Práctica 0: Entorno de desarrollo python con docker compose

Sergio Alonso (zerjioi@ugr.es) y José María Guirao (jmguirao@ugr.es)

Durante las prácticas de la asignatura usaremos Python para el back-end, un lenguaje de alto nivel sencillo, potente, libre, "fácil de aprender", interpretado, multiplataforma y el más usado ahora en el servidor.

En esta primera práctica proponemos un ambiente de desarrollo para python con Docker y docker compose.

Para aquellos que no tengan conocimientos sobre este lenguaje, existen numerosos manuales en Internet que permiten acercarse a la programación Python de manera sencilla y amena como por ejemplo: A Byte of Python o el manual de Google sobre Python, o algún video tutorial.

Usaremos docker y docker compose para el entorno de desarrollo para:

- Aislar nuestro código del resto del ordenador, sin que interfiera lo que pueda instalarse antes o después.
- Conseguir un entorno **idéntico** para todos los participantes del curso, nos ahorramos el problema "pues en mi ordenador funciona" .
- Tener nuestra aplicación lista para subirla a la nube.

Un primer paso será instalar el servidor de Docker. En este guión no se darán instrucciones concretas para la instalación puesto que depende del sistema operativo anfitrión donde desarrollaremos nuestras prácticas. Consulta los enlaces anteriores para instalarlos en tu sistema operativo. Para linux seguir también Linux post-installation steps for Docker Engine.

También puedes consultar el siguiente tutorial: What is Docker and How to Use it With Python.

Una vez instalados Docker y Docker compose podemos construir un primer archivo `docker-compose.yml` para ejecutar los ejercicios de este guión. En este archivo especificamos que se monte un directorio `ejercicios` que debe encontrarse en el mismo directorio que `docker-compose.yml` (yml es un formato para marcado ligero y archivos de configuración).

```
# docker-compose.yml
services:
  app:
    image: python:3.11-alpine
    volumes:
      - ./tests:/tests
    working_dir: /tests
```

y dentro de una carpeta **tests**

```
# tests/Hola_mundo.py
print("Hola mundo!")
```

Si todo va bien podemos ejecutar `tests/Hola_mundo.py` dentro del contenedor `app` con la siguiente orden:

```
> docker compose run app python Hola_mundo.py
```

Docker compose cheatsheet