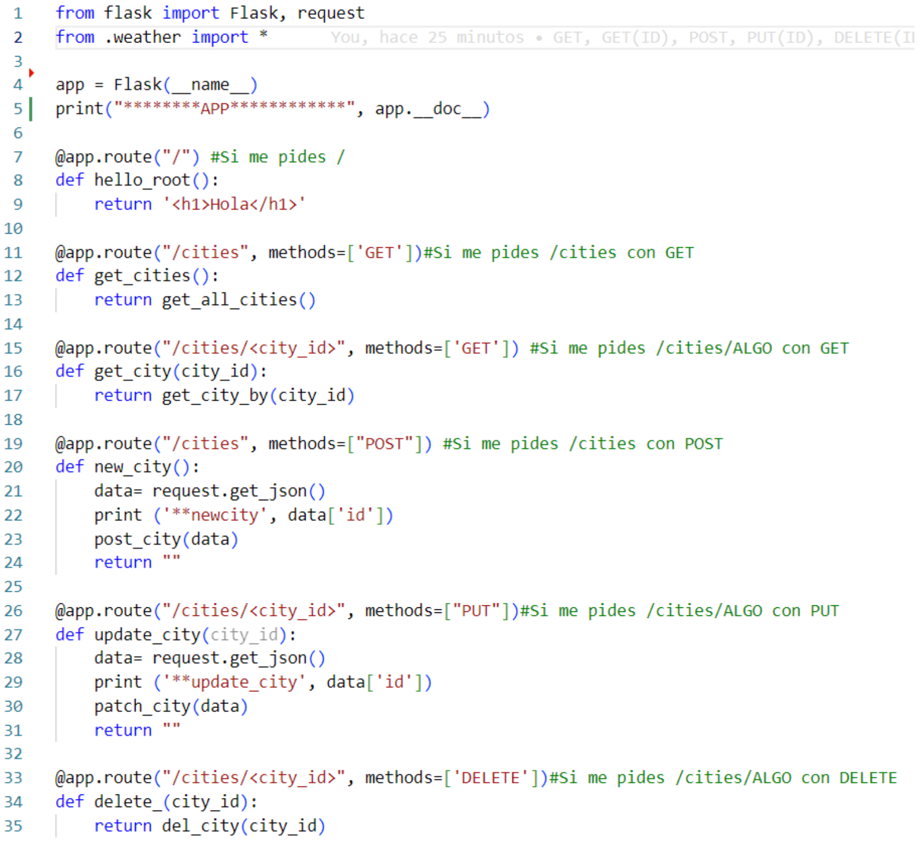
SERVIDOR BACKEND EN PYTHON

COMO FUNCIONA

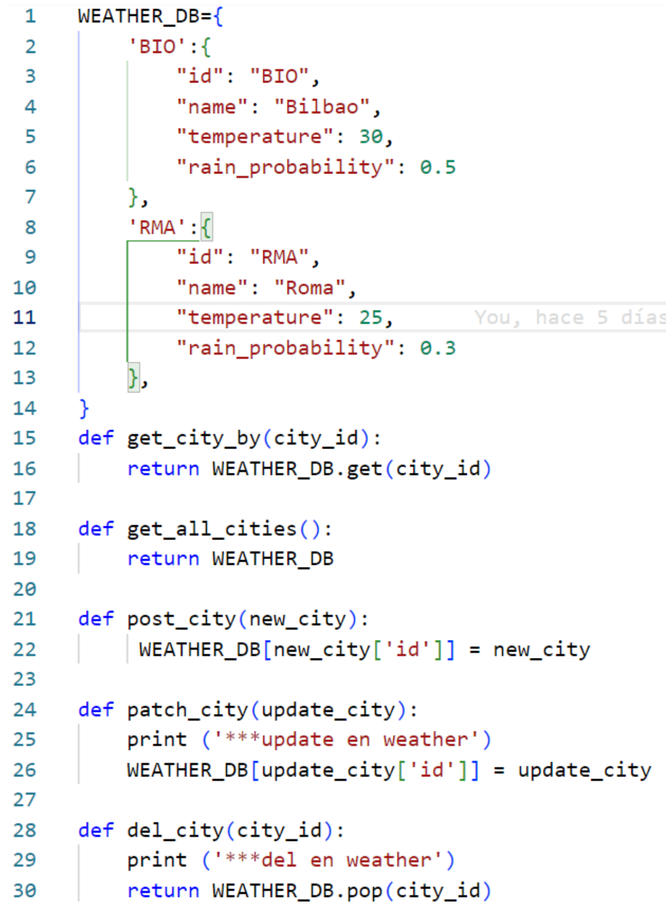
- Importa app de src/webserver.py
- App es una instancia de Flask
- Lo arranca con Debugger

- En webserver.py hace una lista de direcciones y métodos. Si llega un request a la dirección y con el método, ejecuta la función, que llama a la weather.py
- En weather.py hace los cambios en la “BBDD”

webserver.py & weather.py



```python
from flask import Flask, request
from .weather import *     You, hace 25 minutos • GET, GET(ID), POST, PUT(ID), DELETE(II

app = Flask(__name__)
print("********APP************", app.__doc__)

@app.route("/") #Si me pides /
def hello_root():
    return '<h1>Hola</h1>'

@app.route("/cities", methods=['GET'])#Si me pides /cities con GET
def get_cities():
    return get_all_cities()

@app.route("/cities/<city_id>", methods=['GET']) #Si me pides /cities/ALGO con GET
def get_city(city_id):
    return get_city_by(city_id)

@app.route("/cities", methods=["POST"]) #Si me pides /cities con POST
def new_city():
    data= request.get_json()
    print ('**newcity', data['id'])
    post_city(data)
    return ""

@app.route("/cities/<city_id>", methods=["PUT"])#Si me pides /cities/ALGO con PUT
def update_city(city_id):
    data= request.get_json()
    print ('**update_city', data['id'])
    patch_city(data)
    return ""

@app.route("/cities/<city_id>", methods=['DELETE'])#Si me pides /cities/ALGO con DELETE
def delete_(city_id):
    return del_city(city_id)
```



```python
WEATHER_DB={
    'BIO':{
        "id": "BIO",
        "name": "Bilbao",
        "temperature": 30,
        "rain_probability": 0.5
    },
    'RMA':{
        "id": "RMA",
        "name": "Roma",
        "temperature": 25,        You, hace 5 días
        "rain_probability": 0.3
    },
}
def get_city_by(city_id):
    return WEATHER_DB.get(city_id)

def get_all_cities():
    return WEATHER_DB

def post_city(new_city):
     WEATHER_DB[new_city['id']] = new_city

def patch_city(update_city):
    print ('***update en weather')
    WEATHER_DB[update_city['id']] = update_city

def del_city(city_id):
    print ('***del en weather')
    return WEATHER_DB.pop(city_id)
```