

# Gestión de imágenes Docker

Sistemas de Gestión Empresarial

IES Clara del Rey

En esta unidad se tratan dos aspectos a tratar con las imágenes:

- La gestión local de las imágenes de los contenedores Docker
  - listado,
  - eliminación,
  - historico,
  - borrado
  - etc.
- la creación de forma manual con **docker build**
- la creación de los llamados **Dockerfiles**

- **docker images:** qué imágenes tenemos almacenadas localmente

O las de una imagen en particular

docker images ubuntu:14.04

- **docker search:** información de imágenes que podemos descargar en el registro

- **docker pull:** descarga una imagen sin crear un contenedor

`docker pull alpine:3.10`

- **docker history:** versiones en las que se basa

`docker history nginx`

- **docker rmi**: elimina imágenes almacenadas localmente.

Si queremos todas las imágenes que no estén usando los contenedores, podemos usar

**docker rmi \$(docker images -q)**

donde *docker images -q* lista los ids de las imágenes

Para borrar un contenedor es necesario que esté parado. Si está en marcha hay que pararlo previamente.

- **`docker rm IDENTIFICADOR/NOMBRE`**

Para borrar todos los contenedores parados:

**`docker stop $(docker ps -a -q)`** : para los contenedores **`docker rm $(docker ps -a -q)`** : borra los contenedores

*docker ps -a -q* lista los ids de todos los contenedores

# Eliminar todas las imágenes y contenedores

- **docker system:** Gestiona datos de la aplicación de docker
  - docker system df: muestra el espacio usado en disco
  - docker system prune: elimina datos no usados
- **docker container:** Comandos para la gestión de contenedores
  - docker container prune: elimina todos los contenedores parados

Y mezclando instrucciones:

**docker system prune -a**

# Crear imágenes propias desde un contenedor

Podemos entender a un contenedor como una *capa temporal* de una imagen. Con el comando **commit** podemos convertir esa capa temporal en una imagen.

- **docker commit -a “autor” -m “comentario”**  
**ID/NOMBRE-CONTENEDOR usuario/imagen:[version]**

Por ejemplo

```
docker commit -a “jimportillo” -m “Mi sitio Wordpress” LAMP  
jimportillo/wordpress:1
```

A partir de aquí podemos crear nuevos contenedores con esa imagen

```
docker run -it --name wordpress2 jimportillo/wordpress:1
```

(un problema en este caso es la redirección de puertos en wordpress)



# Subir imágenes a un repositorio (Docker Hub)

## 1. Crear el repositorio

Hay que crear una cuenta en **hub.docker.com** e iniciar sesión. Una vez identificado, acceder a *Repositories* y *Create repository*

## 2. Almacenar la imagen

Iniciamos sesión con **docker login** en un terminal.

Crear la imagen con **docker commit** indicando usuario y nombre de imagen

Subir la imagen con **docker push usuario/imagen**

# Crea tus imágenes!

