

Manual para ejecutarlo en Docker

Primero se clona a tu directorio la carpeta de Docker Python con un git clone o se descarga en formato zip.

```
$ git clone https://github.com/docker/python-docker
```

Después se ejecutan los siguientes junto con los requerimientos comandos, aparecerán el Dockerfile y el compose, en mi caso fue sin Python3 solo Python.

```
(.venv) $ python3 -m pip install -r requirements.txt  
(.venv) $ python3 -m flask run
```

Solo quedaría iniciar Docker con el comando init y contestar las siguientes preguntas.

```
$ docker init  
Welcome to the Docker Init CLI!  
  
This utility will walk you through creating the following files with sensible defaults:  
- .dockerignore  
- Dockerfile  
- compose.yaml  
  
Let's get started!  
  
? What application platform does your project use? Python  
? What version of Python do you want to use? 3.11.4  
? What port do you want your app to listen on? 5000  
? What is the command to run your app? python3 -m flask run --host=0.0.0.0
```

Después de instalar todas las herramientas de Docker, toca hacer la conexión con la base de datos, solo es necesario cambiar el localhost por la ip que tenga el pc.

```
import mysql.connector  
  
app = Flask(__name__)  
  
# Configuración de la conexión a la base de datos  
conexion = {  
    'host': '192.168.1.6',  
    'user': 'ang',  
    'password': 'Luca',  
    'database': 'School_db'  
}  
  
connection = mysql.connector.connect(**conexion)
```

Se configura la última línea en el archivo Dockerfile que es con el comando que se correrá nuestro docker.

```
EXPOSE 5000

# Run the application.
CMD python -m flask --app instrumento run --host=0.0.0.0
```

Se instalan las siguientes dependencias, la primera para encriptamiento y la segunda para la conexión a la base de datos.

```
# into this layer.
RUN --mount=type=cache,target=/root/.cache/pip \
    --mount=type=bind,source=requirements.txt,target=requirements.txt \
    python -m pip install -r requirements.txt
RUN pip install bcrypt
RUN pip install Flask mysql-connector-python

# Switch to the non-privileged user to run the application.
USER appuser
```

Después construir una imagen en Docker para poder ejecutarla después de la siguiente manera.

```
PS C:\Users\Angel\OneDrive\Documentos\Universidad\4to ING\Automatiza\Ejercicios\python-docker-main\python-docker
-main> docker build --tag instru .
```

Por ultimo se corre el Docker con el siguiente comando, recordar cambiar el nombre por la imagen que anteriormente se construyó.

```
PS C:\Users\Angel\OneDrive\Documentos\Universidad\4to ING\Automatiza\Ejercicios\python-docker-main\python-docker
-main> docker run -d -p 5000:5000 instru
d1eb411fe64311ab8471ec606f6c2da0bbf49cb61fe6bb2e09eb79d07f6126a0
PS C:\Users\Angel\OneDrive\Documentos\Universidad\4to ING\Automatiza\Ejercicios\python-docker-main\python-docker
```

Nota:

Cada que se hacen cambios en el microservicio o en el Dockerfile se debe reconstruir la imagen para que se guarden los cambios realizados.

Pruebas con Postman.

