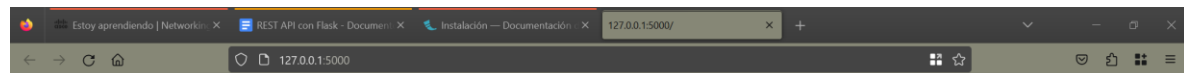


```
'''
    Autor: Salvador Reyes Morales
    Fecha: 31-07-2023
    Descripcion: Repaso
'''

from flask import Flask

app = Flask(__name__)

@app.get('/')
def welcome():
    return '<h1>Welcom comeback<h1>'
```



Welcom comeback

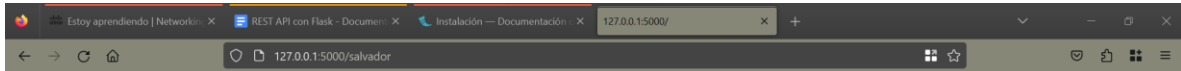
```
'''
    Autor: Salvador Reyes Morales
    Fecha: 31-07-2023
    Descripcion: Repaso
'''

from flask import Flask

app = Flask(__name__)

'''@app.get('/')
def welcome():
    return '<h1>Welcom comeback<h1>'
'''

@app.get('/<nombre>')
def greeting(nombre):
    return f'<h1>Hello {nombre}</h1>'
```



Hello salvador

A screenshot of a Visual Studio Code editor window. The editor shows a file named 'students.py' with the following code:

```
1 '''
2     Autor: Salvador Reyes Morales
3     Fecha: 31-07-2023
4     Descripción: Repaso
5 '''
6
7 from flask import Flask, jsonify
8
9 app = Flask(__name__)
10 students = [
11     {'numero_control':1220100603, 'nombre':'ANZO AVALOS MARIA CITLALLI'},
12     {'numero_control':1219100524, 'nombre':'ARREDONDO GONZALEZ DANIEL ENRIQUE'},
13     {'numero_control':1220100317, 'nombre':'DUARTE VELAZQUEZ DANIEL'},
14     {'numero_control':1220100629, 'nombre':'GUTIERREZ MARTINEZ VALERIA IVONNE'},
15     {'numero_control':1220100632, 'nombre':'LUNA CANTERO ANGEL IVAN'},
16     {'numero_control':1220100209, 'nombre':'MARTINEZ RAMIREZ GUADALUPE MONSERRAT'},
17     {'numero_control':1220100053, 'nombre':'REYES MORALES SALVADOR'},
18     {'numero_control':1220100597, 'nombre':'SALAZAR LEON MARIA GUADALUPE'},
19     {'numero_control':1220100596, 'nombre':'TADEO MARTINEZ ALEJANDRO'},
20     {'numero_control':1220100075, 'nombre':'TORRES GARCIA JOSE ROGELIO'},
21     {'numero_control':1220100595, 'nombre':'TRANQUEÑO HERNANDEZ OSCAR ARMANDO'}
22 ]
23
24 @app.get('/estudiantes')
25 def get_estudiantes():
26     return jsonify(students)
```

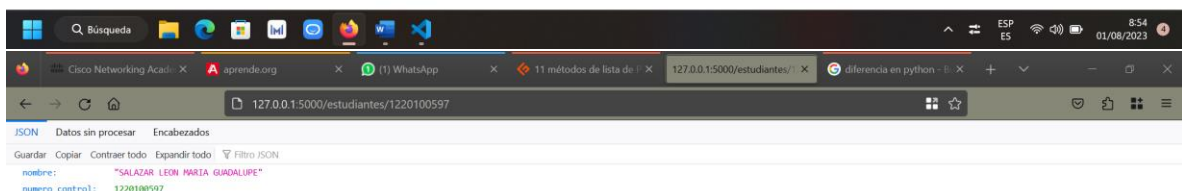
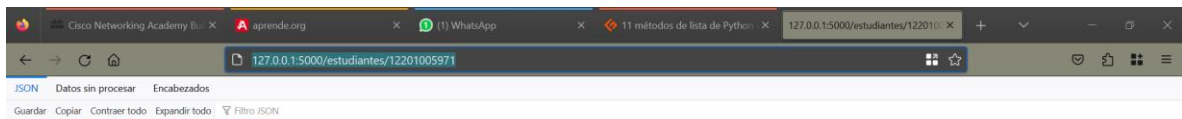
The terminal at the bottom shows the command 'flask --app students run --debug' and its output, including a warning about using a development server and the URL 'http://127.0.0.1:5000'.

JSON Datos sin procesar Encabezados

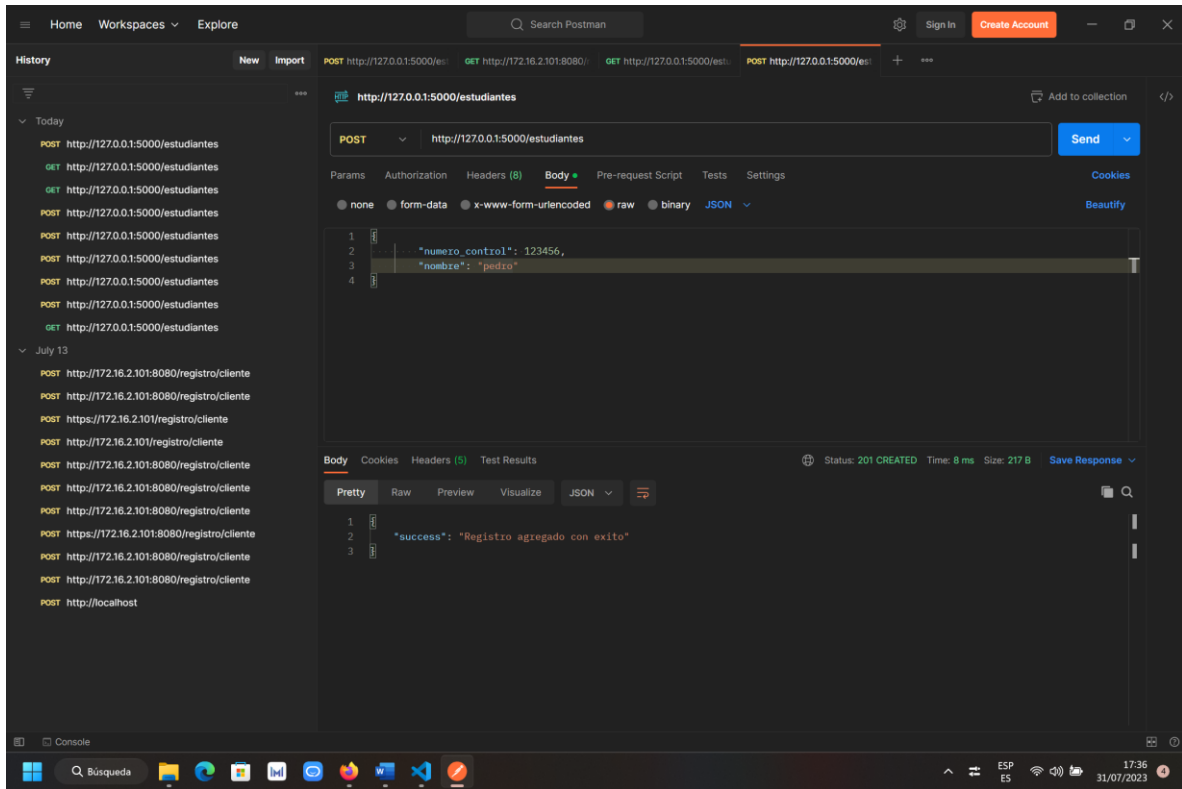
Guardar Copiar Contraer todo Expandir todo Filtro JSON

```
[{"nombre": "ANZO AVALOS MARIA CITALLI", "numero_control": 1220100603}, {"nombre": "ARREDONDO GONZALEZ DANIEL ENRIQUE", "numero_control": 1219100524}, {"nombre": "DUARTE VELAZQUEZ DANIEL", "numero_control": 1220100317}, {"nombre": "GUTIERREZ MARTINEZ VALERIA IVONNE", "numero_control": 1220100629}, {"nombre": "LUNA CANTERO ANGEL IVANI", "numero_control": 1220100632}, {"nombre": "MARTINEZ RAMIREZ GUADALUPE MONSERRAT", "numero_control": 1220100209}, {"nombre": "REYES MORALES SALVADOR", "numero_control": 1220100053}, {"nombre": "SALAZAR LEON MARIA GUADALUPE", "numero_control": 1220100597}, {"nombre": "TADCO MARTINEZ ALEJANDRO", "numero_control": 1220100596}, {"nombre": "TORRES GARCIA JOSE ROGELIO", "numero_control": 1220100075}, {"nombre": "TRANQUIERO HERNANDEZ OSCAR ARMANDO", "numero_control": 1220100595}]
```

```
@app.get('/estudiantes/<numero_control>')
def get_estudiantes(numero_control):
    for i in students:
        if i['numero_control'] == int(numero_control):
            return jsonify(i)
    return {'error': 'Este usuario no se encuentra registrado'}, 202
```



```
@app.post('/estudiantes')
def add_estudiantes():
    datos = request.get_json()
    students.append(datos)
    return {'success': 'Registro agregado con exito'}, 201
```



The screenshot shows the Postman application interface. The left sidebar displays a history of requests, with the current selection being a GET request to `http://127.0.0.1:5000/estudiantes`. The main panel shows the request details, including the URL, method (GET), and headers. The response body is displayed in the 'Body' tab, showing a JSON array of three student records. The status bar at the bottom indicates a 200 OK response with a time of 18 ms and a size of 111 KB.

```
GET http://127.0.0.1:5000/estudiantes
```

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value
Key	Value

Body Cookies Headers (5) Test Results Status: 200 OK Time: 18 ms Size: 111 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "nombre": "ANZO AVALOS MARIA CITLALLI",
3   "numero_control": 1228108603
4 },
5 {
6   "nombre": "ARREDONDO GONZALEZ DANIEL ENRIQUE",
7   "numero_control": 1219106524
8 },
9 {
10  "nombre": "DUARTE VELAZQUEZ DANIEL",
11  "numero_control": 1228108317
12 },
13 }
14
```

The screenshot shows the Postman application interface. The left sidebar displays a history of requests, with the current selection being a GET request to `http://127.0.0.1:5000/estudiantes/1220100053`. The main panel shows the request details, including the URL, method (GET), and headers. The response body is displayed in the 'Body' tab, showing a JSON object with one student record. The status bar at the bottom indicates a 200 OK response with a time of 20 ms and a size of 238 B.

```
GET http://127.0.0.1:5000/estudiantes/1220100053
```

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value
Key	Value

Body Cookies Headers (5) Test Results Status: 200 OK Time: 20 ms Size: 238 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "nombre": "REYES MORALES SALVADOR",
3   "numero_control": 1228100053
4 }
```

The screenshot displays the Postman application interface. On the left, the 'History' panel lists several API requests, including GET requests to `http://127.0.0.1:5000/estudiantes/12201000531` and `http://127.0.0.1:5000/estudiantes`, and POST requests to `http://127.0.0.1:5000/estudiantes` and `http://172.16.2.101:8080/registro/cliente`. The main panel shows a GET request to `http://127.0.0.1:5000/estudiantes/12201000531` with a status of 202 ACCEPTED. The response body is displayed in the 'Body' tab, showing a JSON object with an error message: `{\"error\": \"Este usuario no se encuentra registrado\"}`. The interface includes a top navigation bar with 'Home', 'Workspaces', and 'Explore' options, and a bottom status bar showing system information like 'ESP E5' and the date '01/08/2023'.

Home Workspaces Explore Search Postman Sign In Create Account

History New Import POST http://127.0.0.1:5000/est... GET http://172.16.2.101:8080/ GET http://127.0.0.1:5000/est... POST http://127.0.0.1:5000/est...

Today

- GET http://127.0.0.1:5000/estudiantes/12201000531
- GET http://127.0.0.1:5000/estudiantes/1220100053
- GET http://127.0.0.1:5000/estudiantes

Yesterday

- POST http://127.0.0.1:5000/estudiantes
- GET http://127.0.0.1:5000/estudiantes
- GET http://127.0.0.1:5000/estudiantes
- POST http://127.0.0.1:5000/estudiantes
- POST http://127.0.0.1:5000/estudiantes
- POST http://127.0.0.1:5000/estudiantes
- POST http://127.0.0.1:5000/estudiantes
- POST http://127.0.0.1:5000/estudiantes
- GET http://127.0.0.1:5000/estudiantes

July 13

- POST http://172.16.2.101:8080/registro/cliente
- POST http://172.16.2.101:8080/registro/cliente
- POST https://172.16.2.101/registro/cliente
- POST http://172.16.2.101/registro/cliente
- POST http://172.16.2.101:8080/registro/cliente
- POST http://172.16.2.101:8080/registro/cliente
- POST https://172.16.2.101:8080/registro/cliente
- POST http://172.16.2.101:8080/registro/cliente
- POST http://172.16.2.101:8080/registro/cliente
- POST http://localhost

GET http://127.0.0.1:5000/estudiantes

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (5) Test Results Status: 202 ACCEPTED Time: 10 ms Size: 228 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {  
2   \"error\": \"Este usuario no se encuentra registrado\"  
3 }
```