Práctica
Microservicios con REST usando Flask + MySQL

1. Objetivo

Construir una aplicación web basada en microservicios y una API Rest en Python Flask con persistencia de datos usando una base de datos MySQL

2. Herramientas a Utilizar

- Maquina virtual de Ubuntu en Vagrant
- Python Flask
- Código en repositorio git

3. Desarrollo de la Practica

Aprovisione la Maquina Virtual

Clone el repositorio en:

https://github.com/omondragon/microwebAppBase

Ingrese al directorio del proyecto clonado

```
cd microwebAppBase/
```

Levante y aprovisione la maquina

```
vagrant up
```

Iniciar sesión SSH en el servidor

```
vagrant ssh servidorWeb
```

Ejecutar el microservicio del frontend

```
cd /home/vagrant/frontend
export FLASK_APP=run.py
/usr/local/bin/flask run --host=0.0.0.0 --port 5001
```

Ejecutar el microservicio de gestión de usuarios

```
cd /home/vagrant/microUsers
export FLASK_APP=run.py
/usr/local/bin/flask run --host=0.0.0.0 --port 5002
```

Probar

En un navegador abra http://192.168.80.3:5001

1. Ejercicio

1. Ingrese a la base de datos en mysql y visualice la tabla creada y su contenido

```
vagrant@servidorWeb:~$ systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-04-13 01:39:03 UTC; 39s ago
   Main PID: 974 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 2191)
     Memory: 422.3M
        CPU: 638ms
     CGroup: /system.slice/mysql.service
             └─974 /usr/sbin/mysqld

Apr 13 01:39:02 servidorWeb systemd[1]: Starting MySQL Community Server...
Apr 13 01:39:03 servidorWeb systemd[1]: Started MySQL Community Server.
vagrant@servidorWeb:~$ mysql -u root -proot
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use myflaskapp;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+----------------------+
| Tables_in_myflaskapp |
+----------------------+
| users                |
+----------------------+
1 row in set (0.00 sec)

mysql> describe users;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int          | NO   | PRI | NULL    | auto_increment |
| name     | varchar(255) | YES  |     | NULL    |                |
| email    | varchar(255) | YES  |     | NULL    |                |
```

```
| username | varchar(255) | YES  |     | NULL    |                |
| password | varchar(255) | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)

mysql> select * from users;
+----+-------+-----------------+----------+----------+
| id | name  | email           | username | password |
+----+-------+-----------------+----------+----------+
|  1 | juan  | juan@gmail.com  | juan     | 123      |
|  2 | maria | maria@gmail.com | maria    | 456      |
+----+-------+-----------------+----------+----------+
2 rows in set (0.00 sec)
```

2. Asegúrese de entender el código de la aplicación web desplegada compuesta por los dos microservicios creados y pruebe su funcionalidad
3. Cree un nuevo microservicio que se ejecute en el puerto 5003 para la gestión de productos funcionalidad necesaria para gestionar productos
    a. Cree la tabla productos en la base de datos
    b. Cree la API Rest Correspondiente
    c. Cree los templates necesarios en el microservicio de fronted para su gestión
4. Realice el monitoreo de los microservicios creados usando el mecanismo de service Discovery de Consul
    a. Instale flask-consulate

        ```
        pip3 install flask-consulate
        ```

    b. Siga el README disponible en https://github.com/vsudilov/flask-consulate para registrar el servicio y crear la función de healthcheck.

2. Bibliografía

- Postman. https://www.postman.com/
- Repositorio github para Flask Consulate. https://github.com/vsudilov/flask-consulate
- Instalar flask-consulate. https://pypi.org/project/flask-consulate/