

PRUEBA TÉCNICA PYTHON

En el fichero *app.py* se presenta el código fuente para el proyecto en cuestión. Por el enunciado del problema, no estaba seguro en si pedíais un endpoint para requerir la información de nombre y código postal mediante un comando GET a una fuente externa como una base de datos o, por el contrario, tenía que recibir la información para registrar un nuevo usuario en una base datos. Finalmente me he decantado por la segunda opción: se trata de un endpoint de tipo POST que recibirá peticiones de ingreso de supuestos nuevos usuarios mediante su nombre y su código postal.

Utilizando el código postal recibido, se realiza una petición GET a la API de geonames para obtener la ciudad correspondiente a dicho código postal. Normalmente, la API de geonames nos devolverá un listado con varias ciudades del mundo que pueden corresponderse al código postal. Para el problema en cuestión he decidido quedarme con la primera de todas.

Para simular la entrada de nuevos usuarios, así como para comprobar el correcto funcionamiento de las comunicaciones, se ha utilizado el software Insomnia, que nos permite enviar los comandos GET y POST en formato JSON. Posteriormente, almacenamos dicha información en una base de datos.

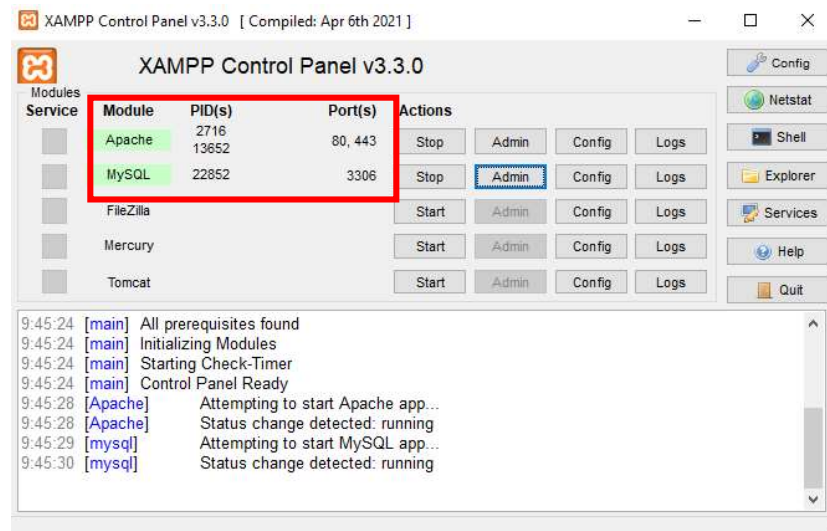
Para la base de datos hemos utilizado el software XAMPP, que es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL e incluye el módulo phpMyAdmin. La base de datos estará formada por 2 tablas:

- Users_master: almacena el nombre del usuario
- Users_details: almacena el código postal y la ciudad

Finalmente, el servidor responde en formato JSON para indicar si el usuario se ha podido añadir satisfactoriamente o si, por el contrario, no ha encontrado el código postal introducido.

A continuación muestro un ejemplo del funcionamiento básico del proyecto:

- 1) Abrimos el XAMPP y arrancamos los módulos de Apache y MySQL, también pulsamos el botón Admin del módulo MySQL para que nos abra la interfaz phpMyAdmin:



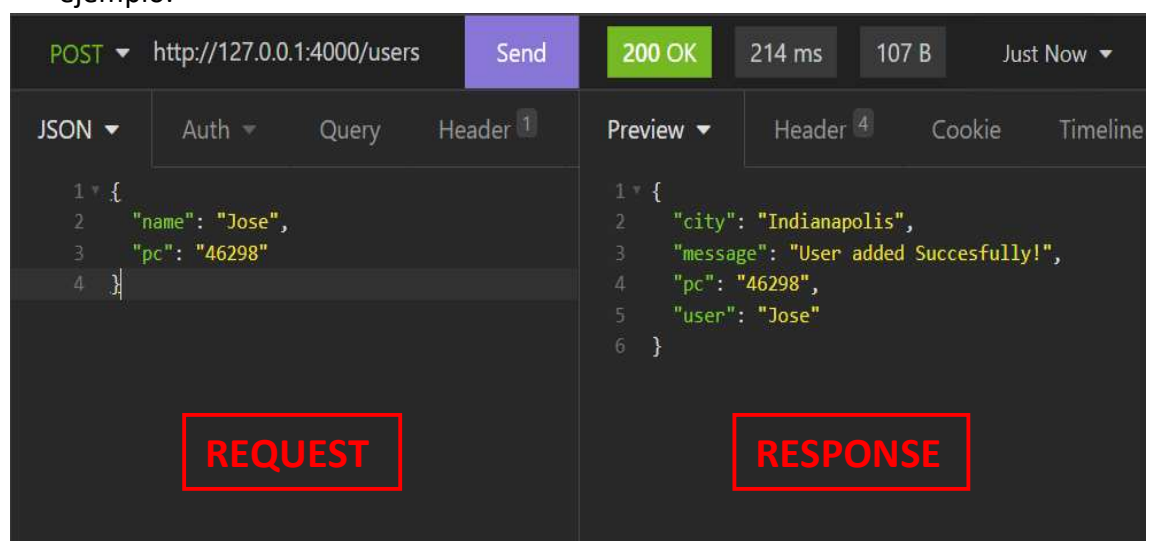
2) Ejecutamos el script *app.py* arrancando de esta manera el servidor:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

(Background on this error at: http://sqlalche.me/e/14/e3q8)
PS C:\Users\HectorJL\Documents\UV_IA3\API_DB\venv\src> python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with windowsapi reloader
* Debugger is active!
* Debugger PIN: 272-482-361
* Running on http://127.0.0.1:4000/ (Press CTRL+C to quit)
  
```

3) Abrimos el programa Insomnia y introducimos los nuevos usuarios, por ejemplo:



- 4) Abrimos phpMyAdmin y comprobamos que el usuario se ha registrado correctamente en la base de datos:

The screenshot shows the phpMyAdmin interface. On the left, the database 'db_users' is selected, and the table 'user_master' is highlighted. The main panel displays the table structure and data. The table has two columns: 'id' and 'name'. A single row is visible with 'id' 1 and 'name' Jose. The interface includes a top navigation bar with options like 'Examinar', 'Estructura', 'SQL', 'Buscar', 'Insertar', 'Exportar', 'Importar', and 'Privilegios'. Below the table, there are options to 'Mostrar todo', 'Número de filas', and 'Filtrar filas'. The status bar at the bottom indicates 'Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0007 segundos.)'.

id	name
1	Jose

1. Tabla user_máster

The screenshot shows the phpMyAdmin interface. On the left, the database 'db_users' is selected, and the table 'user_details' is highlighted. The main panel displays the table structure and data. The table has three columns: 'id', 'pc', and 'city'. A single row is visible with 'id' 1, 'pc' 46298, and 'city' Indianapolis. The interface includes a top navigation bar with options like 'Examinar', 'Estructura', 'SQL', 'Buscar', 'Insertar', 'Exportar', 'Importar', and 'Privilegios'. Below the table, there are options to 'Mostrar todo', 'Número de filas', and 'Filtrar filas'. The status bar at the bottom indicates 'Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0007 segundos.)'.

id	pc	city
1	46298	Indianapolis

2. Tabla user_details