SD	Sistemas Distribuidos
25/26	Práctica no guiada: Seguridad y API Rest
	EVCharging Network

Preámbulo

El objetivo de esta práctica es que los estudiantes implementen, por un lado, dentro de los principios de arquitectura SOA (Service Oriented Architecture), la tecnología de comunicación de basada en servicios REST y por otro algunos de los principios de seguridad vistos en teoría.

Para ello, se consumirán una serie API Rest ya establecidos por un tercero y, por otro, se creará y expondrá otro API Rest desde un back que sea consumido por un front.

Adicionalmente, se implementarán tres aspectos relacionados con la seguridad: cifrado de canal, autenticación segura y principios de auditoría.

Se partirá de la misma práctica ya generada en la primera parte de la asignatura con la misma funcionalidad expuesta.

Especificación

Descripción funcional

La funcionalidad que deberá implementarse será idéntica a la implementada en la release 1 con las siguientes modificaciones:

1- EV_Central deberá comunicarse con un módulo externo EV_W (Weather Control Office) que proporcionarán la viabilidad o no de usar un determinado CP en función a la climatología.

2- Se añadirá un nuevo módulo llamado EV_W (Weather Control Office) el cual determinará el estado del tiempo en las localizaciones donde se encuentran los puntos de recarga. Para ello se comunicará vía API con un proveedor externo (Openweather) como se detalla en los siguientes apartados.

3- Se añadirá un nuevo módulo llamado EV_Registry que permitirá registrar un nuevo CP o darlo de baja. Este módulo se comunicará con los CPs vía API_Rest.

4- Se creará un Front (mediante una simple página web) el cual mostrará el cuadro de control de Central a cualquiera que la invoque. Es, por tanto, una web pública ejecutable en cualquier navegador.

5- Se implementarán los siguientes mecanismos de seguridad:
   - Autenticación segura entre los CPs y el Registry: cifrado del canal y protección segura de credenciales.
   - Auditoría de eventos en la Central.
   - Cifrado de los datos entre la Central y los CPs.

Diseño técnico

Partiendo del diseño técnico ya implementado en la primera parte de la práctica, se modificará y ampliará el mismo para contemplar la siguiente arquitectura:

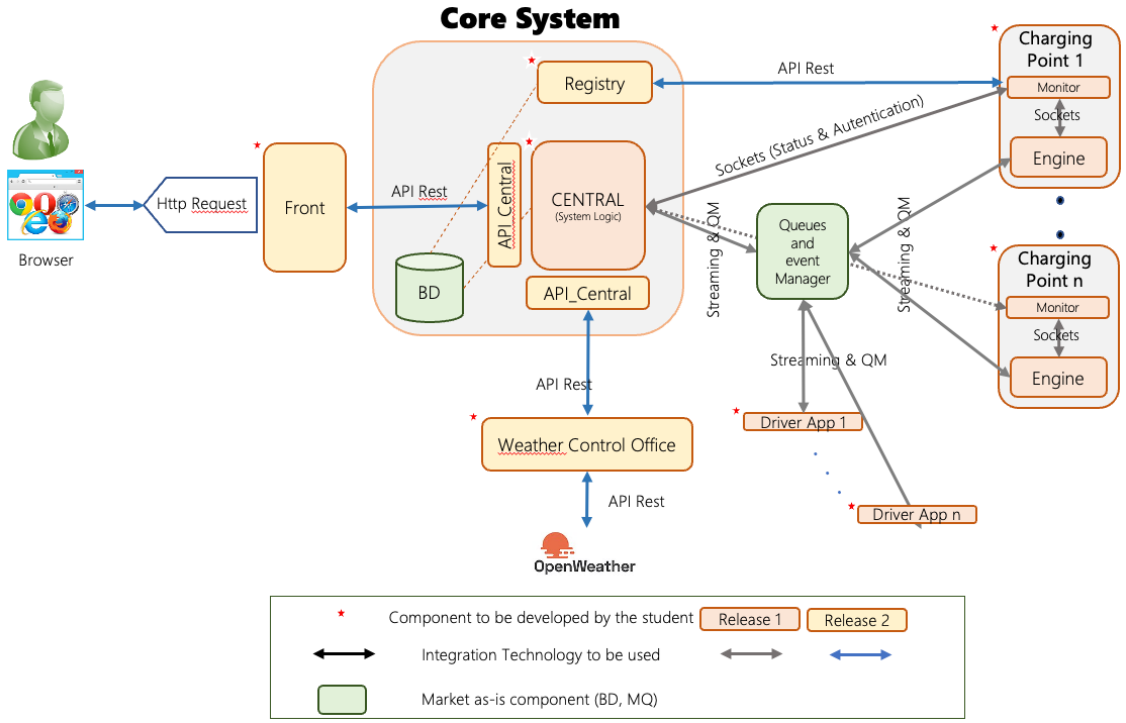


Figura 1. Esquema conceptual del Sistema software, interconexiones entre los componentes y tipos de interfaces.

Al igual que en la primera parte de la práctica, 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.

Registry (EV_Registry):

EV_Registry será el módulo mediante el cual los CPs desde EV_M pueden darse de alta o baja en el sistema. Dispondrá de las opciones de alta, baja y autenticación de un CP.

Los CPs, **previo a su autenticación y posterior prestación de servicios**, deberán realizar una petición de alta para registrar **su ID y localización** en el sistema o darlo de baja. Sin este proceso los CPs no podrán autenticarse ni prestar servicios. Ante un intento de autenticación sin previo registro recibirán una denegación de EV_Central.

El mecanismo de registro de los CPs se realizará mediante un API_Rest que implementará los métodos oportunos (GET, PUT, DELETE, …).

La comunicación entre EV_Registry y EV_CP_M deberá realizarse mediante el establecimiento de un canal seguro (HTTPS, SSL, RSA, …).

Opcionalmente la identificación de los CPs podrá realizarse en base a un certificado (autofirmado) que contenga los datos identificativos de este.

En respuesta al registro del CP, EV_Registry devolverá las claves (credenciales) que el CP deberá usar durante su autenticación en EV_Central.

Core System

CENTRAL:

Será idéntico al ya realizado en la primera parte con las siguientes modificaciones:

1- **Consumo de API_rest de una Oficina de Control de Clima:** EV_Central expondrá un interface (API) para que un nuevo módulo llamado EV_W (Weather Control Office) le informe si la localización en la que se encuentra un CP tiene una climatología adecuada para su uso. Para ello, EV_W ante una situación de alerta climatológica en una determinada localización, **notificará dicha alerta vía API a EV_Central .** Más adelante se indica el funcionamiento de EV_W.

2- **Implementación de la autenticación entre EV_Central y los CP**: Como se ha comentado anteriormente, para poder realizar la autenticación, los CPs previamente se deberán de haber registrado en el EV_Registry. Tras el proceso de registro los CPs estarán disponibles para su uso. Para ello deberán autenticarse en EV_Central con las credenciales que EV_Registry habrá proporcionado a tal propósito. **En el momento de la autenticación, si esta se resuelve con éxito, la central devolverá a EV_M, la clave (de cifrado simétrico ÚNICA POR CP) que deberá ser usada por este para el cifrado de todos los mensajes que envíe a la Central**. La central descifrará los mensajes teniendo en cuenta dicha clave. Estas claves podrán ser revocadas por EV_Central ante una supuesta vulnerabilidad de la seguridad. Para simular este efecto, en EV_Central, se implementará una opción para restaurar claves. Al pulsarla, se borrarán las claves de un CP específico el cual quedará fuera de servicio. Esto obligará a **EV_CP_M a realizar una nueva autenticación mediante una opción incorporada en el mismo** y, de esta manera, obtener sus nuevas claves de cifrado. *Opcional: Tal y como se refleja en el diagrama de arquitectura conceptual, la autenticación seguirá siendo por sockets pero el alumno que lo desee podrá implementar un API Rest entre CP y Central a tal propósito lo que, en la práctica profesional sería más correcto. En este caso Central expondría un API de Autenticación que el CP consumiría.*

3- **Implementación de cifrado en el canal entre EV_Central y los CPs (Engine)**: Como se ha indicado en el punto anterior, se establecerá un sistema de cifrado del canal que los CPs depositan en los Topics de las colas para evitar ataques del tipo MITM (Man In The Middle). Esto se realizará mediante un sencillo sistema de cifrado simétrico elegible por el estudiante con su justificación adecuada. Todos los mensajes entre los CP y Central estrán cifrados. *Nota: Kafka dispone de 3 componentes que permiten implementar mecanismos de seguridad a este propósito. Dichos componentes son: Cifrado de datos SSL/TLS, Autenticación SSL o SASL y Autorización mediante ACL. No se exige al estudiante la incorporación en la práctica*

de estos componentes si bien aquellos que lo deseen pueden hacerlo siendo valorado positivamente.

4- Implementará un **registro de auditoría de todos los eventos que sucedan**, indicando, de forma estructurada, los datos de dicho evento como se indica a continuación:
   a. Fecha y hora del evento.
   b. Quién y desde dónde se produce el evento: IP de la máquina que genera el evento (ej. IP del CP, IP del Usuario)
   c. Qué acción se realiza: Autenticación o intentos de autenticación fallidos o no, Incidencias durante el servicio, cambio de la situación del CP, usuarios o CPs bloqueados, errores, etc.
   d. Parámetros o descripción del evento.

Nota informativa: Los sistemas profesionales deben incorporar estos conceptos de auditoría mediante herramientas específicas que integran un SIEM (Security Information and Event Management).

API_Central:

Este componente expondrá un API_Rest con los métodos oportunos (GET, PUT, DELETE,…) que permitirá desde cualquier componente externo consultar el estado de los Drivers, CPs y las transacciones en curso como se expresaba en la primera parte de la práctica. **Este API permitirá igualmente ser consumido por el módulo EV_W para notificar a la central de una situación de alerta o la cancelación de dicha alerta de una localización en la que se encuentre un determinado CP**. Al igual que en la anterior versión de la práctica, el interfaz front que consuma este API deberá contener todos los elementos necesarios para visualizar claramente el estado de situación de todos los elementos del sistema (CPs, Drivers, mensajes de error, etc) añadiendo en esta ocasión el estado del clima (temperatura) de cada localización.

Base de Datos:

Podrá ser accesible tanto EV_Central como EV_Registry para compartir y actualizar la información de los CPs.

Monitor (EV_M)

Adicionalmente a las funcionalidades ya disponibles, este módulo implementará una nueva opción (disponible en un menú) para poder conectarse al EV_Registry y registrarse en el sistema recibiendo su clave de autenticación. Este proceso se implementará mediante el consumo de un API.

La conexión y autenticación deberá realizarse de forma segura evitando la exposición en claro de los datos de identificación del CP.

A su vez, como se ha descrito en apartados anteriores, durante el proceso de autenticación recibirá la clave para el cifrado simétrico que el EV_CP_E deberá implementar en todos los mensajes con la central.

Weather Control Office (EV_W)

Como se ha indicado en el apartado de descripción funcional de este documento, se añadirá un nuevo módulo al sistema llamado EV_W (Weather Control Office) el cual determinará la viabilidad de operación de un CP en función a la temperatura de la localización donde se encuentra dicho CP.

Para ello, el módulo EV_W dispondrá, a efectos de simulación, mediante una opción de menú o un archivo, la posibilidad de indicar las ciudades donde se encuentran los CP. *Nota: Aunque el alta o baja de un CP en EV_Registry debería actualizar las localizaciones en EV_W notificándoselo automáticamente vía API, por simplicidad, no se exije esta implementación y será suficiente con introducir manualmente en el módulo EV_W esa nueva localización.*

A su vez, EV_W, **cada 4 segundos**, se conectará al servidor de clima OPENWEATHER para preguntarle el tiempo que hace en la ciudad.

Si la temperatura de alguna de las localizaciones está por debajo de los 0º C, EV_W , a través del API_Central, notificará la alerta a EV_Central para que esta, a su vez, envíe un mensaje de parada al CP correspondiente. Si, en ese momento, se está realizando un suministro, el mismo finalizará con normalidad y posteriormente el CP pasará a modo "fuera de servicio"-

Cuando la temperatura vuelva a superar los 0ºC, EV_W lo notificará igualmente a EV_Central para que la operación en el CP se restaure adecuadamente.

Nota: A efectos de la corrección, **cualquier localización podrá ser cambiada a voluntad del profesor en cualquier momento** mediante la opción de menú mencionada, sin necesidad de reiniciar ninguna parte del sistema.

Front

Este módulo consistirá en una simple página web que, haciendo peticiones al API_Central, muestre el cuadro de monitorización de la central con todos sus elementos.

Como se ha comentado anteriormente, este interfaz front deberá contener todos los elementos necesarios para visualizar claramente el estado de situación de todos los elementos del sistema: CPs (incluyendo su estado de registro, activación y token), Drivers, mensajes de error de cualquier parte del sistema, mensajes de estado del EV_W, clima, alertas ante cualquier fallo de cualquier módulo del sistema, etc.

Nota: Aunque al igual que en el resto de componentes el estudiante podrá elegir la tecnología de su preferencia se recomienda, por su sencillez, el uso de Node js.

Drivers

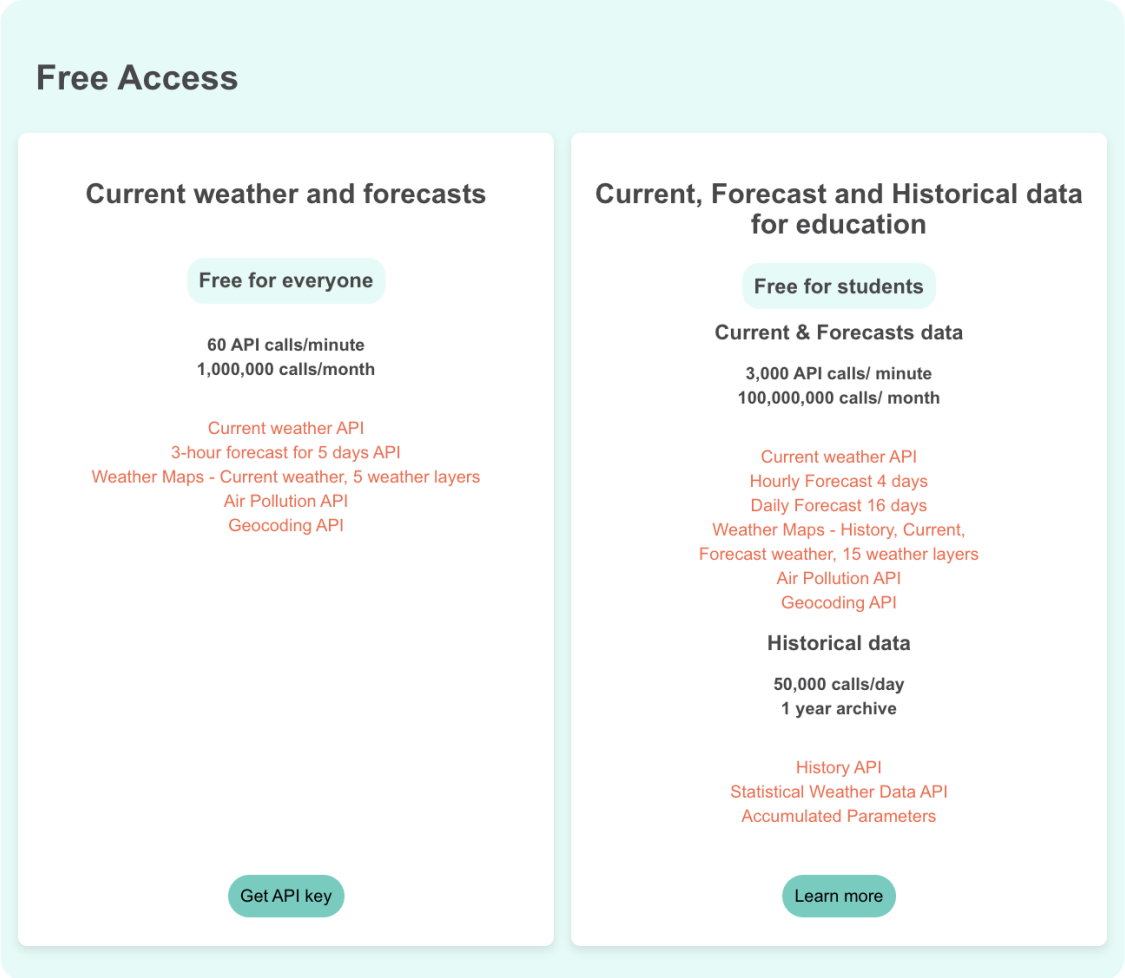
No cambian su funcionalidad respecto de la entrega anterior. No es necesario que implemente mecanismos de cifrado con la central.

Servidor de clima (OPEN WEATHER)

Mediante la funcionalidad aportada por la plataforma OPEN WEATHER (https://openweathermap.org/) es posible obtener múltiples datos relacionados con el clima de cualquier parte del mundo.

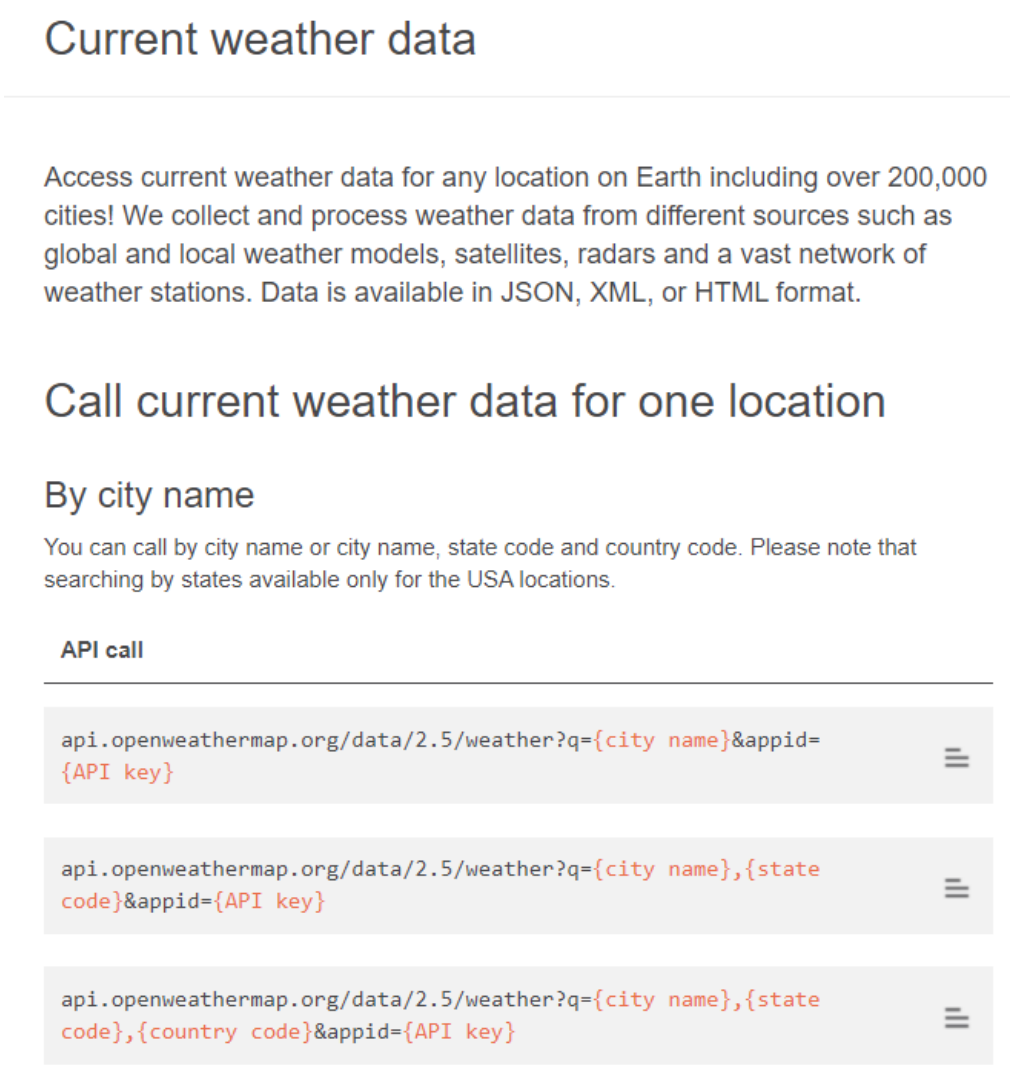
A través de su API el EV_W podrá acceder a dichos datos.

Para ello bastará con que el estudiante se registre en la versión FREE, solicite el API key (Get API Key) y realizar una petición para cada ciudad que quiera consultar.



Modelos de suscripción de Open Weather

De todas las posibilidades que ofrece el API es suficiente con que se acceda a las funcionalidades que se encuentran en los métodos de “Current weather data” como se aprecia en la imagen siguiente. Toda la información al respecto se encuentra disponible en la página web del proveedor (https://openweathermap.org/current):



API Current Weather Data

El JSON de respuesta de las peticiones que se muestran en la anterior imagen es el siguiente del cual solo necesitaremos la temperatura.

JSON

Example of API response


```
{
  "coord": {
    "lon": -122.08,
    "lat": 37.39
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 282.55,
    "feels_like": 281.86,
    "temp_min": 280.37,
    "temp_max": 284.26,
    "pressure": 1023,
    "humidity": 100
  },
  "visibility": 16093,
  "wind": {
    "speed": 1.5,
    "deg": 350
  },
  "clouds": {
    "all": 1
  },
  "dt": 1560350645,
  "sys": {
    "type": 1,
    "id": 5122,
    "message": 0.0139,
    "country": "US",
    "sunrise": 1560343627,
    "sunset": 1560396563
  },
  "timezone": -25200,
  "id": 420006353,
  "name": "Mountain View",
  "cod": 200
}
```

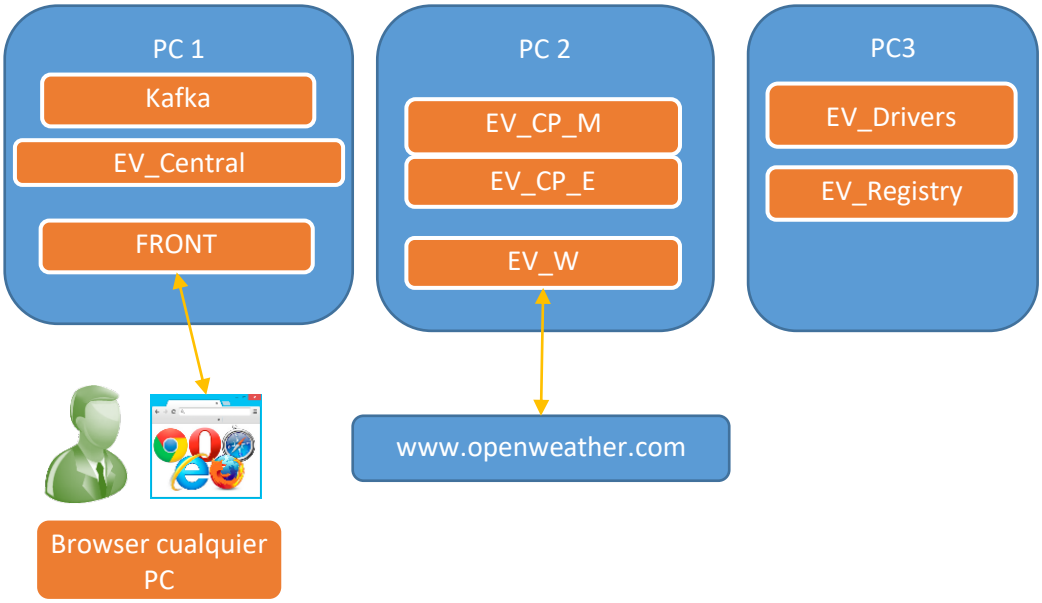

Temperatura en ºK

Ciudad

JSON de respuesta del request al API Current Weather Data

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 que para su prueba es necesario al menos 3 PCs distintos en los que se desplegarán los componentes solicitados proporcionando el siguiente escenario:



Escenario físico 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 con todos los módulos 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 puntuara. Los estudiantes deberán presentar para la evaluación el documento **"Guía de corrección"** cumplimentado para que el profesor pueda validar los apartados implementados.

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 será en la semana del 16/12/2025.