

## **PRACTICA Arquitecturas RESTful**

### **OBJETIVO:**

Implementar un ejemplo real de una arquitectura RESTful que permita acceder a los datos almacenados en una base de datos.

### **PRACTICA RESTfull 1.1:**

Construir un servidor REST con un API que permita consultar una base de datos que contenga el fichero bicimad.csv. La base de datos puede ser la propuesta en clase o cualquier otra base de datos que disponga de un API de consulta para Python.

El API debe permitir las siguientes acciones GET:

1. Consultar la cantidad de fechas disponibles en la base de datos
2. Consultar los códigos de las “base\_stations” de origen
3. Consultar los códigos de las “base\_stations” de destino
4. Recuperar todos los movimientos para un determinado día
5. Recuperar todos los movimientos para un determinado día que hagan un trayecto iniciado en una determinada base de origen
6. Recuperar todos los movimientos para un determinado día que hagan un trayecto finalizado en una determinada base de destino
7. Recuperar todos los movimientos para un determinado día que hagan un trayecto iniciado en una determinada base de origen y finalizado en una determinada base de destino
8. Recuperar todos los movimientos para un determinado día que hagan un trayecto iniciado en una determinada base de origen y finalizado en una determinada base de destino y con una duración de trayecto inferior o superior a un determinado valor

El API debe permitir las siguientes acciones POST:

9. Introducir una nueva línea en la base de datos con la información de fecha, ageRange, user\_type, idunplug\_station, idplug\_station, idunplug\_base, idplug\_base y travel\_time. El campo Fichero debe contener el valor “000000” para todas las entradas introducidas por este mecanismo

El uso del servicio debe estar protegido para que sólo puedan realizar acciones de tipo POST aquellos individuos que posean un usuario y una contraseña de administración. Las consultas de tipo GET serán públicas.

### **ENTREGA:**

Se deberá entregar un documento donde se recojan y analicen los siguientes puntos:

- a) Describir brevemente las partes del código.

- b) Resultados obtenidos de la ejecución y documentación de las pruebas realizadas; por ejemplo, verificar que los elementos introducidos con POST pueden ser consultados con GET, o verificar que, sin las credenciales correctas, el servicio POST no está disponible.
- c) Código de la práctica planteada.

### **PRACTICA RESTfull 1.2:**

Mejorar el servicio anterior con alguno de los siguientes puntos:

1. Implementar una función de tipo PUT que permita actualizar el campo del tiempo de trayecto (travel\_time) para una determinada entrada con valores fecha, ageRange, user\_type, idunplug\_station, idplug\_station, idunplug\_base, idplug\_base.
2. Implementar una función de tipo DELETE que permita eliminar campos de la base de datos pasando una combinación de cualquiera de los parámetros fecha, ageRange, user\_type, idunplug\_station, idplug\_station, idunplug\_base, idplug\_base, fichero. Es decir, si se pasa sólo la fecha, se eliminarán todas las entradas con esa fecha, si se pasan el idunplug\_station y el idplug\_station se eliminarán todos los trayectos entre esas estaciones de origen y destino y si se pasan valores para todas las sólo se eliminará la única entrada que coincida con esa combinación.
3. Implementar un sistema por el cual usar el API sin autenticación para realizar llamadas de tipo GET sólo devuelva 10 elementos de cualquier consulta mientras que si se hace la llamada GET con el usuario de administración devuelve todos los elementos que coincidan con esa consulta.

### **ENTREGA:**

Se deberá entregar un documento donde se recojan y analicen los siguientes puntos:

- a) Describir el código generado
- b) Resultados obtenidos de la ejecución y documentación de las pruebas realizadas
- c) Código de la práctica planteada