

Demo 2

Usando Django desde un contenedor

Pre-requisitos

- Instalar docker. [Tutorial](#)
- Clonar el siguiente [proyecto](#)

Descripción del ejemplo

La idea es ejecutar un "Hola mundo" usando python desde un contenedor docker pero en este caso utilizando un framework de desarrollo web como es [Django](#). De nuevo, no tiene mayor complejidad pero sirve para tener una idea básica de la dinámica de trabajo cuando estamos en un escenario como este.

Generando una imagen

Como se menciona en la presentación, hay ocasiones en las que conviene utilizar una imagen propia para que contenga todo lo necesario para un proyecto. O al menor establecer un único método para personalizarla.

Esto se puede hacer a través de un archivo **Dockerfile**. Vamos a crearlo con el siguiente contenido:

```
FROM python:3

WORKDIR /src

COPY requirements.txt .

RUN pip3 install -r requirements.txt
```

Qué es eso?

- `FROM python:3` sirve para indicar la imagen base sobre la que se va a trabajar.
- `WORKDIR /src` indica que esa ubicación será en la que se van a ejecutar los siguientes comandos del archivo.
- `COPY requirements.txt .` hace la copia del archivo definido de la ubicación actual a la imagen. En este caso va a ser en la ubicación definida en el paso previo.
- `RUN pip3 install -r requirements.txt` es la orden para ejecutar la instalación de las dependencias del proyecto en la imagen.

En este momento, después de generar el archivo, se podría generar una imagen con sus características, para eso se tiene que ejecutar el siguiente comando (**estando en la misma ubicación** del Dockerfile):

```
docker build -t django .
```

Descripción del comando:

- `build` es la orden para que docker genere una nueva imagen.
- `-t django` es el parámetro a utilizar para definir el nombre de la imagen a crear.
- `.` es la indicación de la ubicación del Dockerfile que genera la imagen.

Con la imagen generada se puede pasar a ejecutarla, pero en este caso se deberá hacer con el siguiente comando:

```
docker run --rm -it -v $(pwd):/src -p 8000:8000 django python manage.py  
runserver 0.0.0.0:8000
```

Analizando sus partes:

- `-p 8000:8000` sirve para indicar que lo que el contenedor presente en el puerto de la derecha, se va a poder obtener a través del puerto de la izquierda en el host. En este caso, el puerto 8000 del host se vincula al del contenedor para acceder a la instancia de Django instalada en el mismo.
- `django` es el nombre de la imagen con la que se genera el contenedor, el mismo utilizado en el paso previo.
- `python manage.py runserver 0.0.0.0:8000` es la orden para que se pase a ejecutar el servidor de desarrollo. Se especifica de esa manera la IP para poder hacer la visualización desde el host.

De esta manera, accediendo a **localhost:8000**, si todo salió bien se debería ver una página con un *Hola mundo desde Django!*.