

### T3.3 – Información de Imágenes

Para ello tenemos **dos fuentes principales**:

- La **página de la imagen en DockerHub** recoge información relativa a:
  - Una descripción de la aplicación o funcionamiento de la imagen.
  - Una lista de versiones TAGs disponibles.
  - Variables de entorno interesantes.
  - Cómo ejecutar la imagen.
- La salida de las órdenes **docker image inspect** / **docker inspect** que nos da ya una información más detallada sobre las características, con todos los metadatos de la misma.

Veamos un ejemplo de la misma:

```
# Dos formas de obtener información de la imagen mysql:8.0.22
> docker image inspect mysql:8.0.22
> docker inspect mysql:8.0.22
```

Esta imagen es una imagen parcial, porque se nos muestra mucha información, está en formato JSON (JavaScript Object Notation) y nos da datos sobre aspectos como:

- El **id** y el **checksum** de la imagen.
- Los **puertos** abiertos.
- La **arquitectura** y el **sistema operativo** de la imagen.
- El **tamaño de la imagen**.
- Los **volúmenes**.
- El **ENTRYPOINT** que es lo que se ejecuta al hacer docker run.
- Las **capas**.
- Y muchas más cosas....

Adicionalmente podemos formatear la salida usando el flag **--format/-f**. Una descripción detallada queda fuera de los objetivos de este curso pero vamos a poner varios ejemplos:

```
# Mostrar la arquitectura y el sistema
> docker inspect --format '{{.Architecture}} es la arquitectura y el SO es {{.Os}}' mysql:8.0.22
amd64 es la arquitectura y el SO es linux
# Mostrar la lista de puertos expuestos
> docker inspect --format '{{.Config.ExposedPorts}}' mysql:8.0.22
map[3306/tcp:{ } 33060/tcp:{ }]
```

Además de los comandos que hemos visto en los apartados anteriores la orden **docker image** tiene una gran variedad de **subcomandos**, que si bien no son necesarios para poder empezar con docker si que es bueno conocer que existen, os recomiendo los siguientes:

- **docker image build** para construir una imagen desde un fichero Dockerfile (se verá en el apartado 6).
- **docker image history** para que se nos muestre por pantalla la evolución de esa imagen.
- **docker image save** / **docker image load** (o **docker save** / **docker load**) para guardar imágenes en fichero y cargarlas desde fichero .
- **docker image tag** ( **docker tag**) para añadir TAGs (versiones) a las distintas imágenes.

**Capturas a entregar:** (incluir capturas de todos los comandos necesarios)

- Comprueba la información de la imagen mysql:latest
- Comprueba los puertos expuestos.