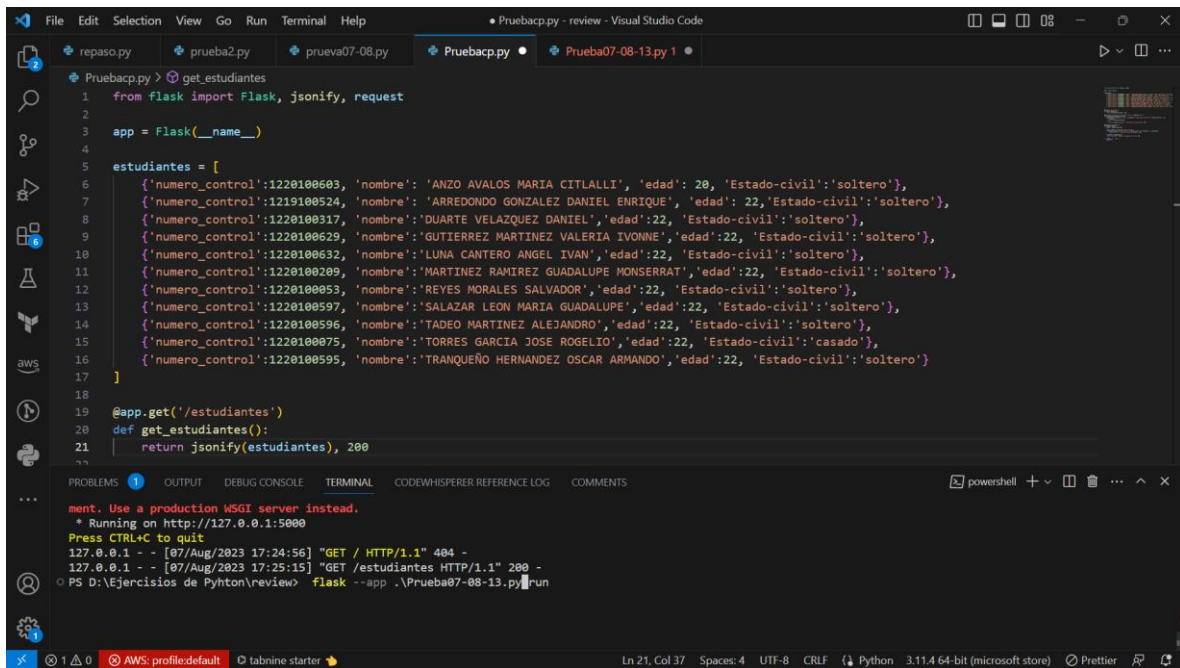


Práctica del ejercicio

- Agregar al menos dos datos más al estudiante, por ejemplo: edad, carrera, etc
- Para el método **ADD** valida qué no se agregue un número de control ya existente.
- Modificar todos los métodos para que los nuevos campos tengan efecto.

Get

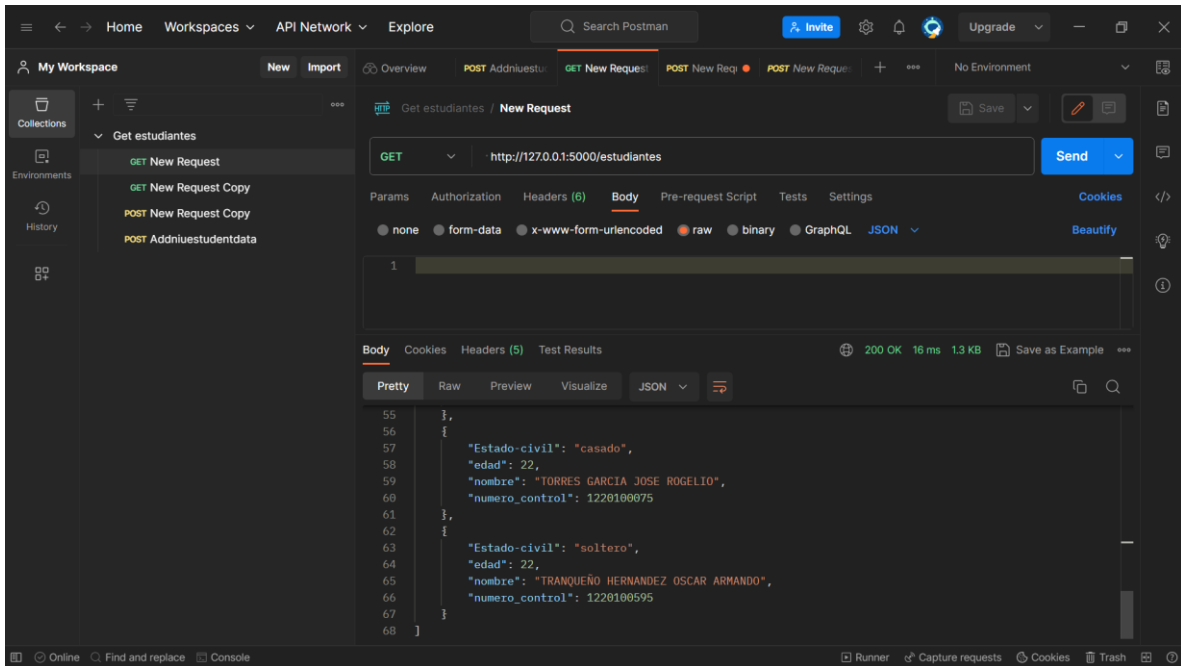


```
File Edit Selection View Go Run Terminal Help
• Pruebacpp.py - review - Visual Studio Code

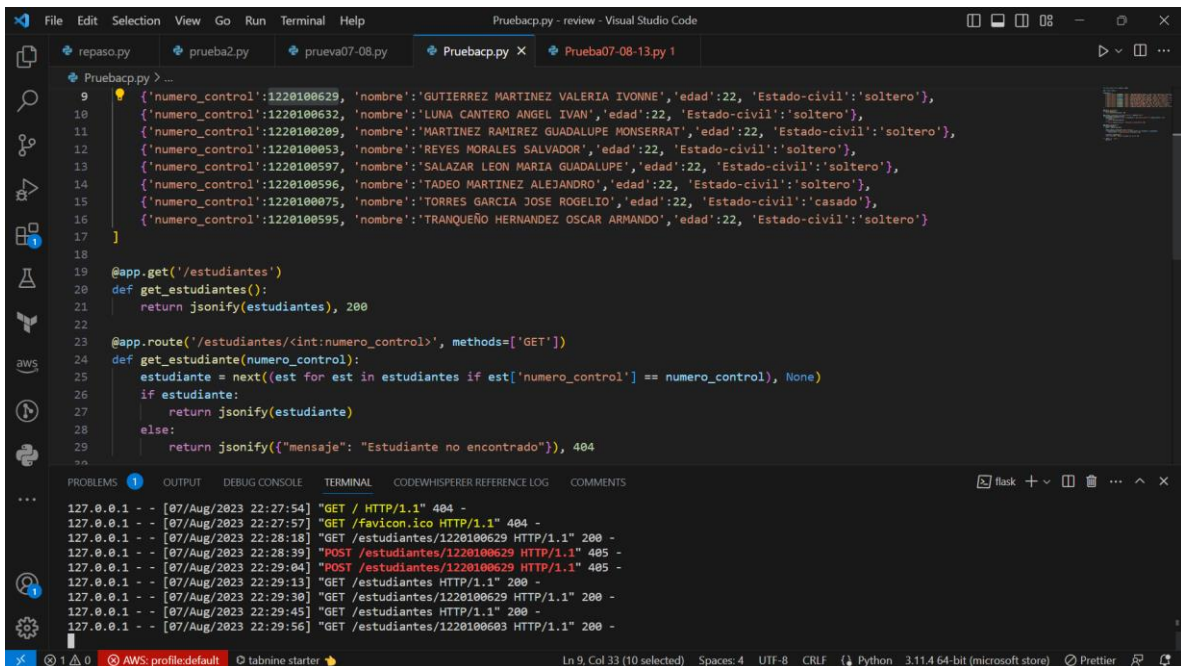
repaso.py prueba2.py prueba07-08.py Pruebacpp.py Prueba07-08-13.py 1

Pruebacpp.py > get_estudiantes
1 from flask import Flask, jsonify, request
2
3 app = Flask(__name__)
4
5 estudiantes = [
6     {'numero_control':1220100603, 'nombre': 'ANZO AVALOS MARIA CITLALLI', 'edad': 20, 'Estado-civil':'soltero'},
7     {'numero_control':1219100524, 'nombre': 'ARREDONDO GONZALEZ DANIEL ENRIQUE', 'edad': 22, 'Estado-civil':'soltero'},
8     {'numero_control':1220100317, 'nombre': 'DUARTE VELAZQUEZ DANIEL', 'edad':22, 'Estado-civil':'soltero'},
9     {'numero_control':1220100629, 'nombre': 'GUTIERREZ MARTINEZ VALERIA IVONNE', 'edad':22, 'Estado-civil':'soltero'},
10    {'numero_control':1220100632, 'nombre': 'LUNA CANTERO ANGEL IVAN', 'edad':22, 'Estado-civil':'soltero'},
11    {'numero_control':1220100209, 'nombre': 'MARTINEZ RAMIREZ GUADALUPE MONSERRAT', 'edad':22, 'Estado-civil':'soltero'},
12    {'numero_control':1220100053, 'nombre': 'REYES MORALES SALVADOR', 'edad':22, 'Estado-civil':'soltero'},
13    {'numero_control':1220100597, 'nombre': 'SALAZAR LEON MARIA GUADALUPE', 'edad':22, 'Estado-civil':'soltero'},
14    {'numero_control':1220100596, 'nombre': 'TADEO MARTINEZ ALEJANDRO', 'edad':22, 'Estado-civil':'soltero'},
15    {'numero_control':1220100075, 'nombre': 'TORRES GARCIA JOSE ROGELIO', 'edad':22, 'Estado-civil':'casado'},
16    {'numero_control':1220100595, 'nombre': 'TRANQUEÑO HERNANDEZ OSCAR ARMANDO', 'edad':22, 'Estado-civil':'soltero'}
17 ]
18
19 @app.get('/estudiantes')
20 def get_estudiantes():
21     return jsonify(estudiantes), 200

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG COMMENTS
ment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [07/Aug/2023 17:24:56] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [07/Aug/2023 17:25:15] "GET /estudiantes HTTP/1.1" 200 -
PS D:\Ejercicios de Python\review> flask --app .\Prueba07-08-13.py run
```



Traer alumnos por numero de control.



Oscar Armando Tranqueño Hernandez

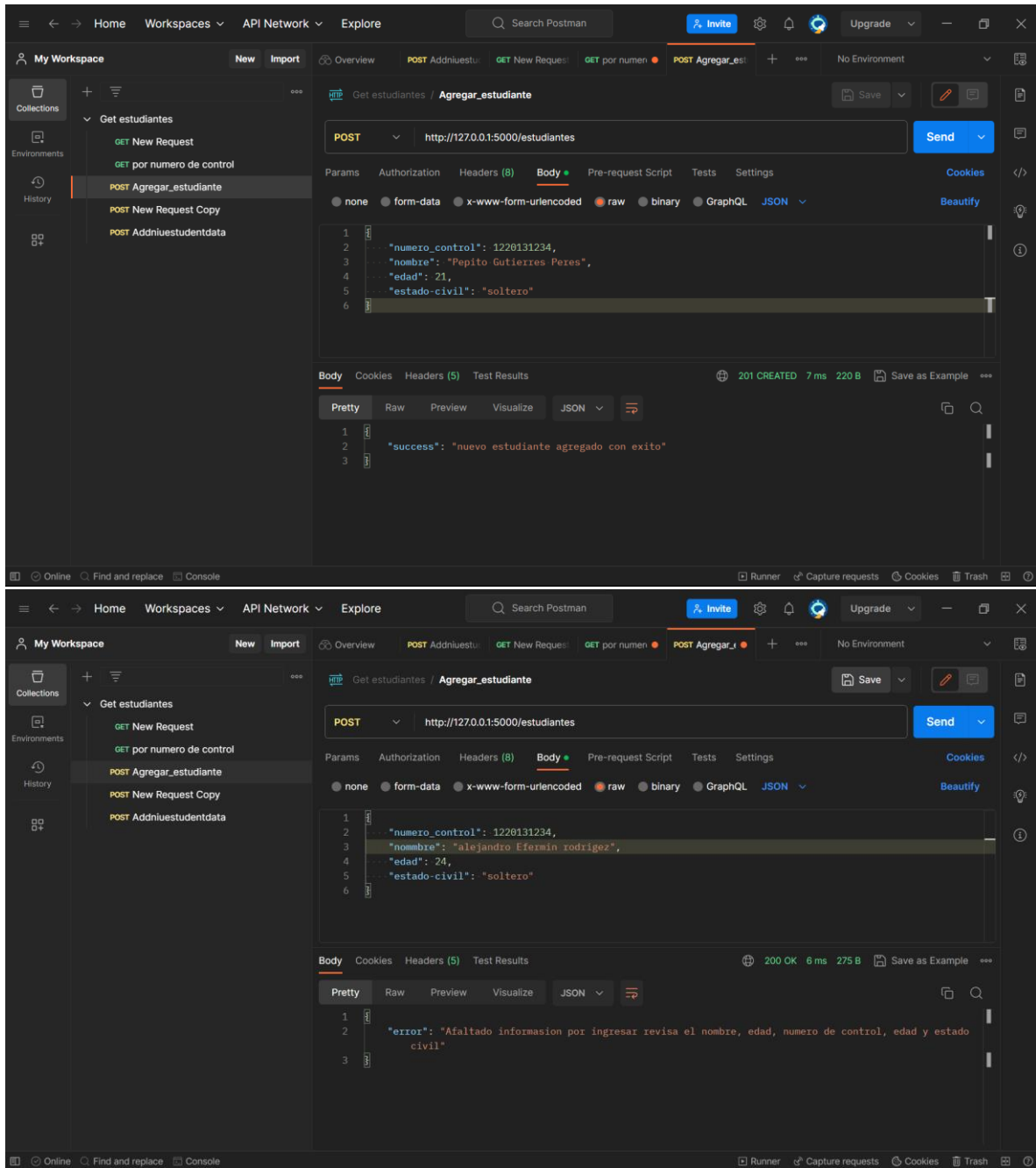
The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', and 'Explore'. The left sidebar shows 'My Workspace' with a collection named 'Get estudiantes' containing a request 'por numero de control'. The main panel displays a GET request to 'http://127.0.0.1:5000/estudiantes/1220100603'. The 'Body' tab is selected, showing a JSON response with status '200 OK', '9 ms', and '269 B'. The JSON data is as follows:

```
1 {
2   "Estado-civil": "soltero",
3   "edad": 20,
4   "nombre": "ANZO AVALOS MARIA CITLALLI",
5   "numero_control": 1220100603
6 }
```

The screenshot shows the Postman application interface with a failed GET request. The main panel displays a GET request to 'http://127.0.0.1:5000/estudiantes/1220100609'. The 'Body' tab is selected, showing a JSON response with status '404 NOT FOUND', '13 ms', and '211 B'. The JSON data is as follows:

```
1 {
2   "mensaje": "Estudiante no encontrado"
3 }
```

Post new estudiante



The image shows a development environment with Postman and Visual Studio Code. In Postman, a GET request is configured to `http://127.0.0.1:5000/estudiantes`. The response is a JSON array of two student objects. In VS Code, the `Pruebas.py` file contains a Flask application with endpoints for creating, retrieving, and updating students.

Postman Request:

- Method: GET
- URL: `http://127.0.0.1:5000/estudiantes`
- Status: 200 OK, 7 ms, 1.39 KB

Postman Response (JSON):

```
[{"numero_control": 1220100595, "edad": 21, "estado_civil": "soltero", "nombre": "Pepito Gutierrez Peres", "numero_control": 1220131234}]
```

VS Code Code (Pruebas.py):

```
20 def get_estudiantes():
21     return jsonify(estudiantes), 200
22
23 @app.route('/estudiantes/<int:numero_control>', methods=['GET'])
24 def get_estudiante(numero_control):
25     estudiante = next((est for est in estudiantes if est['numero_control'] == numero_control), None)
26     if estudiante:
27         return jsonify(estudiante)
28     else:
29         return jsonify({"mensaje": "Estudiante no encontrado"}), 404
30
31 #newStudent
32
33 @app.post('/estudiantes')
34 def newStudent():
35     datos=request.get_json()
36     if 'numero_control' not in datos or 'nombre' not in datos or 'edad' not in datos or 'estado-civil' not in datos:
37         return {'error': 'Afaltado informasion por ingresar revisa el nombre, edad, numero de control, edad y estado civil'}
38     estudiantes.append(datos)
39     return {'success': 'nuevo estudiante agregado con exito'},201
40
41
```

VS Code Terminal Output:

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [07/Aug/2023 22:57:46] "POST /estudiantes HTTP/1.1" 201 -
127.0.0.1 - - [07/Aug/2023 22:59:40] "POST /estudiantes HTTP/1.1" 400 -
127.0.0.1 - - [07/Aug/2023 22:59:55] "POST /estudiantes HTTP/1.1" 400 -
127.0.0.1 - - [07/Aug/2023 23:00:11] "POST /estudiantes HTTP/1.1" 200 -
127.0.0.1 - - [07/Aug/2023 23:01:18] "GET /estudiantes HTTP/1.1" 200 -
```