

PRUEBA 2

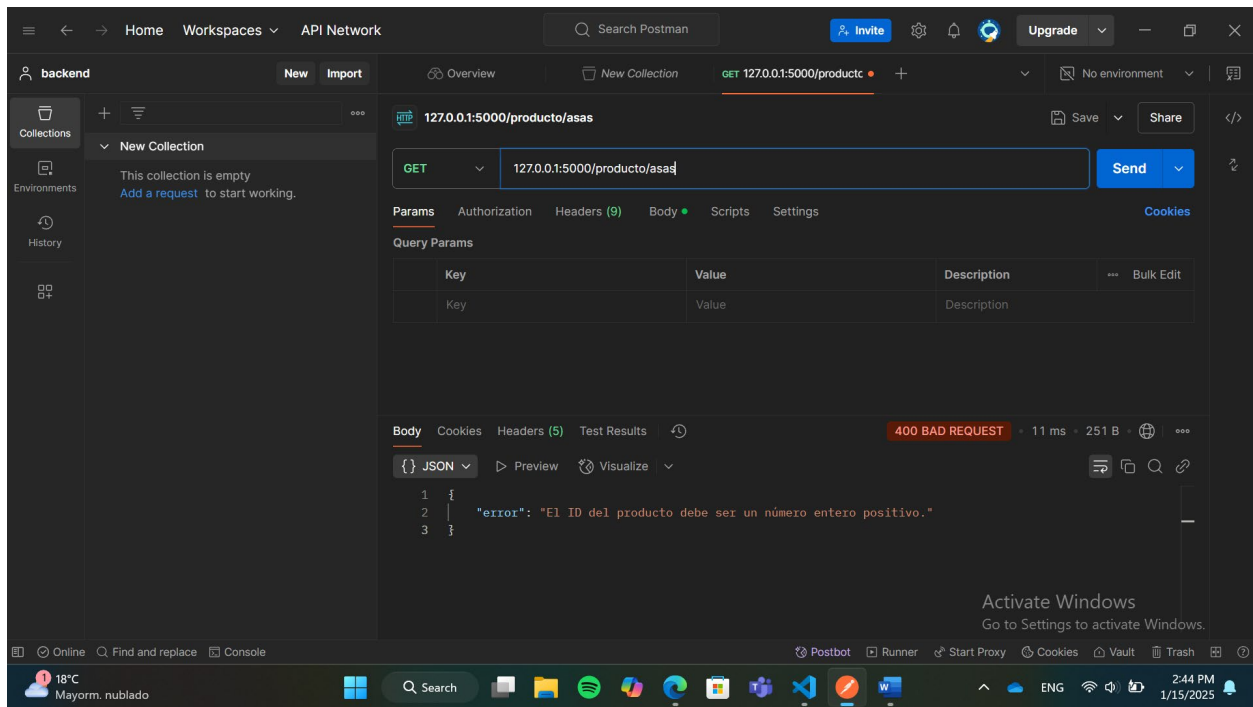
Nombre: Alejandro Alvarez

METODO CONSULTAR

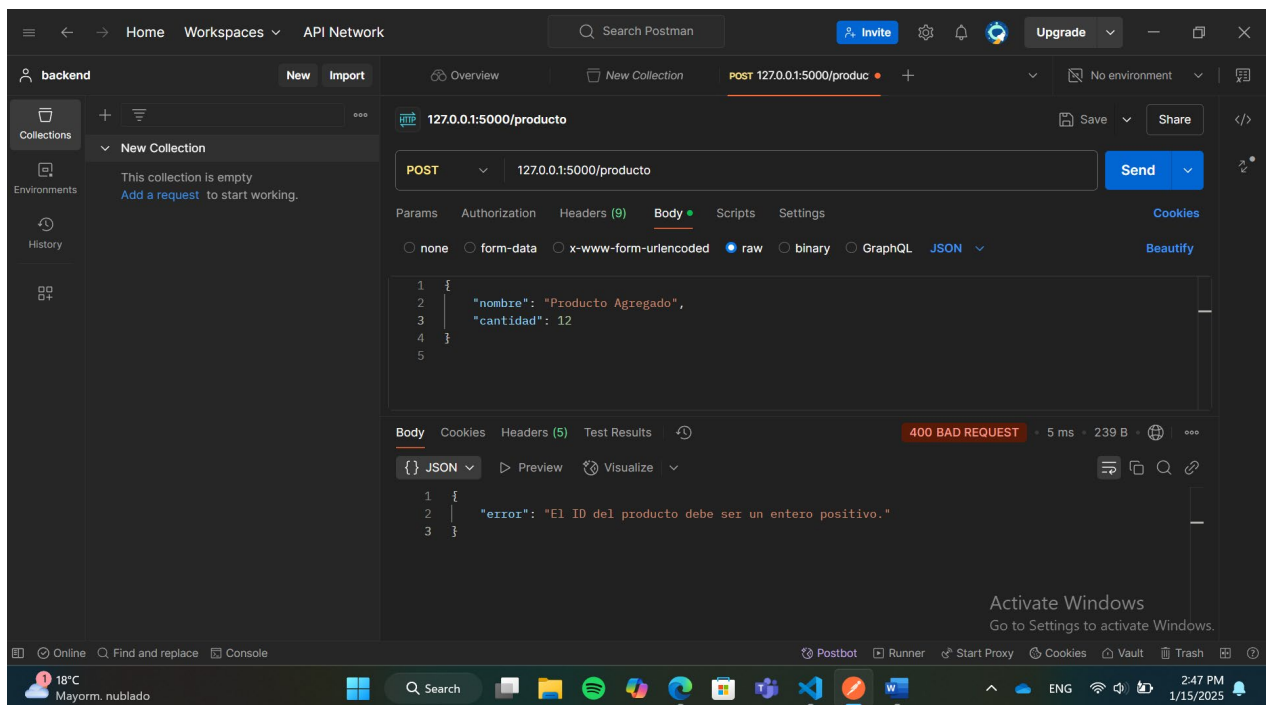
The screenshot displays the Postman application interface. On the left sidebar, the 'Collections' tab is active, showing a 'New Collection' button and a message that the collection is empty. The main workspace shows a GET request to the URL '127.0.0.1:5000/producto/5'. The 'Params' tab is selected, showing a table for Query Params with columns for Key, Value, and Description. The 'Body' tab is also visible, showing a JSON response. The response status is '200 OK' with a response time of 6 ms and a size of 223 B. The JSON body contains the following data:

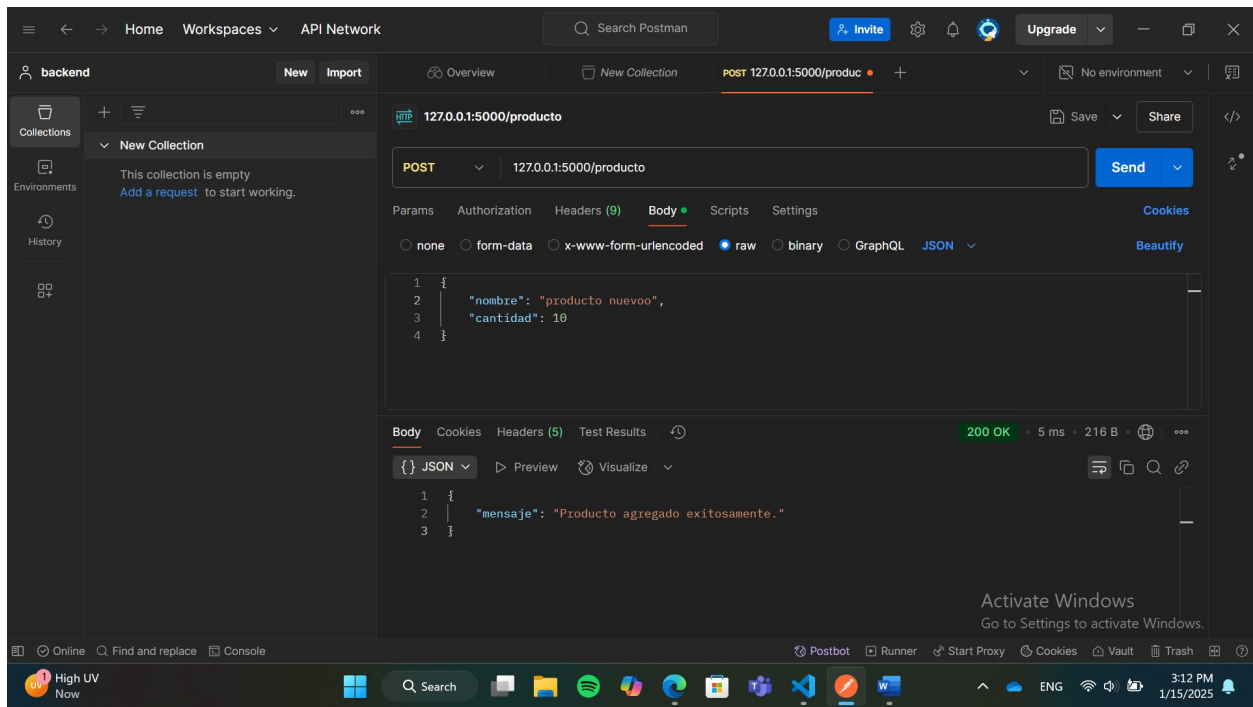
```
{  "cantidad": 60,  "id": 5,  "nombre": "Producto E"}
```

The Windows taskbar at the bottom shows the system clock as 2:44 PM on 1/15/2025, along with various application icons and network status.

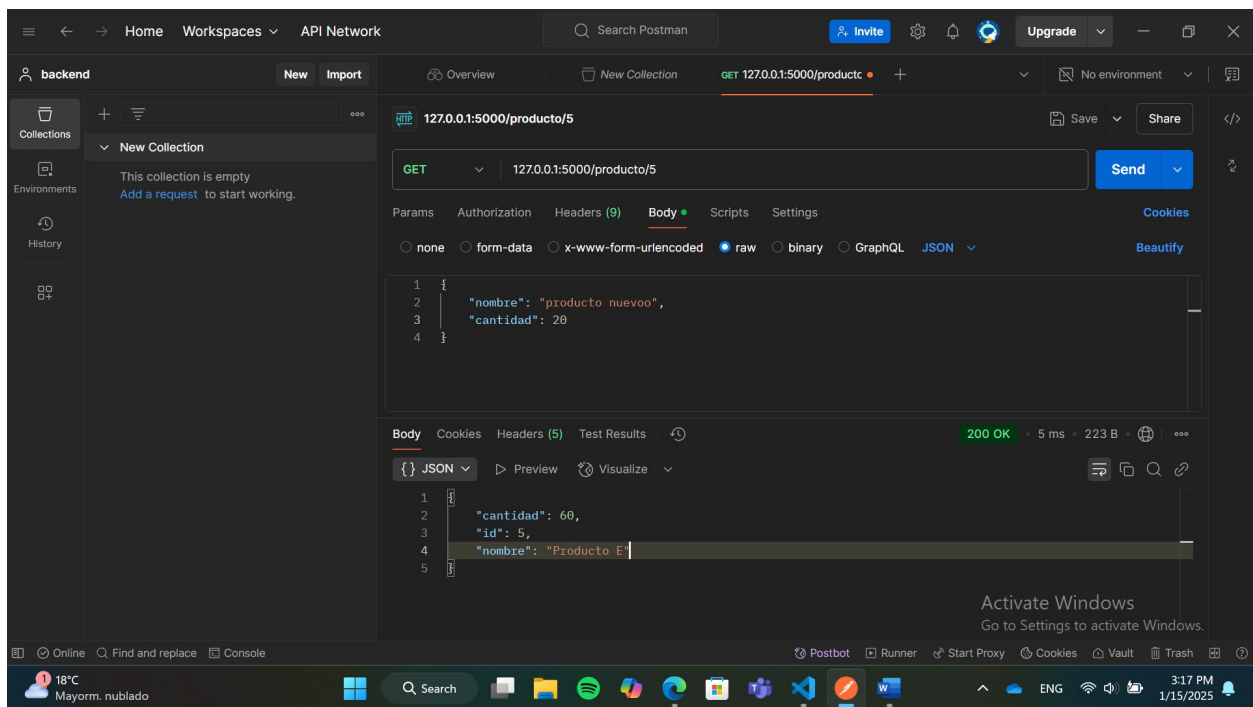


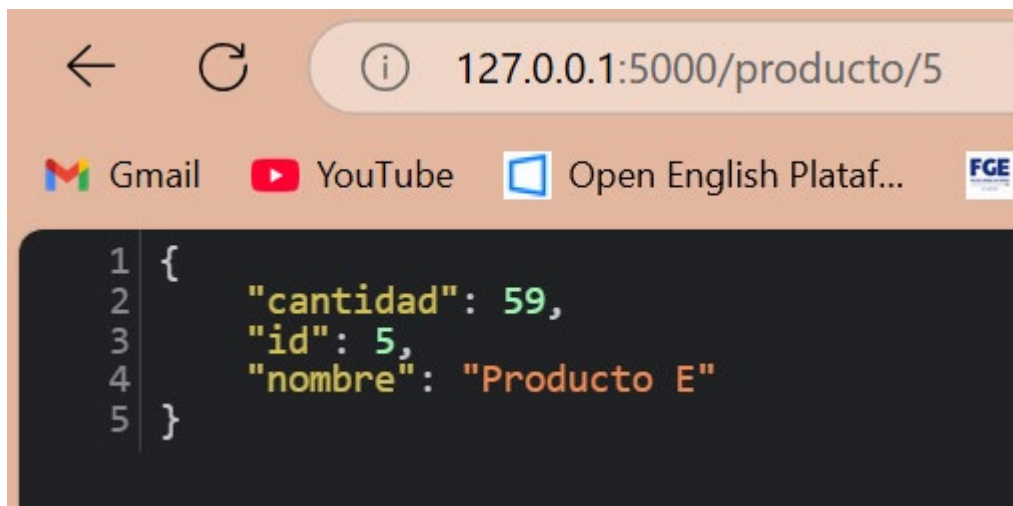
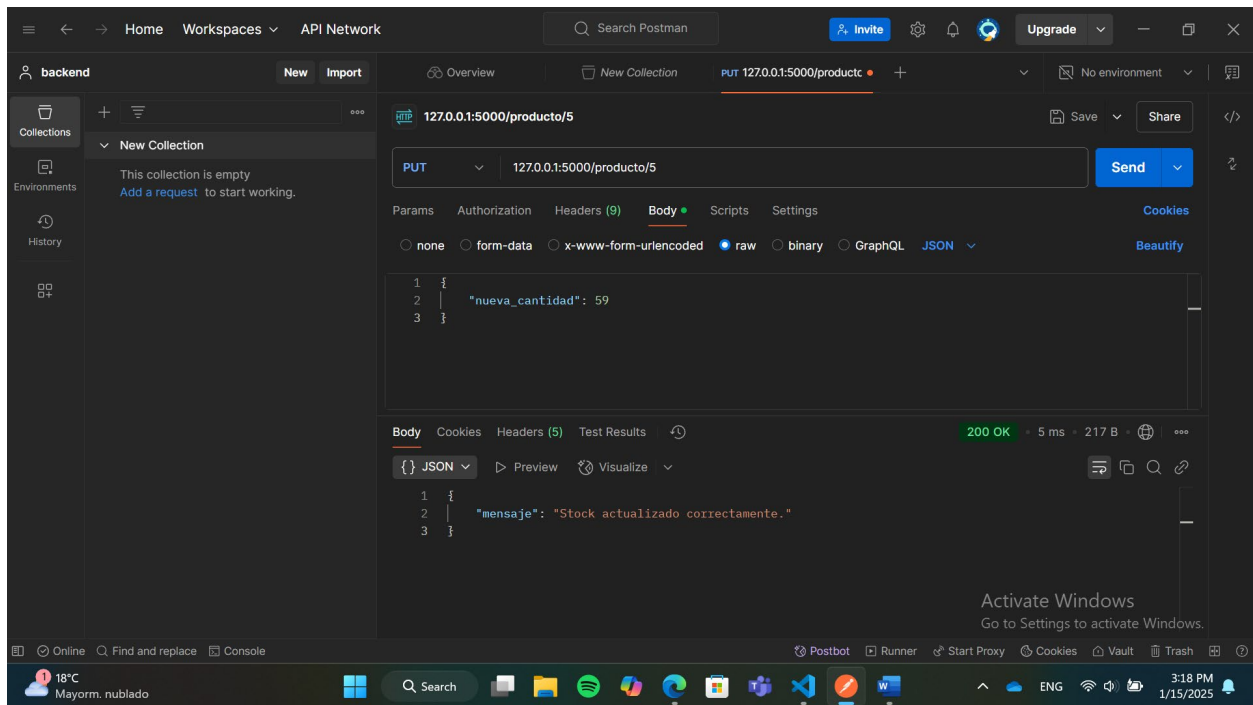
METODO AGREGAR





METODO ACTUALIZAR





PRUEBAS UNITARIOS CON UNITTEST

Para inventario.py

```
aseba@Alejandro MINGW64 ~/OneDrive - Escuela Politécnica Nacional/Escritorio/Prueba2
$ python -m unittest pruebasInventario.py
....Producto existente Producto A con cantidad 100
Producto actualizado Producto A con nueva cantidad 120
....
-----
Ran 8 tests in 0.003s
```

Para app.py

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer:** A project named 'PRUEBA2' is open, containing files 'app.py', 'inventario.py', 'testAPI.py', and 'testInventario.py'. 'testAPI.py' is selected.
- Editor:** Displays the code for 'testAPI.py'. It defines a 'TestAPI' class inheriting from 'unittest.TestCase'. The class contains two test methods:
 - `test_actualizar_stock_api(self)`: Tests the PUT endpoint for updating stock. It uses `app.test_client()` to send a PUT request with JSON data `{ "nueva_cantidad": 120 }`. It asserts that the status code is 200 and that the response data matches `{ "mensaje": "Stock actualizado correctamente." }`.
 - `test_actualizar_stock_producto_no_existente(self)`: (Partially visible at the bottom).
- Terminal:** Shows the output of running the tests. It displays a JSON object `{ 'cantidad': 100, 'id': 1, 'nombre': 'Producto A' }`, followed by a separator line and the message `Ran 3 tests in 0.015s`. Below this, it indicates `FAILED (failures=1)`.
- Status Bar:** Shows the file encoding as UTF-8, line endings as CRLF, and the Python version as 3.12.3.