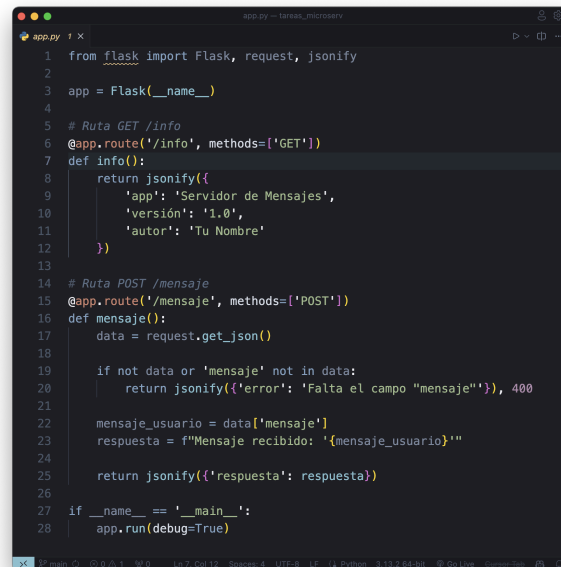


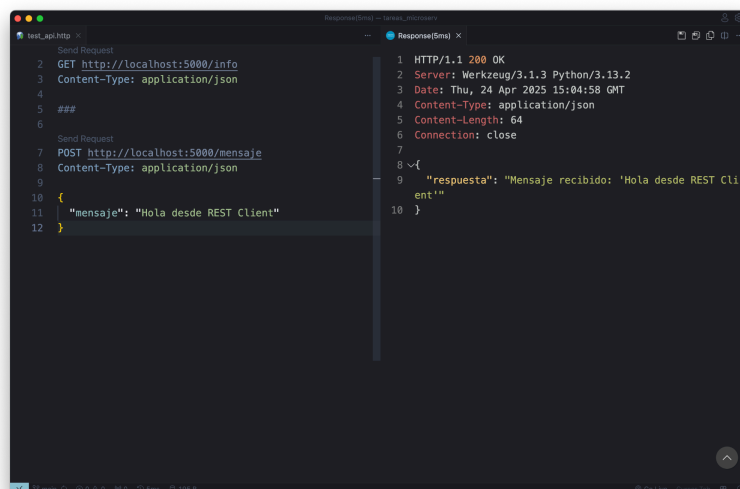
## API en python



```
1 from flask import Flask, request, jsonify
2
3 app = Flask(__name__)
4
5 # Ruta GET /info
6 @app.route('/info', methods=['GET'])
7 def info():
8     return jsonify({
9         'app': 'Servidor de Mensajes',
10        'versión': '1.0',
11        'autor': 'Tu Nombre'
12    })
13
14 # Ruta POST /mensaje
15 @app.route('/mensaje', methods=['POST'])
16 def mensaje():
17     data = request.get_json()
18
19     if not data or 'mensaje' not in data:
20         return jsonify({'error': 'Falta el campo "mensaje"'}) , 400
21
22     mensaje_usuario = data['mensaje']
23     respuesta = f"Mensaje recibido: '{mensaje_usuario}'"
24
25     return jsonify({'respuesta': respuesta})
26
27 if __name__ == '__main__':
28     app.run(debug=True)
```

**Explicación:** Aquí tenemos un api en flask python. Cuando se ejecuta el endpoint el api pide información json, si no hay información envía un mensaje de error. Pero si recibe información envía otra respuesta distinta. Ahora veremos una prueba en vivo.

## Prueba del endpoint



```
Send Request
1 GET http://localhost:5000/info
2 Content-Type: application/json
3
4 ###
5
6 Send Request
7 POST http://localhost:5000/mensaje
8 Content-Type: application/json
9
10 {
11   "mensaje": "Hola desde REST Client"
12 }

Response(Sms)
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.13.2
3 Date: Thu, 24 Apr 2025 15:04:58 GMT
4 Content-Type: application/json
5 Content-Length: 64
6 Connection: close
7
8 {
9   "respuesta": "Mensaje recibido: 'Hola desde REST Client'"
10 }
```

**Explicación:** Aquí podemos ver que se está haciendo una prueba con postman. Se le está enviando información al endpoint y el endpoint está contestando en formato json. En este caso recibió la información correctamente y envió el mensaje de success.