

## Act. 2.2 Crear Microservicio Parte 02

Repositorio:

```
https://github.com/Gilberto-Guzman/Creacion-De-Microservicios/tree/master/Microservicios02
```

Comandos utilizados:

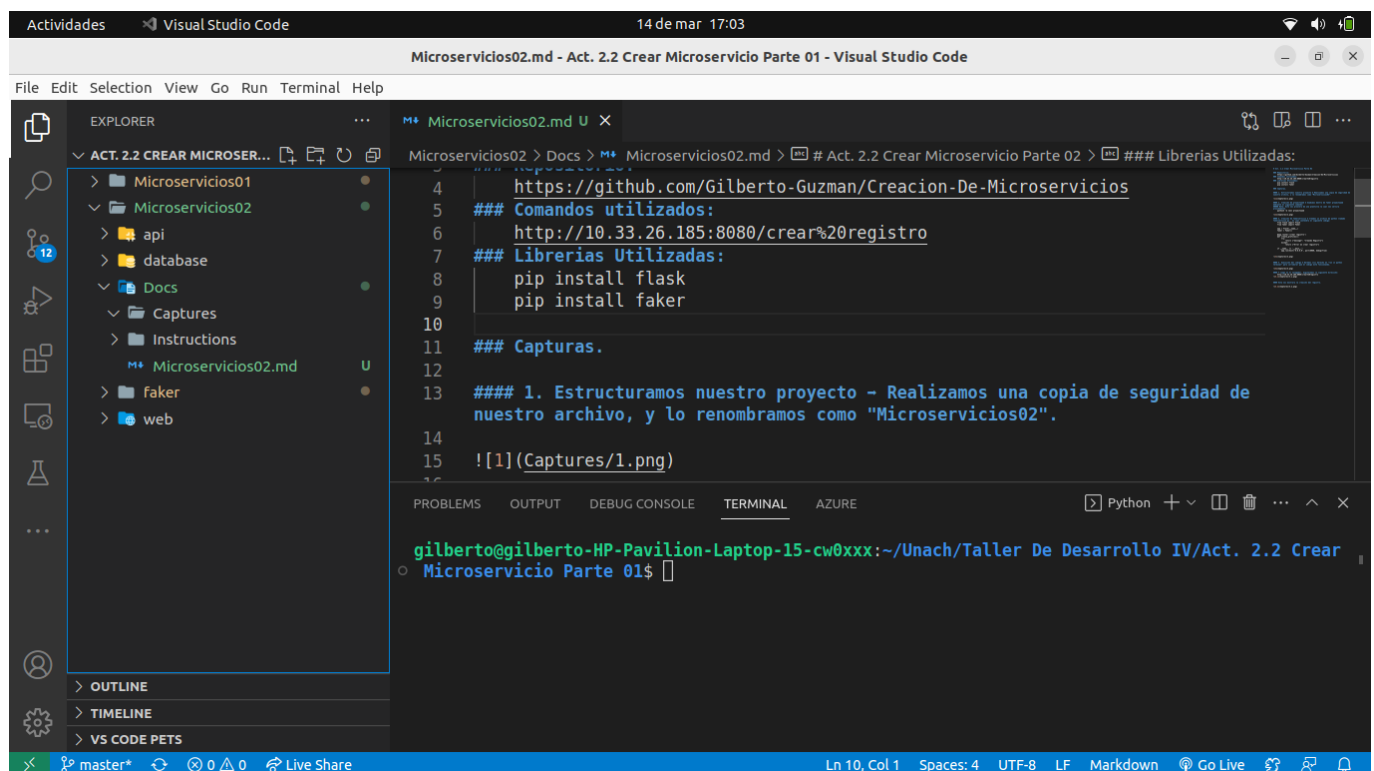
```
http://10.33.26.185:8080/crear%20registro
```

Librerías Utilizadas:

```
pip install flask  
pip install faker
```

Capturas.

**1. Estructuramos nuestro proyecto → Realizamos una copia de seguridad de nuestro archivo, y lo renombramos como "Microservicios02".**



**2. Actualizamos el contenido de fakerservicio.py → Modificamos el código de la siguiente manera:**

```
from flask import Flask, jsonify
from faker import Faker
import itertools

app = Flask(__name__)
faker = Faker()

def random_data():
    profiles = [dict(itertools.islice(faker.profile().items(), 6))
                 for data in range(13)]
    return profiles

@app.route('/crear registro')
def create_profiles():
    try:
        data_profiles = random_data()
        return jsonify(data_profiles)
        # print(data_profiles)
        # return {"message": "Creando Registro"}
    except:
        return {"Error en crear registro"}

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080, debug=True)
```

**Nota:** Se añadió una función llamada "random\_data", la cual generará los datos aleatorios, y se actualizó el funcionamiento de "create\_profiles" para el muestreo de los datos en formato json.

```

4
5 app = Flask(__name__)
6 faker = Faker()
7
8
9 def random_data():
10     profiles = [dict(itertools.islice(faker.profile().items(), 6))
11                 for data in range(13)]
12     return profiles
13
14
15 @app.route('/crear registro')
16 def create_profiles():
17     try:
18         data_profiles = random_data()
19         return jsonify(data_profiles)
20         # print(data_profiles)
21         # return {"message": "Creando Registro"}
22     except:
23         return {"Error en crear registro"}
24
25
26 if __name__ == '__main__':
27     app.run(host='0.0.0.0', port=8080, debug=True)
28

```

**3. Ejecución del código → Daremos clic derecho en "run in python terminal" para corroborar que el código este funcionando.**

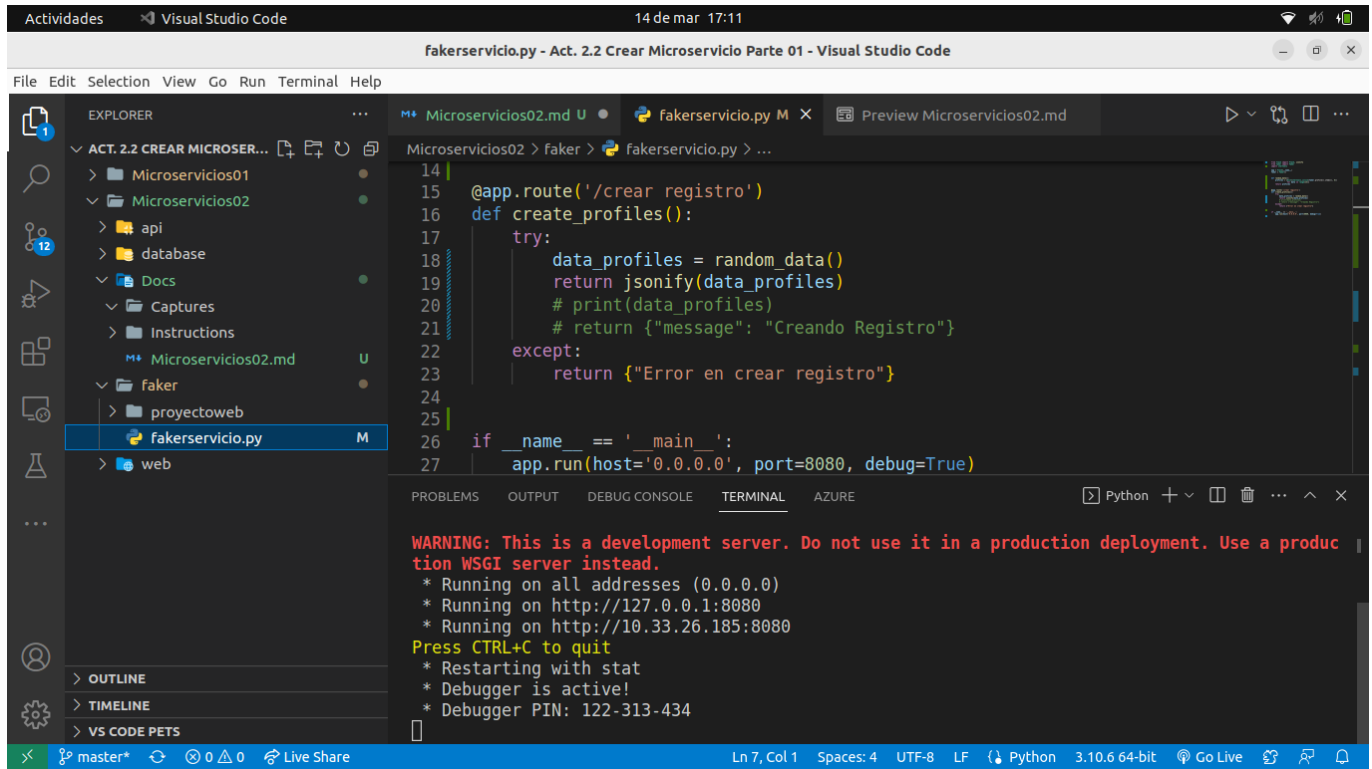
```

44 app.run(host='0.0.0.0', port=8080, debug=True)
45
46 #### Nota: Se añadió una función llamada "random_data", la cual generara los
47 datos aleatorios, y se actualizo el funcionamiento de "create_profiles" para
48 el muestreo de los datos en formato json.
49
50 ![2](Captures/2.png)

```

**Y luego en el navegador, digitaremos la siguiente dirección:**

`http://10.33.26.185:8080/crear%20registro`



```
14 |
15 | @app.route('/crear_registro')
16 | def create_profiles():
17 |     try:
18 |         data_profiles = random_data()
19 |         return jsonify(data_profiles)
20 |         # print(data_profiles)
21 |         # return {"message": "Creando Registro"}
22 |     except:
23 |         return {"Error en crear registro"}
24 |
25 |
26 | if __name__ == '__main__':
27 |     app.run(host='0.0.0.0', port=8080, debug=True)
```

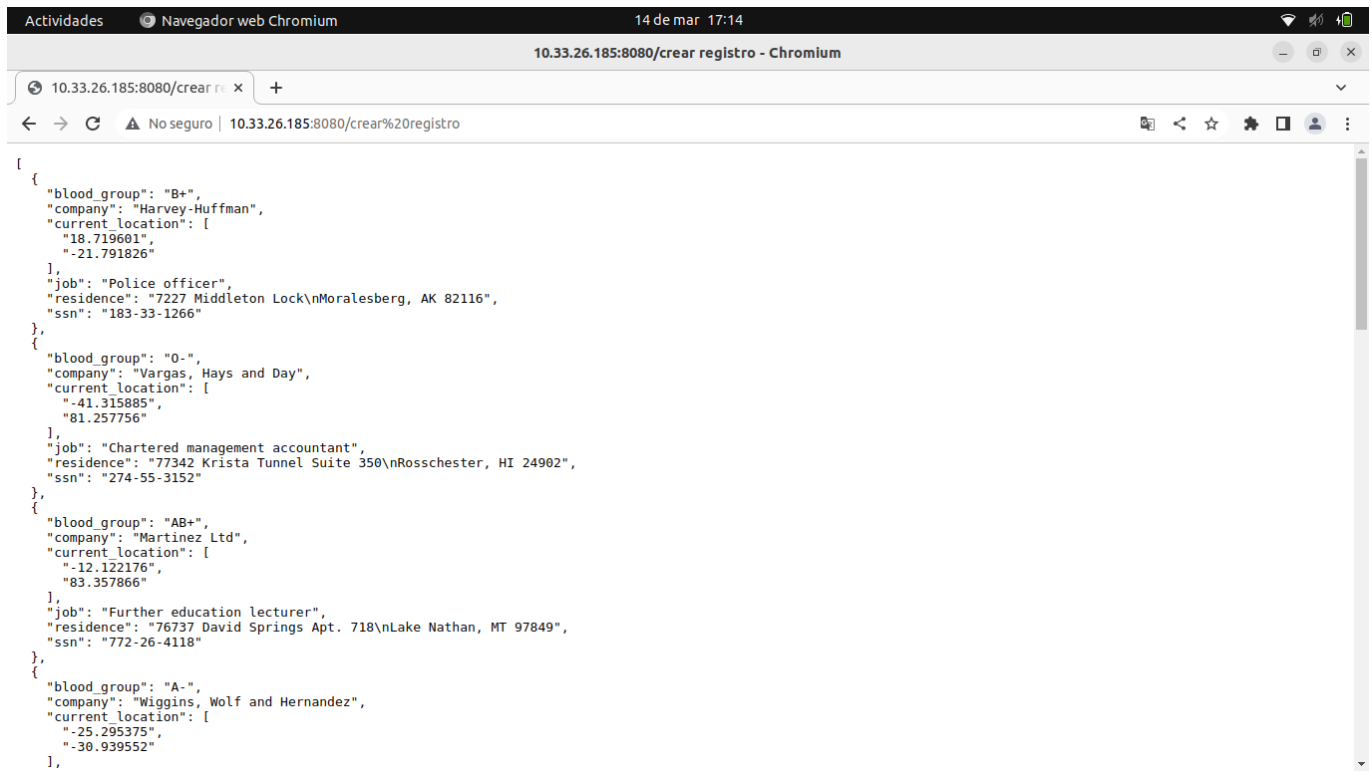
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

- \* Running on all addresses (0.0.0.0)
- \* Running on http://127.0.0.1:8080
- \* Running on http://10.33.26.185:8080

Press CTRL+C to quit

- \* Restarting with stat
- \* Debugger is active!
- \* Debugger PIN: 122-313-434

Esta nos mostrara la creación del registro.



```
[
  {
    "blood_group": "B+",
    "company": "Harvey-Huffman",
    "current_location": [
      "18.719601",
      "-21.791826"
    ],
    "job": "Police officer",
    "residence": "7227 Middleton Lock\\nMoralesberg, AK 82116",
    "ssn": "183-33-1266"
  },
  {
    "blood_group": "O-",
    "company": "Vargas, Hays and Day",
    "current_location": [
      "-41.315885",
      "81.257756"
    ],
    "job": "Chartered management accountant",
    "residence": "77342 Krista Tunnel Suite 350\\nRosschester, HI 24902",
    "ssn": "274-55-3152"
  },
  {
    "blood_group": "AB+",
    "company": "Martinez Ltd",
    "current_location": [
      "-12.122176",
      "83.357866"
    ],
    "job": "Further education lecturer",
    "residence": "76737 David Springs Apt. 718\\nLake Nathan, MT 97849",
    "ssn": "772-26-4118"
  },
  {
    "blood_group": "A-",
    "company": "Wiggins, Wolf and Hernandez",
    "current_location": [
      "-25.295375",
      "-30.939552"
    ],
    "job": " ",
    "residence": " ",
    "ssn": " "
  }
]
```