

Alejandro Tadeo Martínez

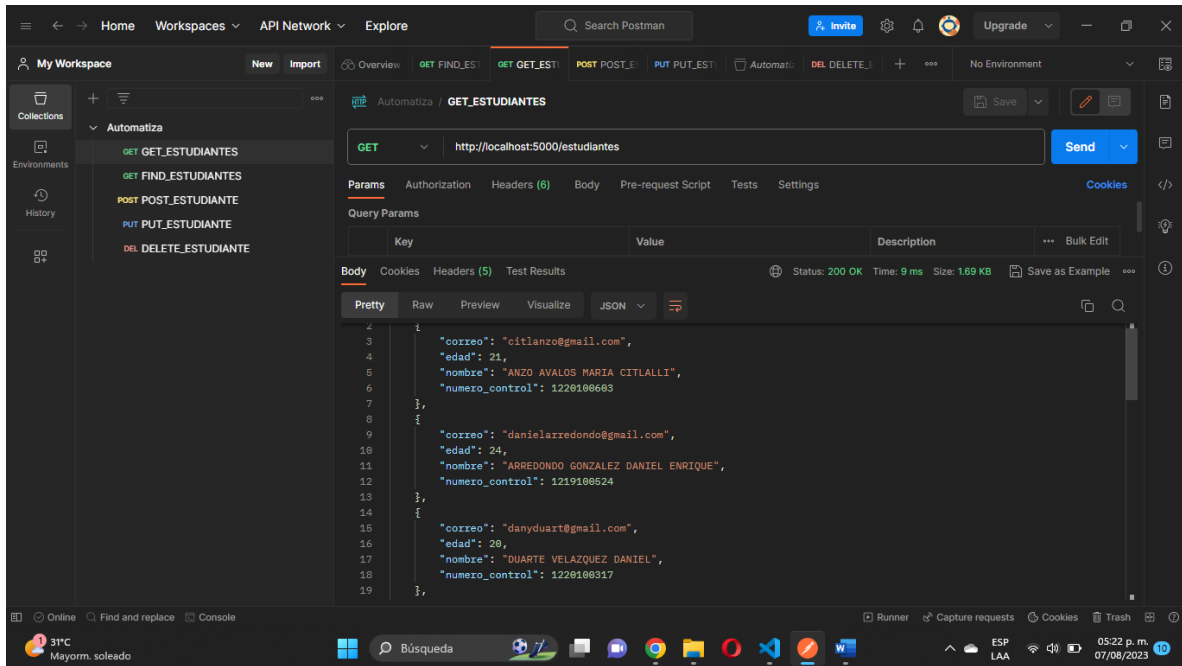
Grupo: GRIC3091

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.

Método GET

```
app.py  repaso.py  students.py X  servidor.py
students.py > ...
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':21, 'correo':'citlanzo@gmail.com'},
7      {'numero_control':1219100524, 'nombre':'ARREDONDO GONZALEZ DANIEL ENRIQUE', 'edad':24, 'correo':'danielarredondo@gmail.com'},
8      {'numero_control':1220100317, 'nombre':'DUARTE VELAZQUEZ DANIEL', 'edad':20, 'correo':'danyduart@gmail.com'},
9      {'numero_control':1220100629, 'nombre':'GUTIERREZ MARTINEZ VALERIA IVONNE', 'edad':21, 'correo':'valeriaivonne@gmail.com'},
10     {'numero_control':1220100632, 'nombre':'LUNA CANTERO ANGEL IVAN', 'edad':21, 'correo':'angelivan@gmail.com'},
11     {'numero_control':1220100209, 'nombre':'MARTINEZ RAMIREZ GUADALUPE MONSERRAT', 'edad':21, 'correo':'monse@gmail.com'},
12     {'numero_control':1220100053, 'nombre':'REYES MORALES SALVADOR', 'edad':21, 'correo':'salvador@gmail.com'},
13     {'numero_control':1220100597, 'nombre':'SALAZAR LEON MARIA GUADALUPE', 'edad':21, 'correo':'mleon@gmail.com'},
14     {'numero_control':1220100596, 'nombre':'TADEO MARTINEZ ALEJANDRO', 'edad':23, 'correo':'tadeo.alejandro.cbtis75@gmail.com'},
15     {'numero_control':1220100075, 'nombre':'TORRES GARCIA JOSE ROGELIO', 'edad':22, 'correo':'joserogtoga@gmail.com'},
16     {'numero_control':1220100595, 'nombre':'TRANQUEÑO HERNANDEZ OSCAR ARMANDO', 'edad':20, 'correo':'armasacre@gmail.com'}
17 ]
18
19 #Obtener toda la lista de estudiantes
20 @app.get('/estudiantes')
21 def get_estudiantes():
22     return jsonify(estudiantes)
```



Método GET_ID (NUMERO DE CONTROL)

```
{ 'numero_control':1220100629, 'nombre': 'GUTIERREZ MARTINEZ VALERIA IVONNE', 'edad':21, 'correo': 'valeriaivonne@gmail.com'},
{ 'numero_control':1220100632, 'nombre': 'LUNA CANTERO ANGEL IVAN', 'edad':21, 'correo': 'angelivan@gmail.com'},
{ 'numero_control':1220100209, 'nombre': 'MARTINEZ RAMIREZ GUADALUPE MONSERRAT', 'edad':21, 'correo': 'monse@gmail.com'},
{ 'numero_control':1220100053, 'nombre': 'REYES MORALES SALVADOR', 'edad':21, 'correo': 'salvador@gmail.com'},
{ 'numero_control':1220100597, 'nombre': 'SALAZAR LEON MARIA GUADALUPE', 'edad':21, 'correo': 'mleon@gmail.com'},
{ 'numero_control':1220100596, 'nombre': 'TADEO MARTINEZ ALEJANDRO', 'edad':23, 'correo': 'tadeo.alejandro.cbti575@gmail.com'},
{ 'numero_control':1220100075, 'nombre': 'TORRES GARCIA JOSE ROGELIO', 'edad':22, 'correo': 'joserogtoga@gmail.com'},
{ 'numero_control':1220100595, 'nombre': 'TRANQUEÑO HERNANDEZ OSCAR ARMANDO', 'edad':20, 'correo': 'armasacre@gmail.com'}

]
#Obtener toda la lista de estudiantes
@app.get('/estudiantes')
def get_estudiantes():
    return jsonify(estudiantes)

#Obtener Estudiante por numero de control
@app.get('/estudiantes/<int:numero_control>')
def get_numero_control(numero_control):
    estudiante = next((i for i in estudiantes if i['numero_control'] == numero_control), None)
    if estudiante:
        return jsonify(estudiante)
    else:
        return {'error': "No se encontró ningún estudiante con ese número de control."}, 404
```

Postman interface showing a REST client request for the endpoint `http://localhost:5000/estudiantes/1220100597`. The request is a `GET` method. The response status is `200 OK` with a time of `14 ms` and a size of `290 B`. The response body is a JSON object:

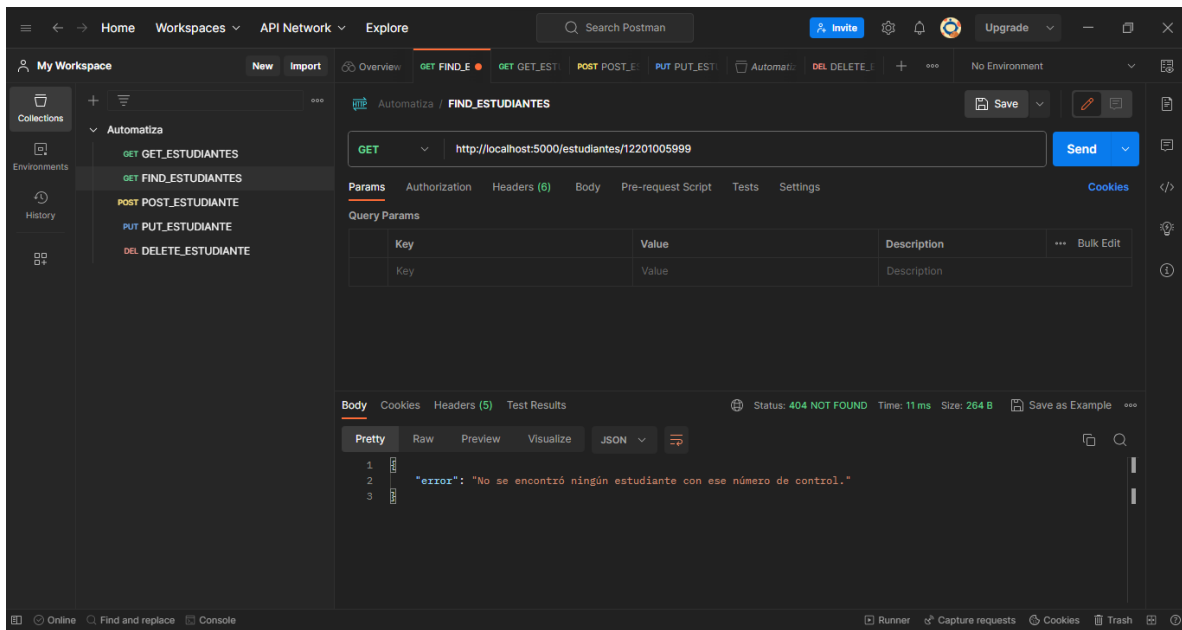
```
1 {
2   "correo": "mleon@gmail.com",
3   "edad": 21,
4   "nombre": "SALAZAR LEON MARIA GUADALUPE",
5   "numero_control": 1228188597
6 }
```

The interface also shows a sidebar with collections and environments, and a bottom status bar with system information.

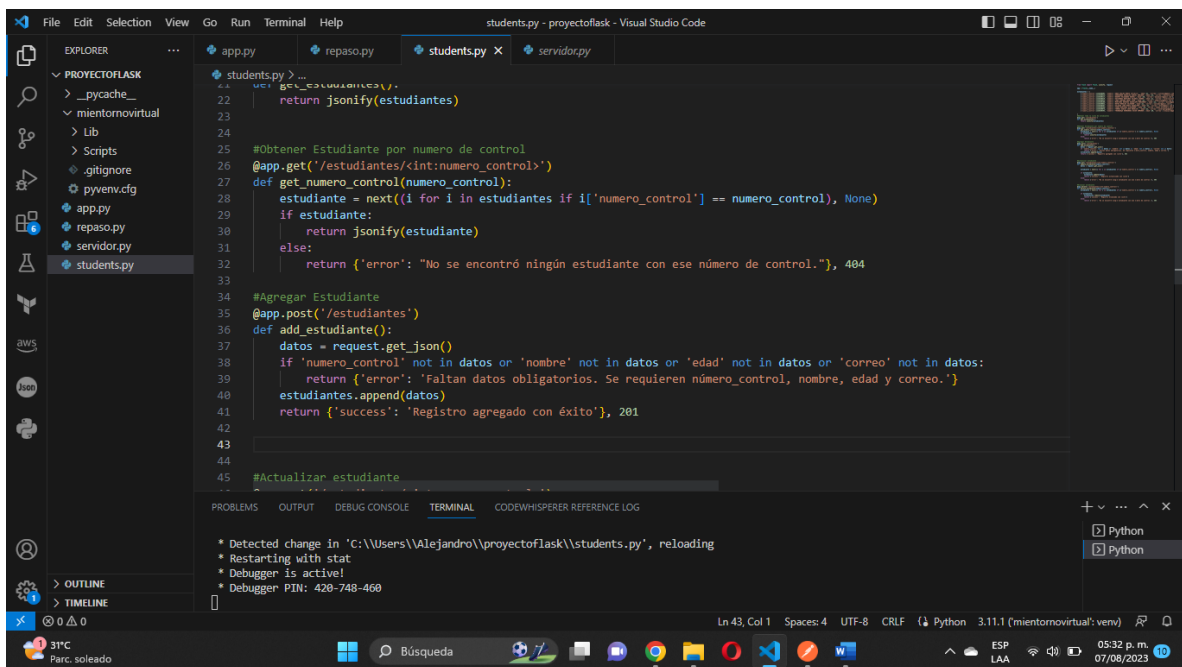
Postman interface showing a REST client request for the endpoint `http://localhost:5000/estudiantes/1220100596`. The request is a `GET` method. The response status is `200 OK` with a time of `13 ms` and a size of `304 B`. The response body is a JSON object:

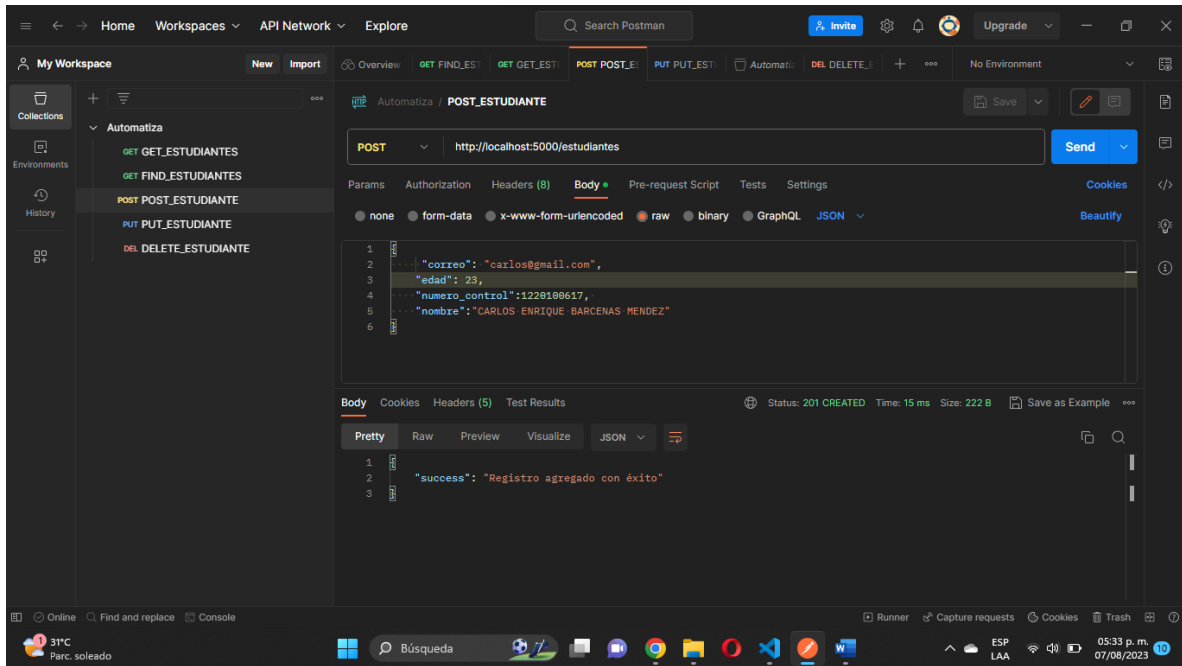
```
1 {
2   "correo": "tadeo.alejandro.cbtis75@gmail.com",
3   "edad": 23,
4   "nombre": "TADEO MARTINEZ ALEJANDRO",
5   "numero_control": 1228188596
6 }
```

The interface also shows a sidebar with collections and environments, and a bottom status bar with system information.

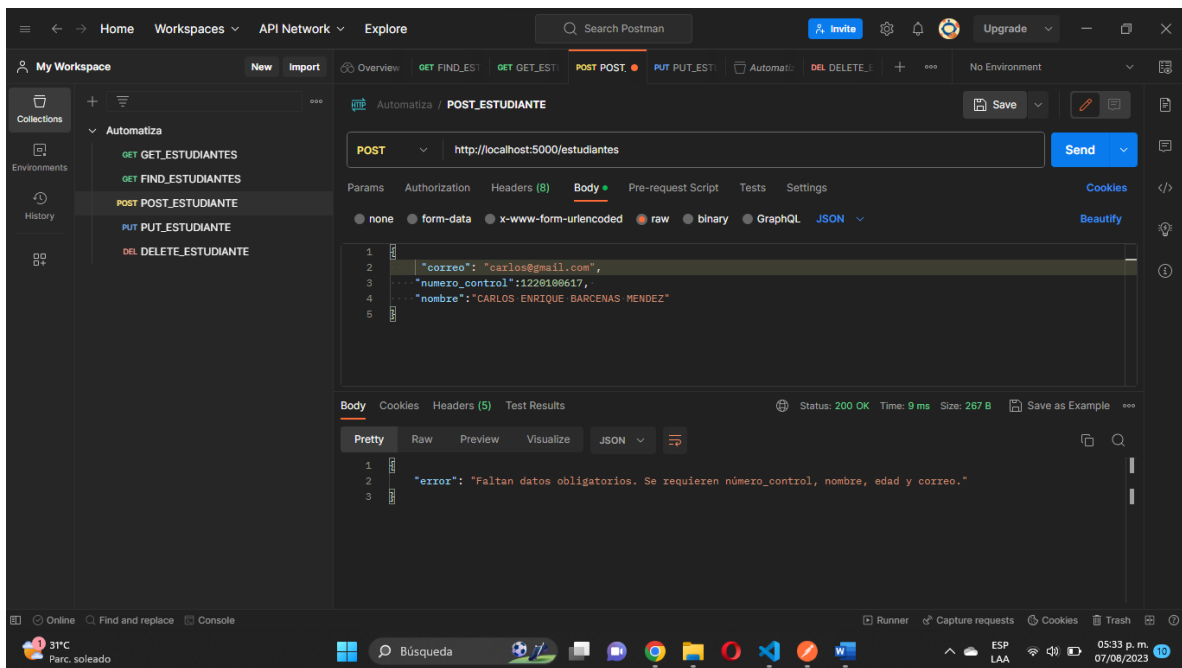


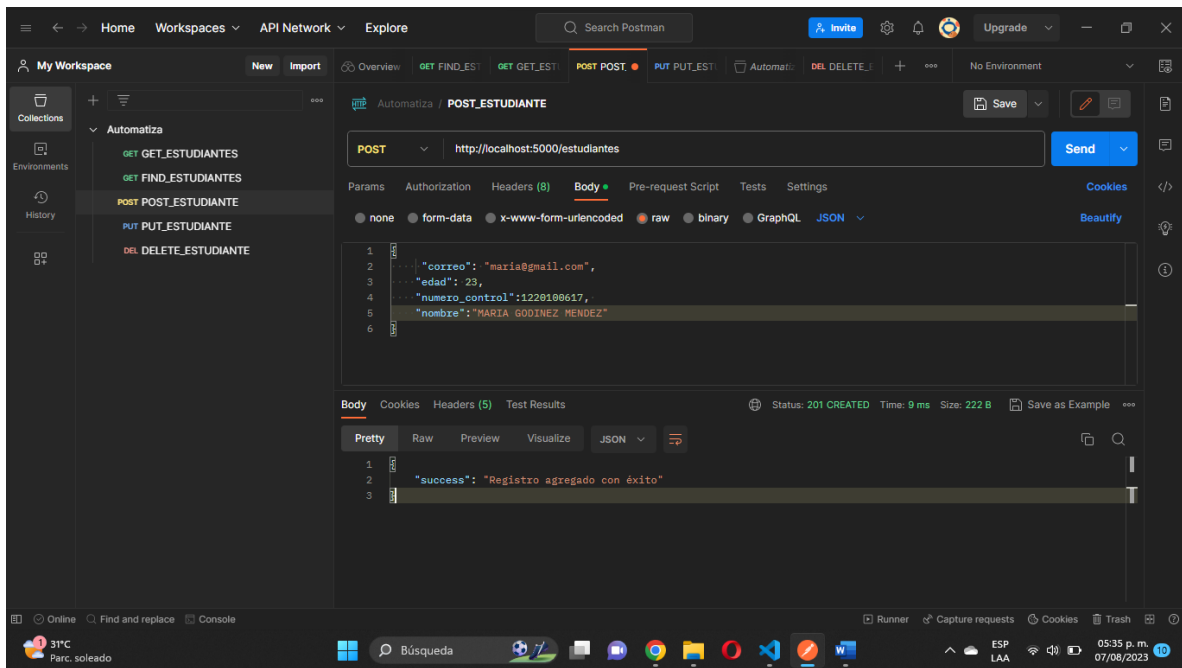
Método POST



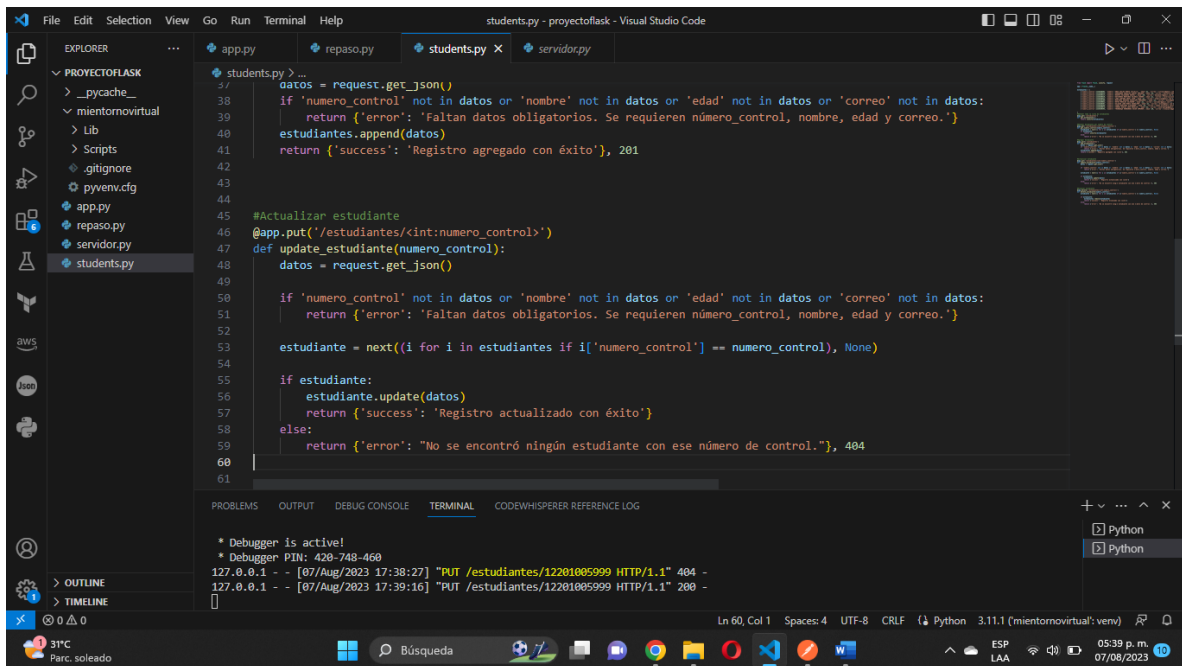


Y si falta algún dato falta esto





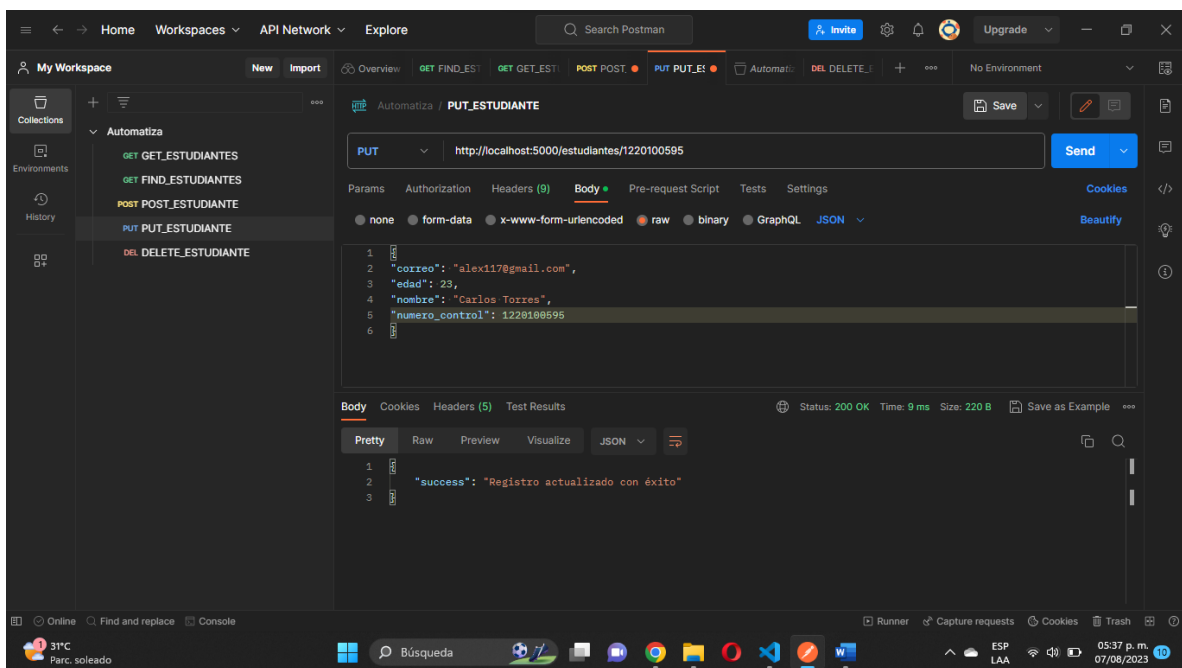
Método PUT



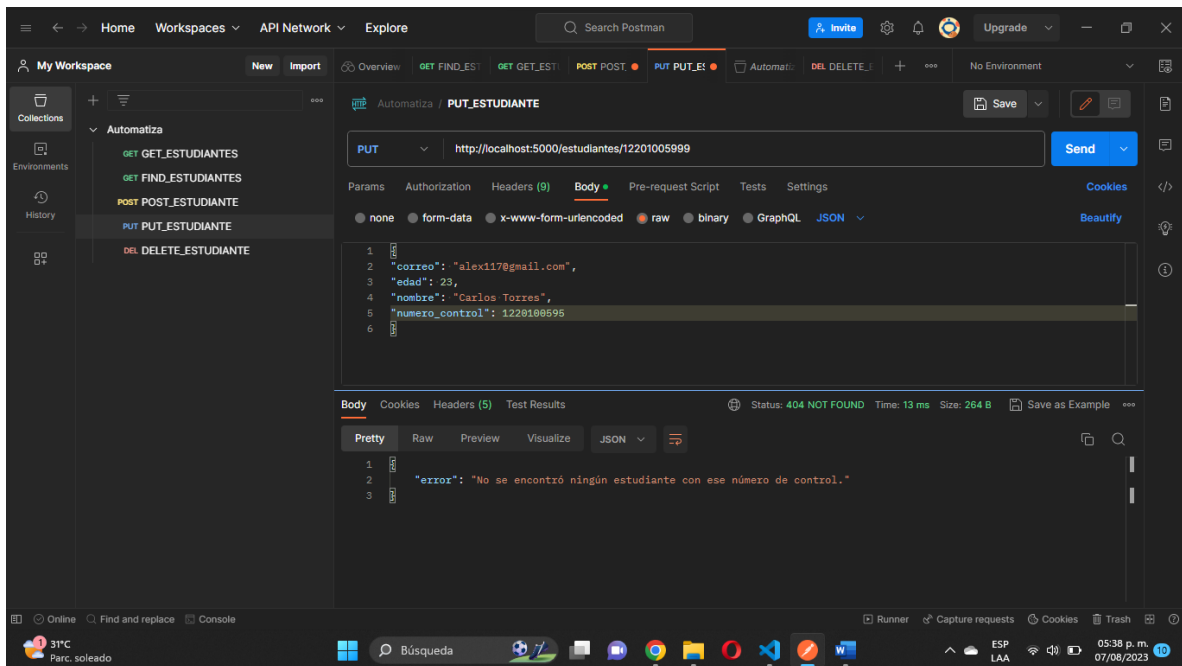
```
students.py
37
38 datos = request.get_json()
39 if 'numero_control' not in datos or 'nombre' not in datos or 'edad' not in datos or 'correo' not in datos:
40     return {'error': 'Faltan datos obligatorios. Se requieren número_control, nombre, edad y correo.'}
41     estudiantes.append(datos)
42     return {'success': 'Registro agregado con éxito'}, 201
43
44 #Actualizar estudiante
45 @app.put('/estudiantes/<int:numero_control>')
46 def update_estudiante(numero_control):
47     datos = request.get_json()
48
49     if 'numero_control' not in datos or 'nombre' not in datos or 'edad' not in datos or 'correo' not in datos:
50         return {'error': 'Faltan datos obligatorios. Se requieren número_control, nombre, edad y correo.'}
51
52     estudiante = next((i for i in estudiantes if i['numero_control'] == numero_control), None)
53
54     if estudiante:
55         estudiante.update(datos)
56         return {'success': 'Registro actualizado con éxito'}
57     else:
58         return {'error': "No se encontró ningún estudiante con ese número de control."}, 404
59
60
61
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG

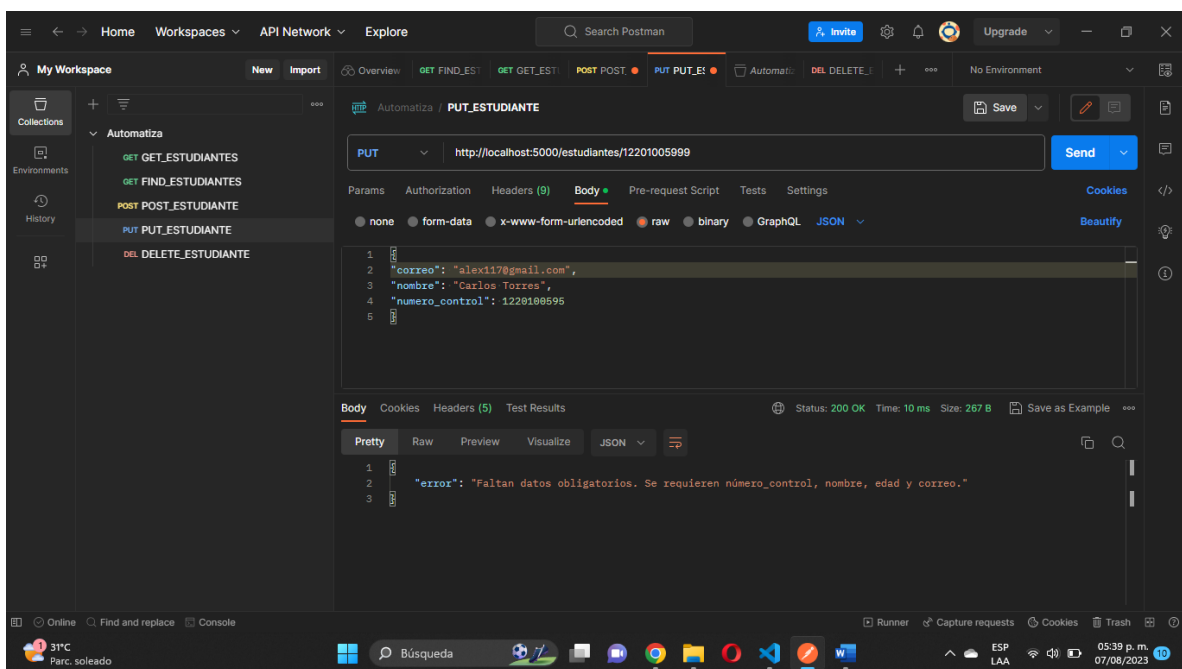
```
* Debugger is active!
* Debugger PIN: 428-748-468
127.0.0.1 - - [07/Aug/2023 17:38:27] "PUT /estudiantes/1220100599 HTTP/1.1" 404 -
127.0.0.1 - - [07/Aug/2023 17:39:16] "PUT /estudiantes/1220100599 HTTP/1.1" 200 -
```



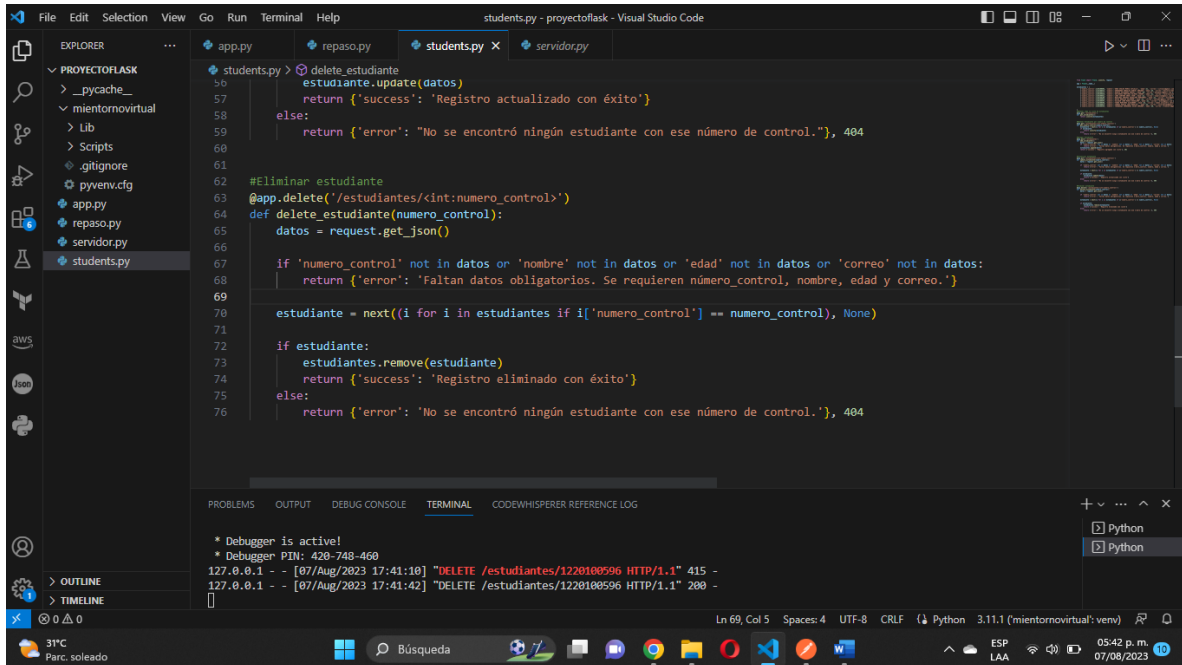
En caso de que no exista ese número de control



Y en caso de que falte algún dato.



Método DELETE



```
File Edit Selection View Go Run Terminal Help
students.py - proyectoflask - Visual Studio Code

EXPLORER
PROJECTOFLASK
  _pycache_
  mientornovirtual
  Lib
  Scripts
  .gitignore
  pyvenv.cfg
  app.py
  repaso.py
  servidor.py
  students.py

students.py
56 delete_estudiante
57     estudiante.update(datos)
58     return {'success': 'Registro actualizado con éxito'}
59 else:
60     return {'error': 'No se encontró ningún estudiante con ese número de control.'}, 404
61
62 #Eliminar estudiante
63 @app.delete('/estudiantes/<int:numero_control>')
64 def delete_estudiante(numero_control):
65     datos = request.get_json()
66
67     if 'numero_control' not in datos or 'nombre' not in datos or 'edad' not in datos or 'correo' not in datos:
68         return {'error': 'Faltan datos obligatorios. Se requieren número_control, nombre, edad y correo.'}
69
70     estudiante = next((i for i in estudiantes if i['numero_control'] == numero_control), None)
71
72     if estudiante:
73         estudiantes.remove(estudiante)
74         return {'success': 'Registro eliminado con éxito'}
75     else:
76         return {'error': 'No se encontró ningún estudiante con ese número de control.'}, 404

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG
* Debugger is active!
* Debugger PIN: 420-748-460
127.0.0.1 - - [07/Aug/2023 17:41:10] "DELETE /estudiantes/1220100596 HTTP/1.1" 415 -
127.0.0.1 - - [07/Aug/2023 17:41:42] "DELETE /estudiantes/1220100596 HTTP/1.1" 200 -
```

Postman interface showing a DELETE request to `http://localhost:5000/estudiantes/1220100596`. The request body is a JSON object:

```
{  "correo": "alex117@gmail.com",  "edad": 23,  "nombre": "Carlos Torres",  "numero_control": 1220100596}
```

The response status is **200 OK** with a body:

```
{  "success": "Registro eliminado con éxito"}
```

Postman interface showing a DELETE request to `http://localhost:5000/estudiantes/1220100599`. The request body is a JSON object:

```
{  "correo": "alex117@gmail.com",  "edad": 23,  "nombre": "Carlos Torres",  "numero_control": 1220100596}
```

The response status is **404 NOT FOUND** with a body:

```
{  "error": "No se encontró ningún estudiante con ese número de control."}
```

