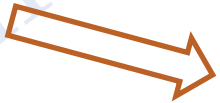
Creación de imágenes docker

El archivo dockerfile

- **Contiene la información necesaria para que docker pueda construir una imagen**
- **Se compone de una serie de comandos con sus correspondientes valores:**

estructura dockerfile

```
COMANDO1  valor
COMANDO2  valor
…
```



ejemplo dockerfile

```
FROM openjdk:8-jdk-alpine
ADD cliente.jar client.jar
EXPOSE 8080
```

Principales comandos

- **FROM. Indica la imagen base de la que se parte**
- **ADD. Archivos que se van a incluir:**

```
ADD ruta_origen  ruta_destino
```

⟹

```
ADD servicio.jar  miservicio.jar
```

- **EXPOSE. Número de puerto por el que el contenedor será accesible a otros contenedores:**

```
EXPOSE 8080
```

- **ENTRYPOINT. Comando que debe ser ejecutado al lanzar el contenedor:**

```
ENTRYPOINT ["ejecutable", "param1", "param2",..]
ENTRYPOINT ["java","-jar","/miservicio.jar"]
```

Ejemplo dockerfile

Se parte de una imagen con JDK 11 instalado

```
FROM openjdk:11
ADD crudlibros.jar crudlibros.jar
EXPOSE 8080
ENTRYPOINT ["java","-jar","/crudlibros.jar"]
```

Construcción de la imagen

- **Para construir la imagen, utilizamos el comando *build* con el siguiente formato:**

```
>docker build -t nombre_imagen ruta_dockerfile
```

- **Si escribimos el comando en la misma carpeta donde está dockerfile:**

```
>docker build -t imagen1 .
```

- **Una vez construida, aparece en nuestro listado de imágenes con *docker images***

Creación y ejecución de contenedor

- **Para crea contenedores a partir de una imagen y ponerlos en ejecución utilizamos el comando *run*.**

- **En el caso de un servicio, deberá mapearse el puerto del contenedor a un puerto de la máquina física:**

```
>docker run -p puerto_equipo:puerto_contenedor nombre_imagen
```

- **Por ejemplo, para crear un contenedor mapeado al puerto 9000 de la máquina física:**

```
> docker run -p 9000:8000 imagen1
```