

Esta práctica se va a basar en el uso de comando docker realizando ejemplos guiados.

Busquemos una imagen de Ubuntu → docker search ubuntu

NAME		DESCRIPTION	STARS
OFFICIAL	AUTOMATED		
ubuntu		Ubuntu is a Debian-based Linux operating sys…	10237
[OK]			
dorowu/ubuntu-desktop-lxde-vnc		Docker image to provide HTML5 VNC interface	368
	[OK]		
rastasheep/ubuntu-sshd		Dockerized SSH service, built on top of offi	236
	[OK]		
consol/ubuntu-xfce-vnc		Ubuntu container with "headless" VNC session	199
	[OK]		
ubuntu-upstart		Upstart is an event-based replacement for th	102

 Si no se especifica ningún tag, descargaremos la imagen latest por defecto → docker pull ubuntu

```
PS C:\WINDOWS\system32> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:6e9f67fa63b0323e9a1e587fd71c561ba48a034504fb804fd26fd8800039835d
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

- Comprobamos que tenemos la imagen de Ubuntu → docker images
- Vamos a crear un contenedor de la imagen descargada → docker run ubuntu

```
PS C:\WINDOWS\system32> docker run ubuntu
PS C:\WINDOWS\system32>
```

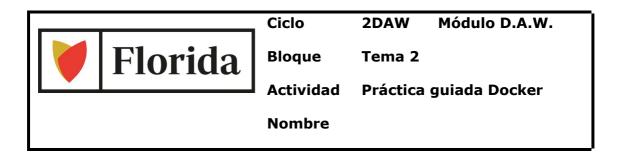
Aparentemente no ha hecho nada, aunque en realidad sí. Ha ejecutado un contenedor de ubuntu y ha terminado.

Vemos los procesos activos que tenemos y no tenemos ninguno →docker ps



 Pero si vemos el histórico de procesos nos encontraremos con el contenedor que ha creado de Ubuntu → docker ps -a

PS C:\WINDOWS\system32> docker ps -a								
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS			
0c412eb17bc9	ubuntu	"/bin/bash"	2 minutes ago	Exited (0) 2 minutes ago				



 Ejecutamos otra vez el comando run pero esta vez hagamos un echo → docker run echo "hola"

```
PS C:\WINDOWS\system32> docker run ubuntu echo hola
hola
PS C:\WINDOWS\system32>
```

Lo que hemos hecho es crear otro contenedor de ubuntu, ejecutar un echo y cerrarlo.

Si ahora vemos los procesos, veremos dos contenedores de ubuntu

```
PS C:\WINDOWS\system32> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
c3134bce7068 ubuntu "echo hola" About a minute ago Exited (0) About a minute ago
c412eb17bc9 ubuntu "/bin/bash" 6 hours ago Exited (0) 6 hours ago
PS C:\WINDOWS\system32>
```

 Para crear un contenedor y que sea interactivo nos ayudamos de las opciones i (interactive) y t (terminal) → docker run -it ubuntu bash

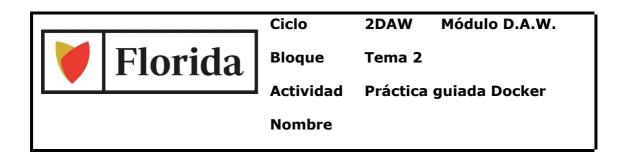
```
PS C:\WINDOWS\system32> docker run -it ubuntu bash
root@efd839931a4d:/#
```

En este punto creemos un fichero dentro del contenedor llamado dentro.txt

```
root@efd839931a4d:/# touch dentro.txt
root@efd839931a4d:/# ls -l
total 64
            2 root root 4096 Oct 29 21:25
drwxr-xr-x
drwxr-xr-x 2 root root 4096 Apr 24 2018
           1 root root
5 root root
                           0 Dec
                                  8 13:41 dentro.txt
-rw-r--r--
                         360 Dec
                                  8 13:40
drwxr-xr-x
           1 root root 4096 Dec
                                  8 13:40
drwxr-xr-x
           2 root root 4096 Apr 24 2018
drwxr-xr-x
```

 Con la opción ctrl+p+q salimos del contenedor y volvemos a la máquina, pero el contenedor sigue activo. Comprobadlo con los procesos que están activos.

```
root@efd839931a4d:/#
PS C:\WINDOWS\system32> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
efd839931a4d ubuntu "bash" 4 minutes ago Up 4 minutes
PS C:\WINDOWS\system32>
```



 Para volver a un proceso o contenedor activo usamos el comando attach y el ID del contenedor que obtendremos ejecutando docker ps (en mi caso efd839931a4d).

```
PS C:\WINDOWS\system32> docker attach efd839931a4d root@efd839931a4d:/# ls -l total 64 drwxr-xr-x 2 root root 4096 Oct 29 21:25 bin drwxr-xr-x 2 root root 4096 Apr 24 2018 boot -rw-r--r- 1 root root 0 Dec 8 13:41 dentro.txt drwxr-xr-x 5 root root 360 Dec 8 13:40 dev drwxr-xr-x 1 root root 8 Dec 8 13:40 dev
```

 Si queremos salir del contenedor y matar su proceso ejecutamos el comando → exit

```
root@efd839931a4d:/# exit
exit
PS C:\WINDOWS\system32> <mark>docker</mark> ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PS C:\WINDOWS\system32>
```

 Para volver a un contenedor que no está activo usaremos el comando start y a continuación attach, ambos seguidos de su ID que encontraremos viendo el histórico de procesos.

```
CONTAINER ID
efd839931a4d
                             IMAGE
                                                          COMMAND
                                                                                       CREATED
                                                                                                                     STATUS
                                                                                                                     Exited (0) About a minute ago
Exited (0) 15 minutes ago
Exited (0) 6 hours ago
                             ubuntu
                                                           "bash"
                                                                                        10 minutes ago
 3134bce7068
                                                           "echo hola"
                                                                                        15 minutes ago
                             ubuntu
0c412eb17bc9
                             ubuntu
                                                           "/bin/bash"
                                                                                       6 hours ago
 PS C:\WINDOWS\system32> docker start efd839931a4d
efd839931a4d
PS C:\WINDOWS\system32> docker attach efd839931a4d root@efd839931a4d:/# ls -1
total 64
 drwxr-xr-x 2 root root 4096 Oct 29 21:25 bin
drwxr-xr-x 2 root root 4096 Apr 24 2018 boot
-rw-r--r- 1 root root 0 Dec 8 13:41 dentro.txt
                                                   8 13:52
                   5 root root
 drwxr-xr-x
```



Ciclo 2DAW Módulo D.A.W.

Bloque Tema 2

Actividad Práctica guiada Docker

Nombre

GESTIÓN DE IMÁGENES Y CONTENEDORES

 Partamos del contenedor que estamos usando de ubuntu donde hemos creado el fichero dentro.txt

Vamos a hacer una imagen del contenedor → docker commit efd839931a4d
 ubuntu con fichero

```
PS C:\WINDOWS\system32> docker ps -a
CONTAINER ID
                                        COMMAND
                   IMAGE
                                                             CREATED
efd839931a4d
                   ubuntu
                                        "bash"
                                                             18 minutes ago
c3134bce7068
                   ubuntu
                                        "echo hola"
                                                            23 minutes ago
                                        "/bin/bash"
0c412eb17bc9
                   ubuntu
                                                             6 hours ago
PS C:\WINDOWS\system32> docker commit efd839931a4d ubuntu_con_fichero
sha256:d292634424600d5a4496cb0dbeadfce806b151b8e791ba00ef29fb10317b351c
PS C:\WINDOWS\system32>
```

 Si vemos las imágenes que tenemos nos debería aparecer la que acabamos de crear

```
PS C:\WINDOWS\system32> <mark>docker</mark> images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu_con_fichero latest d29263442460 About a minute ago 64.2MB
```

 Podemos empaquetar la imagen para poder compartirla con el comando save → docker save -o miimagen.tar ubuntu_con_fichero

Para borrar imágenes usamos el comando rmi → docker rmi ubuntu_con_fichero

```
PS C:\WINDOWS\system32> docker rmi ubuntu_con_fichero
Untagged: ubuntu_con_fichero:latest
Deleted: sha256:d292634424600d5a4496cb0dbeadfce806b151b8e791ba00ef29fb10317b351c
Deleted: sha256:4f8b297f29f8907a19f200af5e640e0789a929f9863832bf08c91e3c6191e7fa
```

Si hubiéramos tenido contenedores de esa imagen, no nos hubiera dejado borrarla. Tendríamos que haber borrado primero los contenedores y después la imagen, con el comando rm más el ID del contenedor \rightarrow docker rm ID



Ciclo 2DAW Módulo D.A.W.

Bloque Tema 2

Actividad Práctica guiada Docker

Nombre

Para importar una imagen a partir de un fichero usamos load docker load –input miimagen.tar

Comprobamos las imágenes que tenemos

PS C:\WINDOWS\system32> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu_con_fichero latest d29263442460 20 minutes ago 64.2MB