

A la izquierda está el servidor con las rutas, en el centro esta el rest para inserta los datos y en la derecha esta la prueba

The screenshot shows a VS Code editor with three main panels. The left panel displays the file explorer with a project named 'COMP2052' containing files like 'Mod-1-lec-2', 'Mod-1-lec-3', 'app.py', 'datos.json', 'insertrest', and 'README.md'. The middle panel shows the source code for 'app.py', which is a Flask application. It includes routes for '/info' (GET), '/crear_usuario' (POST), and '/usuarios' (GET). The '/crear_usuario' route checks if the request is JSON, validates the 'nombre' and 'correo' fields, and appends a new user to a list. The right panel shows the REST client interface with two requests. The first request is a POST to 'http://127.0.0.1:5000/crear_usuario' with a JSON body: {'nombre': 'Xavier', 'correo': 'xavier@gmail.com'}. The second request is a GET to 'http://127.0.0.1:5000/info'. The response for the second request is shown on the far right, indicating a 200 OK status and returning a JSON object with the application details.

```
from flask import Flask, request, jsonify

app = Flask(__name__)

usuarios = []

@app.route("/info", methods=["GET"])
def info():
    return jsonify({
        "app_name": "Sistema de Gestión",
        "version": "1.0",
        "description": "API para gestionar usuarios"
    })

@app.route("/crear_usuario", methods=["POST"])
def crear_usuario():
    print(request.data)
    if not request.is_json:
        return jsonify({"error": "Content-Type debe ser application/json"}), 415
    data = request.json
    print(data)
    nombre = data.get("nombre")
    correo = data.get("correo")
    if not nombre or not correo:
        return jsonify({"error": "El nombre y el correo son obligatorios"}), 400
    usuario = {"nombre": nombre, "correo": correo}
    usuarios.append(usuario)
    return jsonify(usuario), 201

@app.route("/usuarios", methods=["GET"])
def get_usuarios():
    return jsonify(usuarios)

if __name__ == "__main__":
    app.run(debug=True)
```

```
POST http://127.0.0.1:5000/crear_usuario
Content-Type: application/json

{
  "nombre": "Xavier",
  "correo": "xavier@gmail.com"
}
```

```
GET http://127.0.0.1:5000/info
```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.1.3 Python/3.13.2
Date: Wed, 23 Apr 2025 19:34:23 GMT
Content-Type: application/json
Content-Length: 267
Connection: close

{
  "app_name": "Sistema de Gestión",
  "version": "1.0",
  "description": "API para gestionar usuarios"
}
```