siguientes argumentos:

- IP y puerto del Broker/Bootstrap-server del gestor de colas.
- ID del cliente: Identificador único con que ese cliente está dado de alta en CENTRAL.

Esta aplicación, además de poder solicitar un suministro puntualmente en un CP, podrá leer un fichero todos los servicios que va a solicitar. Enviará el primero a EC\_Central y, tras su conclusión (sea en éxito o fracaso), esperará 4 segundos y pasará a solicitar el siguiente servicio.

## Protocolos de comunicación

### Protocolo OCPP

En el mundo real, el protocolo más habitual de conexión de los puntos de recarga es el denominado OCPP (Open Charge Point Protocol) [openchargealliance.org](https://openchargealliance.org)

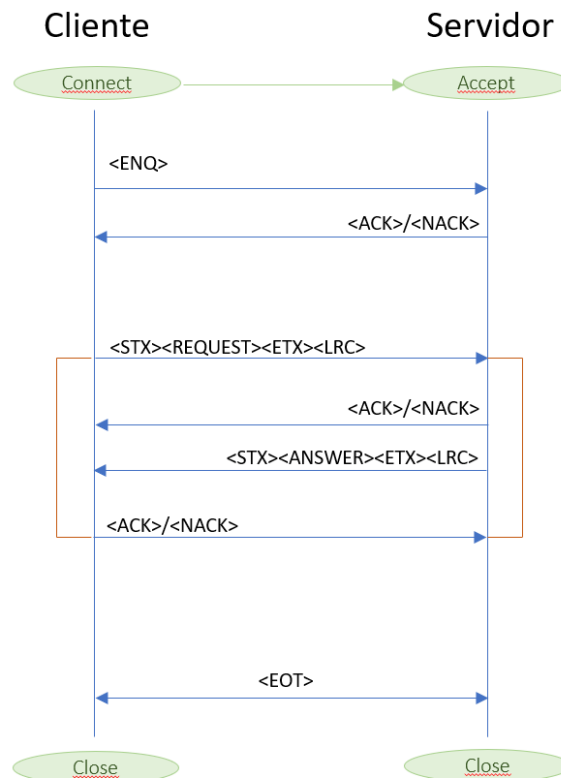
OCPP es un estándar abierto de comunicación entre un punto o estación de recarga (CS) y el sistema de gestión de la red (CSMS equivalente a nuestro EV\_Central). Su objetivo es garantizar la interoperabilidad: que cargadores y plataformas de distintos fabricantes “hablen el mismo idioma”, evitando el “vendor lock-in”, es decir, que los fabricantes de los sistemas tengan cautivos a los usuarios. Hoy es el estándar de facto del sector y, desde 2024/2025, cuenta con estatus de norma IEC 63584 (y su versión 2.1 como IEC 63584-210), lo que refuerza su adopción global.

Como resumen muy de alto nivel de su funcionamiento podemos decir:

- **Arquitectura:** la estación se conecta al CSMS y se intercambian mensajes para alta (boot), autorización, inicio/fin de sesión, telemetrías, eventos, etc.
- **Transporte y formato:** en las versiones modernas se usa **WebSocket + JSON**; en **1.6** existen sabores **JSON (1.6J)** y **SOAP**. El CP inicia la conexión WebSocket y envía/recibe tramas **CALL / CALLRESULT / CALLERROR**.
- Ejemplo de llamada 1.6J:  
`["2","UID-123","BootNotification",{"chargePointVendor":"ACME","chargePointModel":"X1"}]`

## Protocolo estándar sockets

Como se comenta al principio de este documento, la comunicación mediante sockets obliga a los actores involucrados a definir la mensajería y protocolo con precisión. Los estudiantes podrán definir los protocolos de comunicación entre los distintos módulos del sistema a su criterio. **Los profesores recomiendan, aunque no es obligatorio**, usar una mensajería basada en el estándar de empaquetado <STX><DATA><ETX><LRC> que corresponde al siguiente flujo:



Donde:

- <REQUEST> y <ANSWER>: Contienen el mensaje transmitido entre ambos puntos con los campos separados por un carácter determinado:
  - o Ej.: <REQUEST>: Código Operación#campo1#...#campo n
- <LRC>: Se define como el XOR(MESSAGE) byte a byte y sirve para validar que la transmisión del mensaje se ha realizado satisfactoriamente y ha sido recibida de forma correcta por el destinatario el cual responderá al mismo con un <ACK> o <NACK>

## Aclaraciones finales

Antes incongruencias funcionales o aspectos no considerados en los anteriores apartados que impidan la correcta implementación de la práctica, los estudiantes deberán implementar la alternativa que entiendan más conveniente previa consulta con el profesor para consensuar dicha implementación evitando un excesivo nivel de complejidad.

## Guía mínima de despliegue

Para la correcta evaluación de la práctica es necesario comprobar que la aplicación distribuida solicitada es desplegada en un entorno verdaderamente distribuido. Es por ello por lo que para su prueba es necesario al menos 3 PCs distintos en los que se desplegarán los componentes solicitados. Al menos se ha de desplegar el siguiente escenario. **El uso de la tecnología Docker para el despliegue es altamente recomendable y se valorará positivamente**, aunque no estrictamente obligatorio:

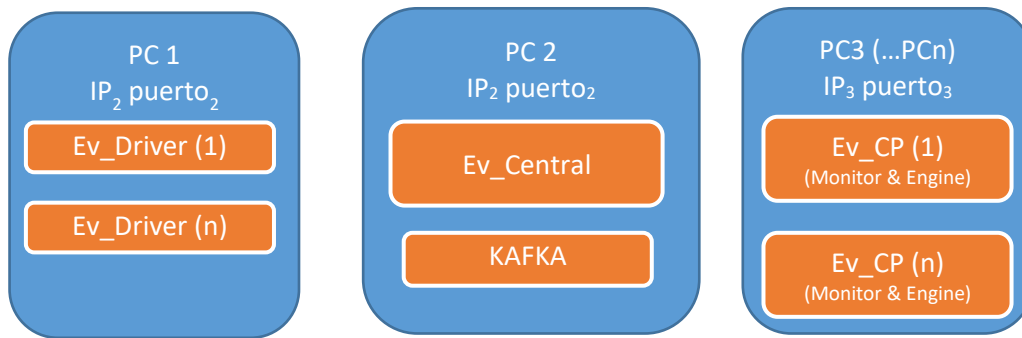


Figura 4. Escenario físico mínimo para el despliegue de la práctica.

## Entregables y evaluación

La evaluación de la práctica se realizará en los laboratorios. **Se podrá realizar en grupos de hasta 2 personas sin perjuicio de que, durante el momento de la corrección, el profesor pueda preguntar a cualquier estudiante del grupo por cualquier aspecto de cualquiera de los módulos.** Los estudiantes deben desplegar por ellos mismos la práctica que resuelve el enunciado anterior. Deben desplegar un sistema completo, es decir, una central, los CPs (engine y monitor) y conductores todos ellos interconectados entre sí. **Este requisito es indispensable para poder realizar la corrección.** Además, deben poderse evaluar positiva o negativamente todos los apartados que aparecerán en la Guía de corrección que se entregará a tal propósito. Cada uno de los apartados puntúa de forma variable, por tanto, cada apartado no implementado o que no pueda comprobarse su correcto funcionamiento no podrá ser tenido en cuenta y por tanto no puntuará. Los estudiantes deberán presentar para la evaluación el documento **“Guía de corrección”** cumplimentado para que el profesor pueda validar los apartados implementados.

Aunque se indicará en la guía de corrección, se adelantan aquí las opciones de despliegue:

- 1- **En máquinas del laboratorio y/o componentes ubicados remotamente accesibles desde el laboratorio .** Opción mejor valorada.
- 2- **En una red con portátiles propios:** En caso de que la opción anterior no funcione se acepta la creación de una red con portátiles pero siempre con el criterio de despliegue distribuido. Sería menos valorado (a criterio del profesor puede restar algún punto) pero no impide su presentación ni buena evaluación.

**3- En un portátil con máquinas virtuales (o dockerizando) en diferentes redes dentro de un solo equipo.**

**4- En un portátil en modo local host:** Será evaluable, pero con una penalización importante (a criterio del profesor).

**En todos los casos se puede y será valorado muy positivamente el uso dockers realizando diferentes contenedores con las distintas aplicaciones separadas para distribuirlas en las máquinas.**

Los estudiantes deberán entregar, además, mediante la funcionalidad de evaluación del UACloud antes de la fecha establecida a su profesor de prácticas una **memoria de prácticas**, con el código fuente y compilados generados, así como un documento donde se detalle la siguiente información. El formato es libre, pero debe ser un documento ordenado y debidamente formateado, cuidando la redacción y ortografía.

- Portada con los nombres, apellidos y DNI de los estudiantes, año académico y el título de la práctica.
- Un informe donde se indique el nombre de los componentes software desarrollados y una descripción de cada uno de ellos, explicando y enviando además el código fuente de todos ellos.
- El detalle, paso a paso, de una guía de despliegue de la aplicación, que deberá ser la misma que utilice cuando haga la corrección de la práctica.
- Capturas de pantalla que muestren el funcionamiento de las distintas aplicaciones conectadas.

Cada profesor de prácticas podrá solicitar a los estudiantes cualquier otra evidencia que el profesor considere adecuada para poder formalizar la evaluación.

**La fecha de entrega se realizará en la semana del 27/10/2025.**