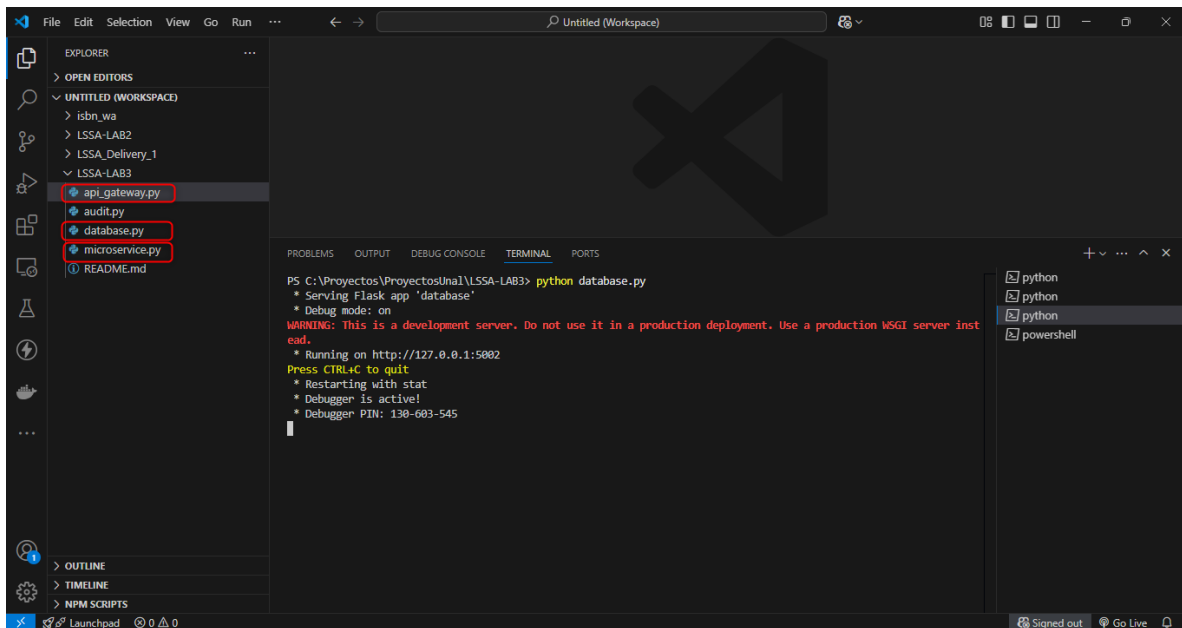


## Step 1: Create the API Gateway

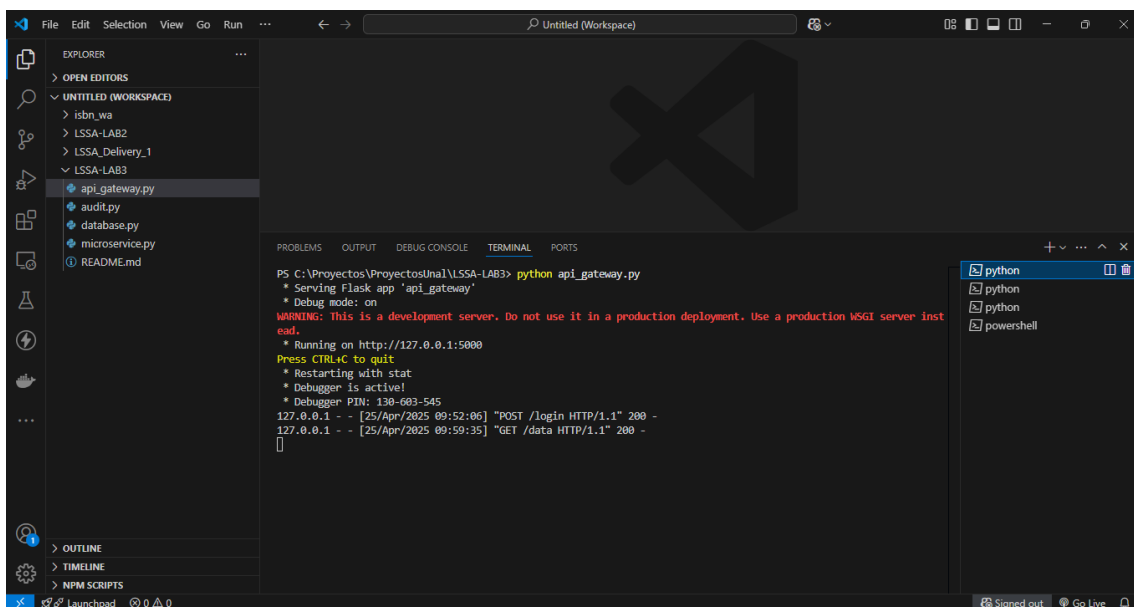
## Step 2: Create the Microservice (MS)

## Step 3: Create the Database (DB)

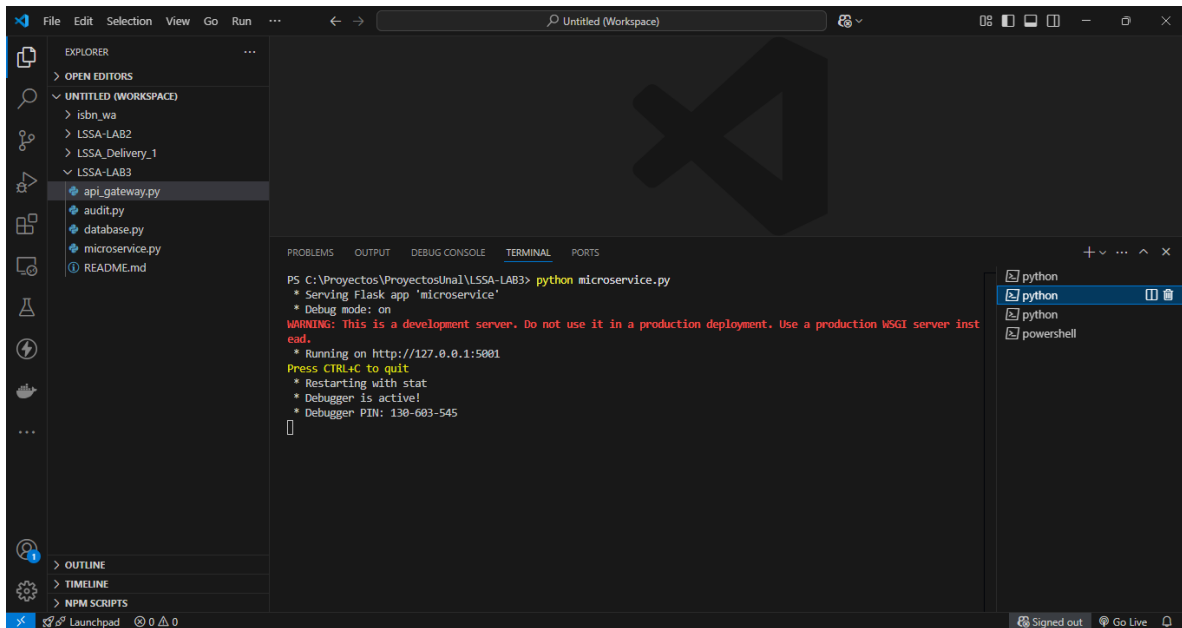


## Step 4: Test the Architecture

### 1. API Gateway:



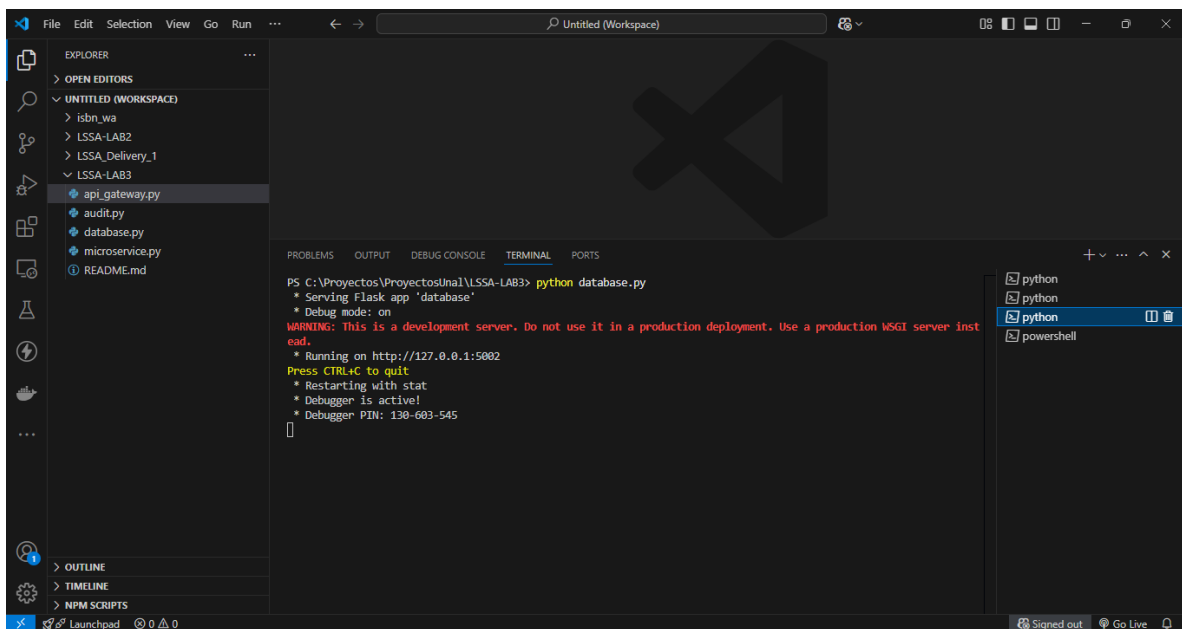
## 2. Microservice:



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The file explorer shows a workspace named 'UNTITLED (WORKSPACE)' containing a folder 'LSSA-LAB3' with files 'api\_gateway.py', 'audit.py', 'database.py', 'microservice.py', and 'README.md'. The 'microservice.py' file is selected. The terminal window at the bottom shows the command 'python microservice.py' being executed. The output indicates that the Flask app 'microservice' is running on http://127.0.0.1:5001. A warning message states: 'WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.' The terminal also shows instructions to press CTRL+C to quit, restart with 'stat', and provides the debugger PIN: 130-603-545.

```
PS C:\Proyectos\ProyectosUnal\LSSA-LAB3> python microservice.py
* Serving Flask app 'microservice'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5001
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 130-603-545
```

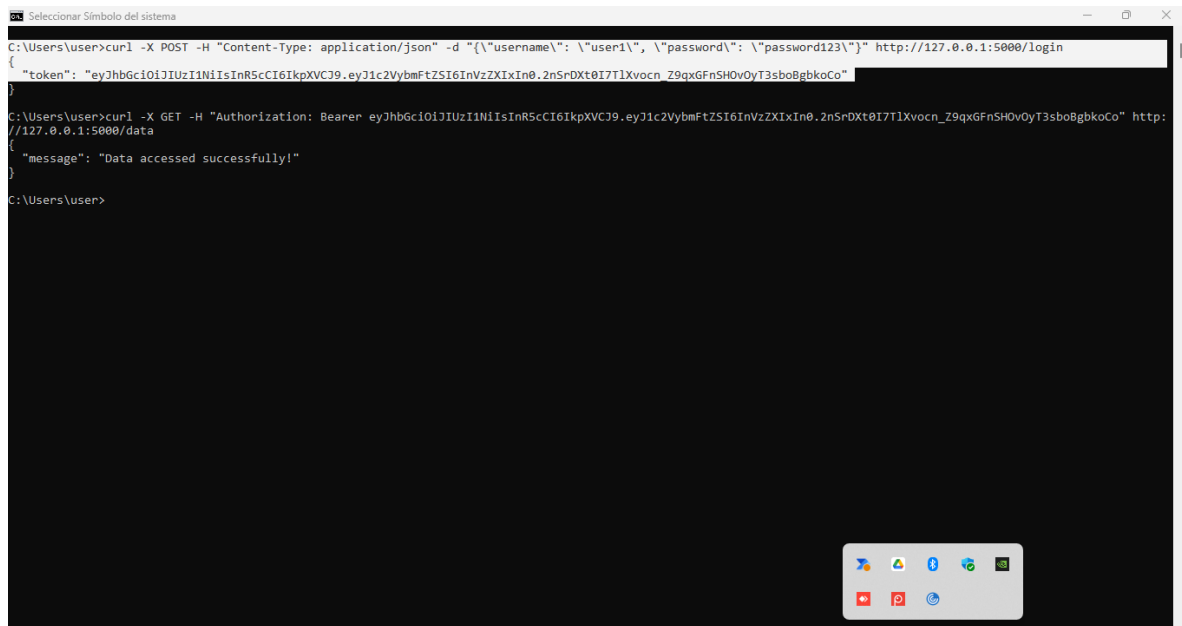
## 3. Database:



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The file explorer shows a workspace named 'UNTITLED (WORKSPACE)' containing a folder 'LSSA-LAB3' with files 'api\_gateway.py', 'audit.py', 'database.py', 'microservice.py', and 'README.md'. The 'database.py' file is selected. The terminal window at the bottom shows the command 'python database.py' being executed. The output indicates that the Flask app 'database' is running on http://127.0.0.1:5002. A warning message states: 'WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.' The terminal also shows instructions to press CTRL+C to quit, restart with 'stat', and provides the debugger PIN: 130-603-545.

```
PS C:\Proyectos\ProyectosUnal\LSSA-LAB3> python database.py
* Serving Flask app 'database'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5002
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 130-603-545
```

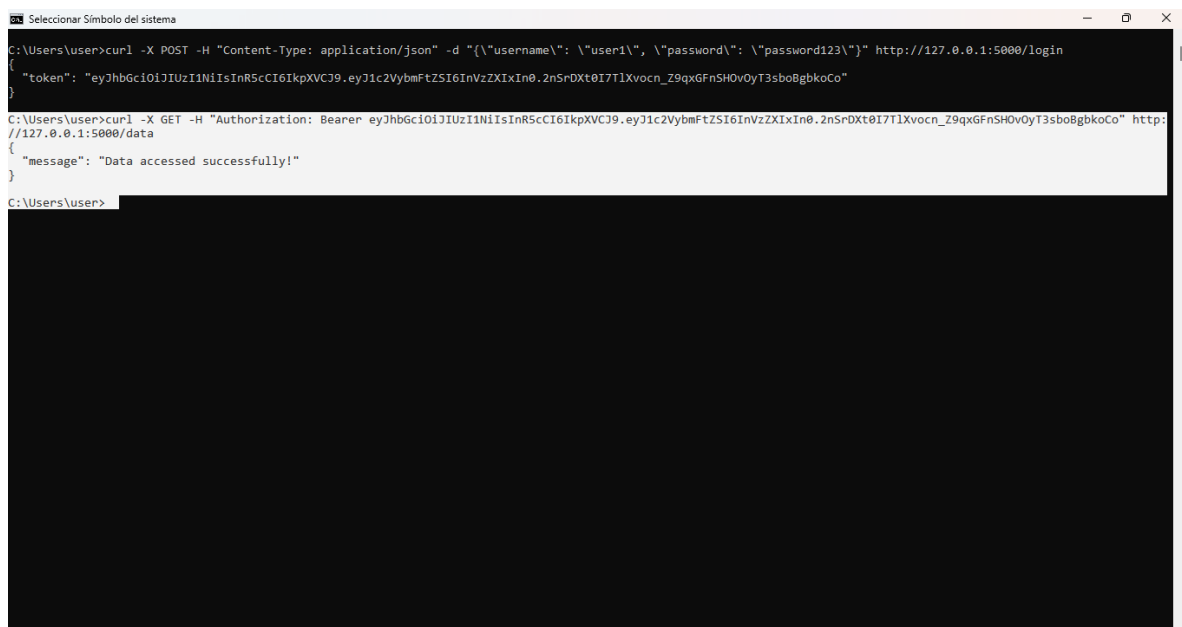
## 4. Obtain JWT Token



A terminal window titled "Seleccionar Símbolo del sistema" with a dark background. It shows the execution of two curl commands. The first is a POST request to http://127.0.0.1:5000/login with a JSON body containing username and password. The response is a JSON object with a token. The second is a GET request to http://127.0.0.1:5000/data with the Bearer token in the Authorization header. The response is a JSON object with a success message. A Windows taskbar is visible at the bottom right.

```
Selecciónar Símbolo del sistema
C:\Users\user>curl -X POST -H "Content-Type: application/json" -d '{"username": "user1", "password": "password123"}' http://127.0.0.1:5000/login
{"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InVzZXIuIn0.2nSrDXt0I7TlXvocn_Z9qxGFnSH0vOyT3sboBgbkoCo"}
C:\Users\user>curl -X GET -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InVzZXIuIn0.2nSrDXt0I7TlXvocn_Z9qxGFnSH0vOyT3sboBgbkoCo" http://127.0.0.1:5000/data
{"message": "Data accessed successfully!"}
C:\Users\user>
```

## 5. Access Protected Routes



A terminal window titled "Seleccionar Símbolo del sistema" with a dark background. It shows the same two curl commands as in the previous block. The first command is a POST request to http://127.0.0.1:5000/login, and the second is a GET request to http://127.0.0.1:5000/data. The responses are the same as in the previous block. The terminal window is larger, showing more of the dark background.

```
Selecciónar Símbolo del sistema
C:\Users\user>curl -X POST -H "Content-Type: application/json" -d '{"username": "user1", "password": "password123"}' http://127.0.0.1:5000/login
{"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InVzZXIuIn0.2nSrDXt0I7TlXvocn_Z9qxGFnSH0vOyT3sboBgbkoCo"}
C:\Users\user>curl -X GET -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InVzZXIuIn0.2nSrDXt0I7TlXvocn_Z9qxGFnSH0vOyT3sboBgbkoCo" http://127.0.0.1:5000/data
{"message": "Data accessed successfully!"}
C:\Users\user>
```