SD	Sistemas Distribuidos
21/22	Práctica no guiada: Seguridad y API Rest
	Gestor de colas en parques temáticos “Fun with Queues”

Preámbulo

El objetivo de esta práctica es que los alumnos 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 establecido 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 con las siguientes modificaciones:

1- El FWQ_Engine deberá comunicarse con un sistema real que nos proporcionará , entre muchas de sus variables, la temperatura que hace en cualquier ciudad del mundo que le preguntemos.

2- Se creará un Front (mediante una simple página web) la cual mostrará el estado del parque a cualquiera que la invoque. Es, por tanto, una web pública.

3- Se modificará FWQ_Registry para que un visitante pueda conectarse al registro vía API_Rest en vez de Sockets. Habrá por tanto visitantes que se conecten por sockets y otros por API_Rest.

4- Se implementarán los siguientes mecanismos de seguridad:

- Autenticación segura entre los vistantes y el Registry: cifrado del canal y protección segura de las contraseñas.
- Auditoría de eventos en el Registry.
- Cifrado de los datos entre Engine y los visitantes.

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:

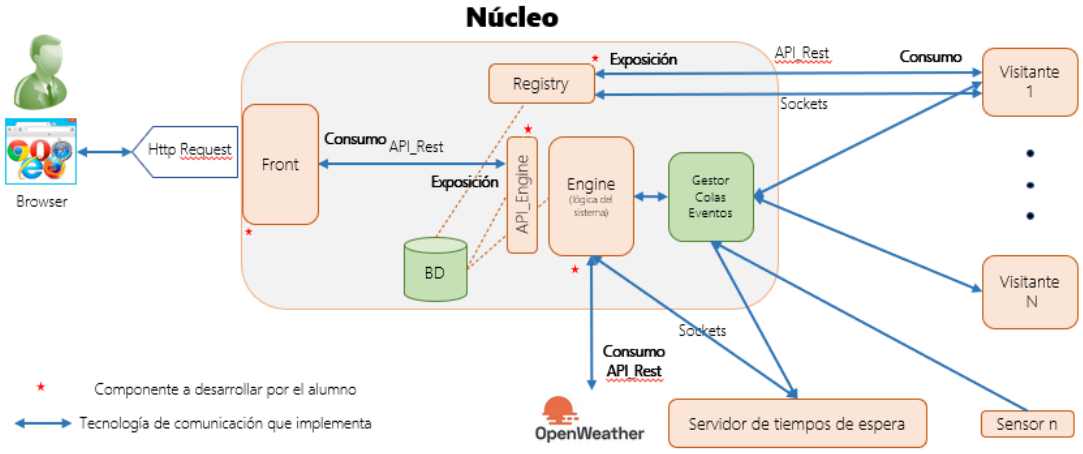


Figura 1. Esquema conceptual del Sistema software, interconexiones entre los componentes y tipos de interfaces.

Al igual que en la primera parte de la práctica, los componentes pueden ser desarrollados en el lenguaje de preferencia del alumno: Java, C/C++, . NET, Python, etc. asumiendo el alumno la responsabilidad de su conocimiento y forma de desplegarlo en el laboratorio.

Núcleo

Engine:

Será idéntico al ya realizado en la primera parte, pero tendrá las siguientes modificaciones:

1- **Consumo de API_rest de un servidor de clima:** Adicionalmente a su conexión con el FWQ_WaitingTimeServer, consumirá el API que ofrece el proveedor de OPEN WEATHER para obtener el tiempo de 4 ciudades distintas. El alumno deberá elegir de una lista que creará por si mismo 4 ciudades al azar y, por cada una de ellas, hacer la llamada al API de OPEN_WEATHER. *Nota: En el momento de la corrección, el profesor podrá solicitar al alumno que añada cualquier ciudad a su criterio.* Se asume que las 4 ciudades consumidas corresponden a las cuatro regiones de 10x10 identificadas en el mapa del resort. Si la temperatura en cualquiera de ellas está por debajo de 20ºC o por encima de 30º, esa zona del parque quedará bloqueada y sus atracciones cerradas obligando a redirigir a los visitantes hacia otra atracción.

2- Deberá ser capaz de guardar en la BD cada cierto tiempo el mapa del parque en curso. El tiempo de guardado será a decidir por el alumno, pero deberá ser razonablemente pequeño (1 segundo) para que, como se indica en los siguientes apartados, se pueda seguir en un componente externo la transmisión de la situación del parque.

3- **Implementación de seguridad con Visitantes**: Se establecerá un sistema de cifrado del canal que los Visitantes depositan en los Topics de las colas para evitar ataques del tipo MITM (Man In The Middle). Se realizará un sencillo sistema de cifrado con intercambio de claves y certificados (simétrico o asimétrico). *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 alumno la incorporación en la práctica de estos componentes si bien aquellos que lo deseen pueden hacerlo siendo valorado positivamente.*

API_Engine:

Este componente expondrá un API_Rest con los métodos oportunos (GET, PUT, DELETE,…) que permitirá desde cualquier componente externo consultar el estado de los visitantes y el mapa en curso.

Registry:

Implementará un API_Rest con los métodos oportunos (GET, PUT, DELETE, …) para el registro de los visitantes.

Se deberá proteger adecuadamente tanto los datos tratados como la conexión. Así, las contraseñas de los visitantes se protegerán mediante un algoritmo de cifrado irreversible.

Para la protección del canal se podrá usar un sistema de cifrado asimétrico (SSL, RSA, …).

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 visitante),
c. Qué acción se realiza: Alta, Modificación, Baja, Error
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).

Base de Datos:

No contendrá modificaciones respecto de la práctica anterior siempre que el alumno esté almacenando en ella los datos necesarios para la realización de esta práctica incluido el mapa del parque y el estado de los visitantes.

A la misma podrán acceder en esta ocasión tanto FWQ_Engine como API_Engine.

Visitante

Se desarrollará un nuevo tipo de visitante que se conectará al Registry mediante el consumo de un API que este implementará a tal propósito.

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

Front

Este módulo consistirá en una simple página web que, haciendo peticiones al API_Engine muestre el mapa en curso y el estado de los visitantes.

Deberá controlar cualquier aspecto que evite un funcionamiento incorrecto. Así, por ejemplo, si cualquier módulo está caído, deberá mostrar un mensaje al respecto.

Nota: Aunque al igual que en el resto de componente el alumno puede elegir la tecnología de su preferencia se recomienda, por su sencillez, el uso de Node js con arreglo a las indicaciones que se han ofrecido en la práctica guiada entregada.

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 FWQ_Engine podrá acceder a dichos datos.

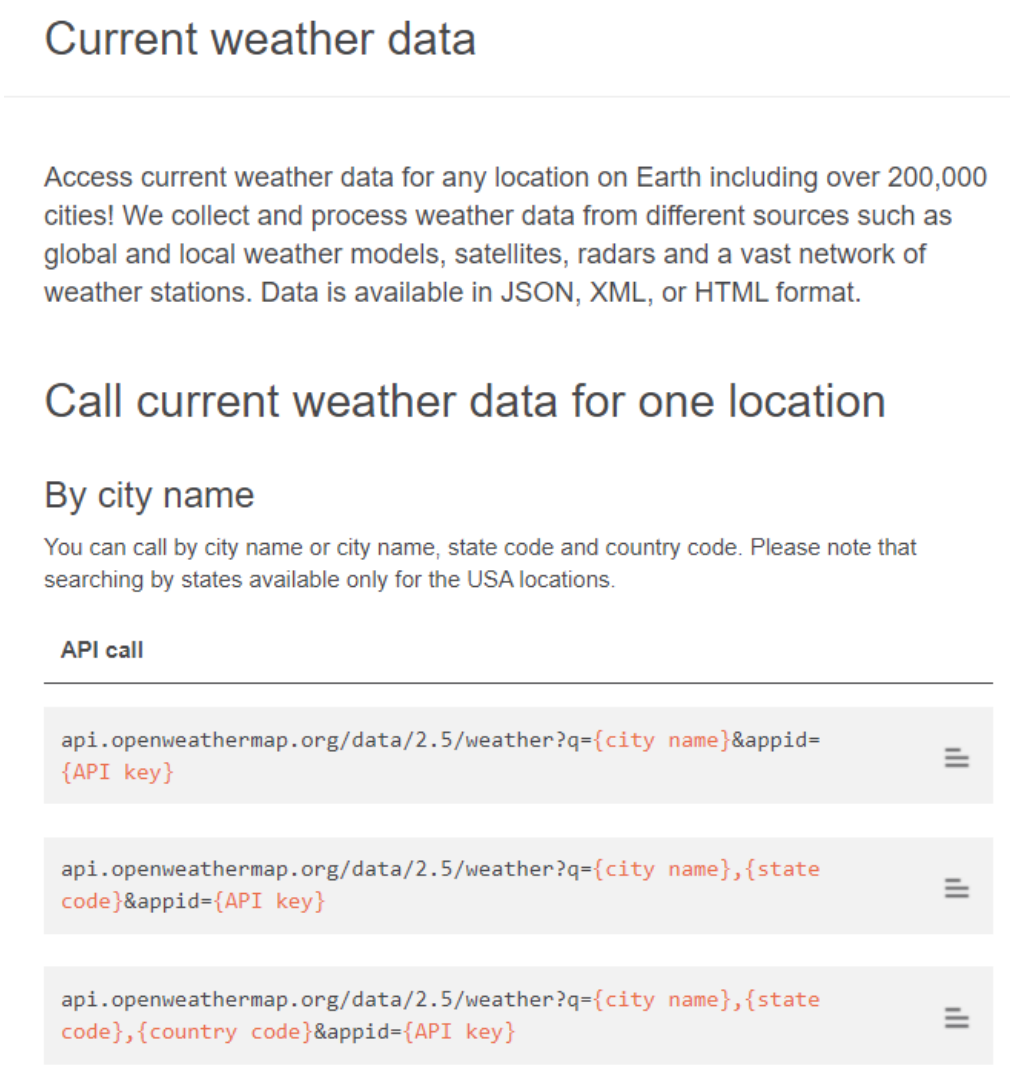
Para ello bastará con que el alumno se registre en la versión FREE, solicite el API key (Get API Key) y realizar una petición para cada ciudad que quiera consultar.

Current weather and forecasts collection

Free	Startup	Developer	Professional	Enterprise
Get API key	40 USD / month Subscribe	180 USD / month Subscribe	470 USD / month Subscribe	2,000 USD / month Subscribe
60 calls/minute 1,000,000 calls/month	600 calls/minute 10,000,000 calls/month	3,000 calls/minute 100,000,000 calls/month	30,000 calls/minute 1,000,000,000 calls/month	200,000 calls/minute 5,000,000,000 calls/month
Current Weather	Current Weather	Current Weather	Current Weather	Current Weather
Minute Forecast 1 hour*	Minute Forecast 1 hour**	Minute Forecast 1 hour	Minute Forecast 1 hour	Minute forecast 1 hour
Hourly Forecast 2 days*	Hourly Forecast 2 days**	Hourly Forecast 4 days	Hourly Forecast 4 days	Hourly Forecast 4 days
Daily Forecast 7 days*	Daily Forecast 16 days	Daily Forecast 16 days	Daily Forecast 16 days	Daily Forecast 16 days
National Weather Alerts*	National Weather Alerts**	National Weather Alerts	National Weather Alerts	National Weather Alerts
Historical weather 5 days*	Historical weather 5 days**	Historical weather 5 days	Historical weather 5 days	Historical weather 5 days
Climatic Forecast 30 days	Climatic Forecast 30 days	Climatic Forecast 30 days	Climatic Forecast 30 days	Climatic Forecast 30 days
Bulk Download	Bulk Download	Bulk Download	Bulk Download	Bulk Download
Basic weather maps	Basic weather maps	Advanced weather maps	Advanced weather maps	Advanced weather maps
Historical maps	Historical maps	Historical maps	Historical maps	Historical maps
Global Precipitation Map	Global Precipitation Map	Global Precipitation Map	Global Precipitation Map	Global Precipitation Map
Road Risk API	Road Risk API	Road Risk API	Road Risk API	Road Risk API
Air Pollution API	Air Pollution API	Air Pollution API	Air Pollution API	Air Pollution API
Geocoding API	Geocoding API	Geocoding API	Geocoding API	Geocoding API
Weather widgets	Weather widgets	Weather widgets	Weather widgets	Weather widgets
Uptime 95%	Uptime 95%	Uptime 99.5%	Uptime 99.5%	Uptime 99.9%

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
}
```

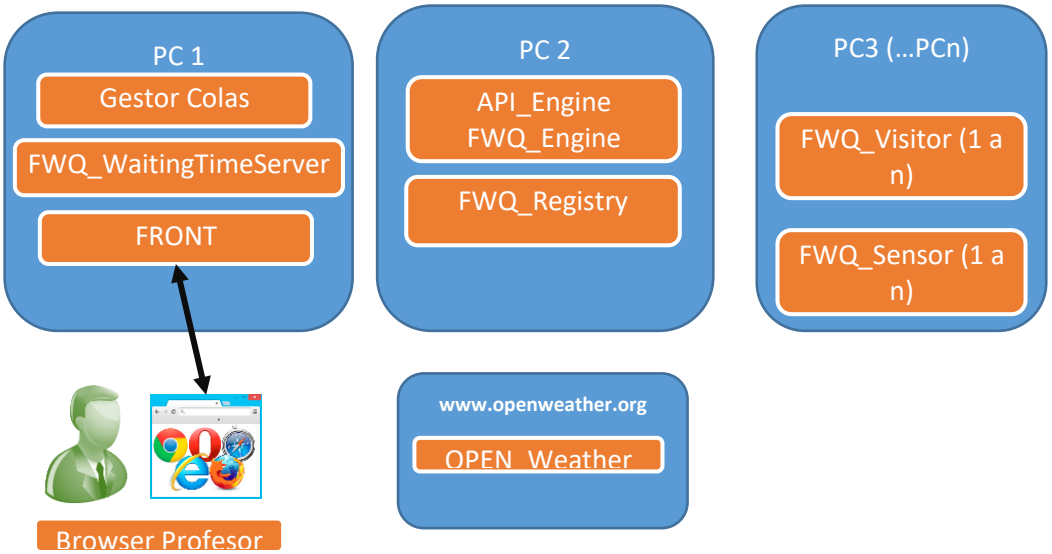

Temperatura en ºK

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 alumno del grupo por cualquier aspecto de cualquiera de los módulos**. Los alumnos 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 alumnos deberán presentar para la evaluación el documento **"Guía de corrección"** cumplimentado para que el profesor pueda validar los apartados implementados.

Los alumnos 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 alumnos, 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 alumnos cualquier otra evidencia que el profesor considere adecuada para poder formalizar la evaluación.

La fecha de entrega será en la semana del 20/12/2021.