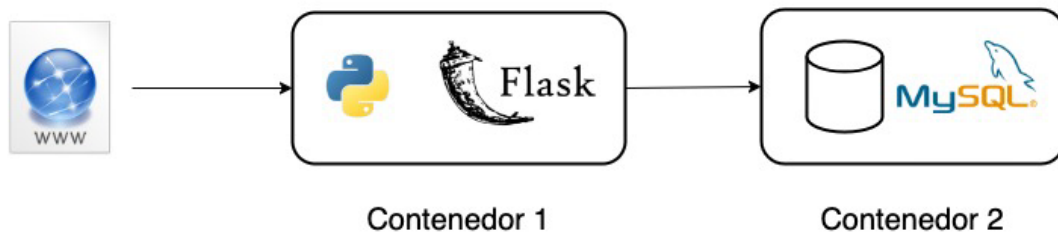


Problema Lab

Backend

APLICACIÓN WEB PYTHON Y MYSQL CON DOCKER

Disponemos de una aplicación web desarrollada en Python con Flask y MySQL como motor de base de datos.



Necesitamos *Dockerizar* nuestra aplicación para mejorar los flujos de trabajo del equipo. ¿Te apuntas a este reto?

Objetivos de este ejercicio

Con esta práctica se pretende que empieces a introducir Docker en tu flujo de trabajo y cojas soltura en el uso del sistema.

Descripción de la actividad

Necesitamos montar una red de contenedores Docker, la llamaremos "todo_lan", donde irán alojados nuestros servicios, en este caso, un Python (con el framework Flask) y una base de datos MySQL. A continuación, te compartimos algunas claves que te pueden ayudar para resolver cada bloque de tareas.

1. Flask

Contenedor montado a partir de una imagen Python (no te preocupes si no has trabajado nunca con Python, este ejercicio va sobre Docker).

- Recuerda que en la página oficial de cada imagen en Docker Hub tienes toda la información sobre ella y algunos ejemplos de uso.
- En el repositorio de GitHub de la asignatura dispones del código de la aplicación Python, así como de un *requirements.txt* que serán los paquetes Python para instalar con “pip install”.
- <https://github.com/Qualentum/backend> (puedes localizarlo en la carpeta ‘Lab1/El problema’).

2. MySQL

Contenedor montado a partir de una imagen MySQL.

- Deberás permitir que el servicio Flask se conecte a este contenedor para poder montar la base de datos, sus tablas y poder empezar a trabajar con la data. No te preocupes, una vez ambos servicios sean capaces de comunicarse, la propia aplicación Flask se encargará de crear las tablas y demás en la base de datos.

Formato de entrega

Tu solución al ejercicio debe contemplar dos ficheros:

- El fichero **dockerfile** que te permita generar tu propia imagen personalizada de Python, con las variables de entorno para poder conectarte al servicio de base de datos, el volumen de datos de la aplicación y Flask corriendo.
- El fichero **docker-compose.yml** que permita montar la app con la red `todo_lan` y ambos contenedores corriendo (uno con la imagen personalizada creada anteriormente y otro con un servicio MySQL).

Por lo tanto, para realizar la entrega tendrás que crear un .zip con ambos ficheros. Si precisas añadir un *readme.txt* en ese .zip para indicar algo o lo que consideres, adelante con ello.

Criterios de corrección

Cuando visualices el clip con la solución del ejercicio, verás que lo importante en esta práctica son los dos ficheros generados, `dockerfile` y `dockercompose.yml`, que deberán permitir montar la red de servicios para que nuestra aplicación web funcione. También resultará clave en tu autoevaluación que ambos procesos, el planteamiento de la solución y cómo la has implementado, sean óptimos.

Y si no has conseguido que te funcione al 100%, no te agobies, envíanos lo que tienes, el ejercicio hasta donde hayas podido trabajar, que seguro ibas por el buen camino.

Anexo

¿Qué material necesitas?

App Python (consulta el repositorio Github de la asignatura, carpeta Lab 1).

Tips

En la sección de Action Learning del vídeo-lab, cuando montábamos nuestro WordPress, el profesor te mostró algunas instrucciones de Python que te pueden resultar muy útiles.

¿Quieres más?

Si te has quedado con gana de más, ¡¡adelante!! Se trata de aprender y trabajar con Docker; si quieres añadir algún servicio adicional o montar tu propio entorno con el stack que uses, siéntete libre.



[Qualentum.com](https://www.qualentum.com)