



MICROSERVICIOS

Emanuel Alejandro Gutierrez Romero



NOVEMBER 13, 2023

COMPUTACION TOLERANTE A FALLAS

Prof. Michel Emanuel López Franco

Realmente este es un ejemplo bastante sencillo de un microservicio, comenzamos por importar nuestra librería Flask, instalándola primero claramente.

```
from flask import Flask
```

Después de ello definimos el nombre del módulo usando `__name__`, ayudando a la librería Flask a determinar el punto del servidor, rutas y otros detalles de la aplicación.

```
app = Flask(__name__)
```

Después definimos una ruta en específico y como lo vamos a buscar dentro del navegador, creamos una función que se ejecutará dentro del url y devolverá un mensaje diciendo el clima de hoy dentro de esa página.

```
@app.route('/getWeatherForecast')
def getWeatherOnline():
    return 'Hoy será un día soleado dentro del microservicio!'
```


Después hacemos una verificación para ejecutar el programa y hacemos el llamado a que Flask ejecute las solicitudes entrantes.

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

Ahora vemos como se ve.

```
PS C:\Users\ALDO\Desktop\Microservicio> & C:/Users/ALDO/AppData/Local/Programs/Python/Python310/python.exe c:/Users/ALDO/Desktop/Microservicio/app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 105-530-405
127.0.0.1 - - [13/Nov/2023 19:07:57] "GET /getWeatherForecast HTTP/1.1" 200 -
127.0.0.1 - - [13/Nov/2023 19:07:57] "GET /favicon.ico HTTP/1.1" 404 -
```

Introducimos la dirección que nos asignó y la ruta al final.

 127.0.0.1:5000/getWeatherForecast

Como podemos ver el programa funciona correctamente.

