

# Actividades Docker

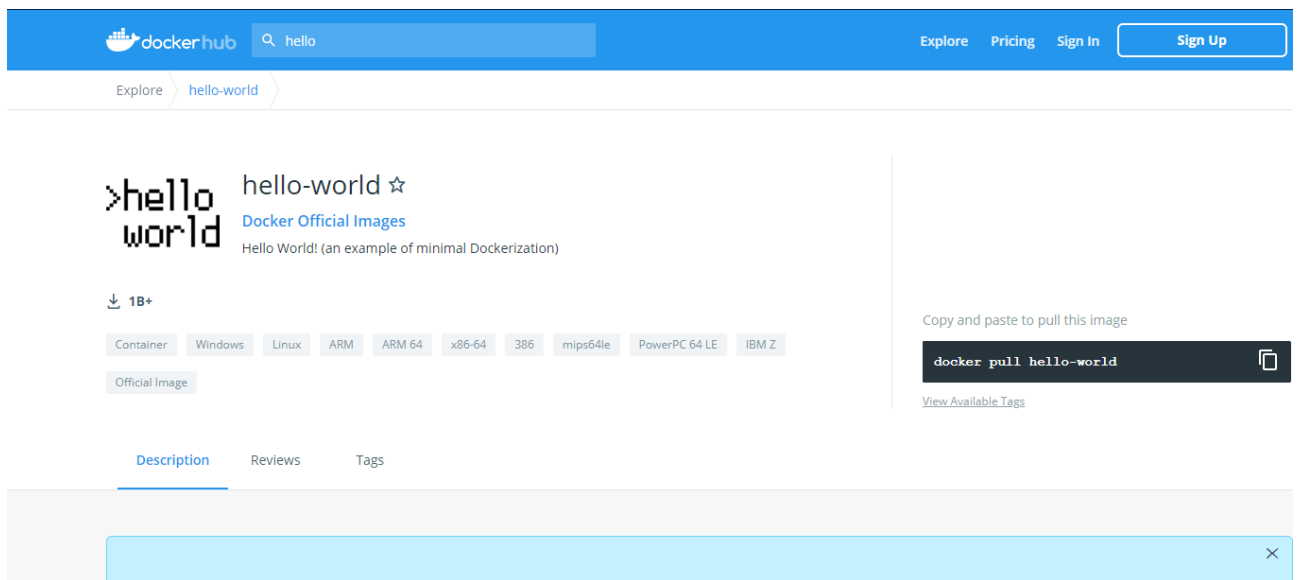
Juan Carlos Navidad García

### ACTIVIDAD:

1. Verifica las imágenes que tienes en tu sistema

```
PS C:\Users\jcnave> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd         latest    f3cffeea581b   38 minutes ago 138MB
mysql         8.0.24    0627ec6901db   3 weeks ago    556MB
mysql         latest    0627ec6901db   3 weeks ago    556MB
PS C:\Users\jcnave>
```

2. Busca en docker-hub otra imagen de hola-mundo y descargatela



3. Muestra el historial de cambios de una de las imágenes que tengas descargadas

```
PS C:\Users\jcnave> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd         latest    f3cffeea581b   42 minutes ago 138MB
mysql         8.0.24    0627ec6901db   3 weeks ago    556MB
mysql         latest    0627ec6901db   3 weeks ago    556MB
hello-world    latest    d1165f221234   2 months ago   13.3kB
PS C:\Users\jcnave> docker history d1165f221234
IMAGE          CREATED        CREATED BY          SIZE      COMMENT
d1165f221234   2 months ago  /bin/sh -c #(nop)  CMD ["/hello"]      0B
<missing>      2 months ago  /bin/sh -c #(nop)  COPY file:7bf12aab75c3867a... 13.3kB
PS C:\Users\jcnave>
```

4. Muestra las características de la imagen ultima que te hayas descargado

```
PS C:\Users\jcnave> docker image inspect d1165f221234
[
  {
    "Id": "sha256:d1165f2212346b2bab48cb01c1e39ee8ad1be46b87873d9ca7a4e434980a7726",
    "RepoTags": [
      "hello-world:latest"
    ],
    "RepoDigests": [
      "hello-world@sha256:f2266cbfc127c960fd30e76b7c792dc23b588c0db76233517e1891a4e357d519"
    ],
    "Parent": "",
    "Comment": "",
    "Created": "2021-03-05T23:25:25.230064203Z",
    "Container": "f5a78ef54769bb8490754e9e063a89f90cc8eee6a6c5a0a72655826e99df116e",
    "ContainerConfig": {
      "Hostname": "f5a78ef54769",
      "Domainname": "",
      "User": "",
      "AttachStdin": false,
      "AttachStdout": false,
      "AttachStderr": false,
      "Tty": false,
      "OpenStdin": false,
      "StdinOnce": false,
      "Env": [
        "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
      ],
      "Cmd": [
        "/bin/sh",
        "-c",
        "#(nop) ",
        "CMD [\"/hello\"]"
      ],
      "Image": "sha256:77fe0a37fa6ce641a004815f2761a9042618557d253f312cd3da61780e372c8f",
      "Volumes": null,
      "WorkingDir": "",
      "Entrypoint": null,
      "OnBuild": null,
      "Labels": {}
    },
    "DockerVersion": "19.03.12",
    "Author": "",
    "Config": {
      "Hostname": "",
      "Domainname": ""
    }
  }
]
```

5. Elimina la última imagen descargada con un solo comando

```
PS C:\Users\jcnave> docker rmi d1165f221234
Untagged: hello-world:latest
Untagged: hello-world@sha256:f2266cbfc127c960fd30e76b7c792dc23b588c0db76233517e1891a4e357d519
Deleted: sha256:d1165f2212346b2bab48cb01c1e39ee8ad1be46b87873d9ca7a4e434980a7726
Deleted: sha256:f22b99068db93900abe17f7f5e09ec775c2826ecfe9db961fea68293744144bd
PS C:\Users\jcnave> █
```

6. Utiliza el comando `info` para mostrar información de sistema (imágenes, contenedores...)

```
PS C:\Users\jcnave> docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  app: Docker App (Docker Inc., v0.9.1-beta3)
  buildx: Build with BuildKit (Docker Inc., v0.5.1-docker)
  compose: Docker Compose (Docker Inc., 2.0.0-beta.1)
  scan: Docker Scan (Docker Inc., v0.8.0)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 2
 Server Version: 20.10.6
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: runc io.containerd.runc.v2 io.containerd.runtime.v1.linux
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 05f951a3781f4f2c1911b05e61c160e9c30eaa8e
 runc version: 12644e614e25b05da6fd08a38ffa0cfe1903fdec
 init version: de40ad0
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 5.4.72-microsoft-standard-WSL2
 Operating System: Docker Desktop
 OSTYPE: linux
 Architecture: x86_64
 CPUs: 12
 Total Memory: 12.44GiB
 Name: docker-desktop
```

### 7. Utiliza la opción `--help` sobre el comando `run` para ver que realiza

```
PS C:\Users\jcnave> docker run --help

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container

Options:
  --add-host list          Add a custom host-to-IP mapping
                           (host:ip)
  -a, --attach list        Attach to STDIN, STDOUT or STDERR
  --blkio-weight uint16    Block IO (relative weight),
                           between 10 and 1000, or 0 to
                           disable (default 0)
  --blkio-weight-device list
                           Block IO weight (relative device
                           weight) (default [])
  --cap-add list           Add Linux capabilities
  --cap-drop list          Drop Linux capabilities
  --cgroup-parent string   Optional parent cgroup for the
                           container
  --cgroupns string        Cgroup namespace to use
                           (host|private)
                           'host': Run the container in
                           the Docker host's cgroup namespace
                           'private': Run the container in
                           its own private cgroup namespace
                           '': Use the cgroup
                           namespace as configured by the
                           default-cgroupns-mode
                           option on the daemon (default)
  --cidfile string         Write the container ID to the file
  --cpu-period int         Limit CPU CFS (Completely Fair
                           Scheduler) period
  --cpu-quota int          Limit CPU CFS (Completely Fair
                           Scheduler) quota
  --cpu-rt-period int      Limit CPU real-time period in
                           microseconds
  --cpu-rt-runtime int     Limit CPU real-time runtime in
                           microseconds
  -c, --cpu-shares int     CPU shares (relative weight)
  --cpus decimal           Number of CPUs
  --cpuset-cpus string     CPUs in which to allow execution
                           (0-3, 0,1)
  --cpuset-mems string     MEMs in which to allow execution
                           (0-3, 0,1)
  -d, --detach             Run container in background and
                           print container ID
```

### 8. Del resultado anterior investiga para que sirven las siguientes opciones

- `-d`: Para abrir un contenedor en segundo plano.
- `-e`: Establece variables de entorno.
- `-h`: Establece un nombre de red para el contenedor.
- `-i`: Establece una IP para el contenedor.
- `-t`: Asigna un pseudo-TTY.
- `-w`: Directorio de trabajo dentro del contenedor.

## Curso: DOCKER PARA EL DESARROLLO

---

- -u: Tienes que acompañarlo de un usuario o un UID.
- -v: Montar en un volumen específico.
- -p: Establecer puertos.
- --name: Le da un nombre al contenedor.
- --network: Conecta el contenedor a una red determinada.
- --ip: Le da una IP concreta al contenedor.
- --mount: Para montar un disco duro.
- --rm: Para eliminar un contenedor

### ACTIVIDAD:

CREA LOS SIGUIENTES CONTENEDORS

# Ejecuto un servidor Apache sin el flag -d ni redirección de puertos. Se bloquea el terminal mostrando los logs y tendré que salir con Ctrl+C

> docker run httpd

```
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using
172.17.0.22. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using
172.17.0.22. Set the 'ServerName' directive globally to suppress this message
[Mon Dec 07 18:27:28.561909 2020] [mpm_event:notice] [pid 1:tid 140253864719488]
AH00489: Apache/2.4.46 (Unix) configured -- resuming normal operations
[Mon Dec 07 18:27:28.562072 2020] [core:notice] [pid 1:tid 140253864719488] AH00094:
Command line: 'httpd -D FOREGROUND'
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\jcnave> docker run httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
69692152171a: Pull complete
7284b4e0cc7b: Pull complete
3678b2d55ccd: Pull complete
ab492cf0b2a4: Pull complete
991f7f97a9d8: Pull complete
Digest: sha256:e4c2b93c04762468a6cce6d507d94def02ef4dc285278d0d926e09827f4857db
Status: Downloaded newer image for httpd:latest
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172
.17.0.2. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172
.17.0.2. Set the 'ServerName' directive globally to suppress this message
[Wed May 12 08:18:24.538684 2021] [mpm_event:notice] [pid 1:tid 140530468656256] AH00489: Apache
/2.4.46 (Unix) configured -- resuming normal operations
[Wed May 12 08:18:24.538794 2021] [core:notice] [pid 1:tid 140530468656256] AH00094: Command lin
e: 'httpd -D FOREGROUND'
█
```

# Ejecuto un servidor Apache en background y accediendo desde el exterior a través del puerto 8888 de mi máquina.

> docker run -d -p 8888:80 httpd

```
PS C:\Users\jcnave> docker run -d -p 8080:80 httpd
b21116d749886f8e9510a6fac328622f134e82a630c4deb1d55b8676cd2044ae
PS C:\Users\jcnave> █
```

## Curso: DOCKER PARA EL DESARROLLO

---

# Creación de un servidor de base de datos mariadb accediendo desde el exterior a través del puerto 3306 y estableciendo una contraseña de root mediante una variable de

# entorno

> docker run -it -d -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=root mariadb

```
PS C:\Users\jcnave> docker run -it -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root mariadb
Unable to find image 'mariadb:latest' locally
latest: Pulling from library/mariadb
345e3491a907: Pull complete
57671312ef6f: Pull complete
5e9250ddb7d0: Pull complete
2d512e2ff778: Pull complete
57c1a7dc2af9: Pull complete
5f1da40ab8a2: Pull complete
5d5cfc668726: Pull complete
deb86f297614: Pull complete
6861f77e80d5: Pull complete
81a5dea1514d: Pull complete
e3b5cf124c10: Pull complete
3352c2c9d21c: Pull complete
Digest: sha256:36288c675a192bd0a8a99cd6ba0780e31df85f0bfd0cbb204837cd108be3d236
Status: Downloaded newer image for mariadb:latest
80fc5164ef64b1f52d611a4be8cd4783e9f978cba4932d987f849b674ecbe92a
PS C:\Users\jcnave> █
```



# Arrancar el contenedor con nombre servidorWeb (debe estar parado)

> docker start servidorWeb

```
PS C:\Users\jcnave> docker run -it -d -p 3306:3306 --name servidorWeb -e MYSQL_ROOT_PASSWORD=root mariadb
67137838e8b360f5cd2029b1ae104fc0623270028cbcc8d3a05fc91b0cdd02ec
PS C:\Users\jcnave> █
```

# Crear un contenedor de debian 9 y mostrar el contenido de una carpeta establecida con el parámetro -w

> docker run -it -w /etc debian:9 ls

```
PS C:\Users\jcnave> docker run -it -w /etc debian:9 ls
Unable to find image 'debian:9' locally
9: Pulling from library/debian
bfde2ec33fbc: Pull complete
Digest: sha256:4f8deed5755299e2d7ae1f5649656de89a6b7e30c0be37a3a1b4a49f0af51f7e
Status: Downloaded newer image for debian:9
adduser.conf      hostname          nsswitch.conf    rmt
alternatives      hosts            opt              securetty
apt               init.d           os-release       security
bash.bashrc       iproute2         pam.conf         selinux
bindresvport.blacklist issue            pam.d            shadow
cron.daily        issue.net        passwd           shadow-
debconf.conf      kernel           passwd-          shells
debian_version    ld.so.cache      profile          skel
default           ld.so.conf       profile.d        staff-group-for-usr-local
deluser.conf      ld.so.conf.d     rc0.d            subgid
dpkg              libaudit.conf    rc1.d            subuid
environment       localtime        rc2.d            systemd
fstab             login.defs       rc3.d            terminfo
gai.conf          logrotate.d      rc4.d            timezone
group             machine-id       rc5.d            update-motd.d
group-            mke2fs.conf     rc6.d
gshadow           motd             rcS.d
host.conf         mtab            resolv.conf
PS C:\Users\jcnave> █
```

## Curso: DOCKER PARA EL DESARROLLO

Al crear el contenedor se ejecuta la orden `ls` desde el directorio `/etc`, posteriormente el contenedor pasa a estar parado. Y ya no podremos acceder a él. Explicaremos en el próximo apartado el porqué.

Conforme vayamos creando contenedores hay dos órdenes que nos van a interesar para hacer un seguimiento de qué tenemos en nuestro sistema:

# Mostrar los contenedores en ejecución (Estado Up)

> `docker ps`

```
PS C:\Users\jcnave> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
1d5f0ae9b56d   httpd     "httpd-foreground"     13 minutes ago Up 13 minutes  0.0.0.0:8080->80/tcp, :::8080->80/tcp
8aadf347743a   httpd     "httpd-foreground"     37 minutes ago Up 37 minutes  0.0.0.0:8888->80/tcp, :::8888->80/tcp
PS C:\Users\jcnave>
```

# Mostrar todos los contenedores creados ya estén en ejecución (Estado Up) o parados (Estado Exited)

> `docker ps -a`

```
PS C:\Users\jcnave> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
1d5f0ae9b56d   httpd     "httpd-foreground"     14 minutes ago Up 14 minutes  0.0.0.0:8080->80/tcp, :::8080->80/tcp
1212235a1ab7   debian:9   "ls"                   18 minutes ago Exited (0) 18 minutes ago
80fc5164ef64   mariadb   "docker-entrypoint.s..." 37 minutes ago Exited (0) 27 minutes ago
8aadf347743a   httpd     "httpd-foreground"     38 minutes ago Up 38 minutes  0.0.0.0:8888->80/tcp, :::8888->80/tcp
PS C:\Users\jcnave>
```

### ACTIVIDAD:

CREA LOS SIGUIENTES CONTENEDORS

### EJECUTAR ÓRDENES EN LOS CONTENEDORES. DOCKER EXEC

Con los contenedores en ejecución vamos a querer ejecutar ordenes en ellos. Querremos realizar operaciones como:

- Instalar paquetes.
- Modificar o ver el contenido de ciertos ficheros.
- Habilitar ciertos módulos de servicios
- etc...

Esto lo podemos hacer de **dos maneras** o bien **obteniendo un terminal** del contenedor y ejecutando las órdenes necesarias desde allí **o bien directamente** ejecutando una orden determinada "contra" el contenedor. Para ambos casos voy a necesitar la orden **docker exec** y es **NECESARIO QUE EL CONTENEDOR ESTÉ EN EJECUCIÓN**.

La sintaxis de esta orden es bastante sencilla y muy similar a la de docker run:

**docker exec [opciones] nombre\_contenedor orden [argumentos]**

Algunas de las opciones más importantes son:

- **-it** (-i y -t juntos) si vamos a querer tener interactividad con el contenedor ejecutando un shell (/bin/bash normamente). Una vez tenemos el terminal ya podremos trabajar desde dentro del propio sistema.
- **-u o --user** si quiero ejecutar la orden como si fuera un usuario distinto del de root.
- **-w o --workdir** si quiero ejecutar la orden desde un directorio concreto.

Lo vamos a ver mejor con algunos ejemplos:

# Obtener un terminal en un contenedor que ejecutar un servidor Apache (httpd) y que se llama web

> docker exec -it web /bin/bash

root@5d96ce1f7374:/usr/local/apache2#

```
PS C:\Users\jcnave> docker run -it -d -p 8080:80 --name web httpd
1d5f0ae9b56d19d0567659a03881c0a43946b9716013533becf31ea01066fd53
PS C:\Users\jcnave>
PS C:\Users\jcnave> docker exec -it web /bin/bash
root@1d5f0ae9b56d:/usr/local/apache2#
```

# Mostrar el contenido de la carpeta /usr/local/apache2/htdocs del contenedor web. Como no hace falta interactividad no es necesario -it

> docker exec web ls /usr/local/apache2/htdocs

```
PS C:\Users\jcnave> docker exec web ls /usr/local/apache2/htdocs
index.html
PS C:\Users\jcnave>
```

# Crear directamente un fichero "HOLA MUNDO" en el directorio raíz del servidor apache. Utilizo sh -c para ordenes compuestas o complejas

> docker exec -it web sh -c "echo 'HOLA MUNDO' > /usr/local/apache2/htdocs/index.html"

```
PS C:\Users\jcnave> docker exec -it web sh -c "echo 'HOLA MUNDO' > /usr/local/apache2/htdocs/index.html"
PS C:\Users\jcnave> docker exec -it web /bin/bash
root@1d5f0ae9b56d:/usr/local/apache2# cat /usr/local/apache2/htdocs/index.html
HOLA MUNDO
root@1d5f0ae9b56d:/usr/local/apache2#
```

Adicionalmente existe otra orden que nos va a ser de mucha utilidad cuando trabajemos con contenedores. la orden **docker cp** que me permite mover ficheros desde mi sistema al contenedor y desde el contenedor a mi sistema. Su sintaxis es muy sencilla y la vamos a ilustrar con dos ejemplos, uno en cada sentido:

# Copiar mi fichero prueba.html al fichero /usr/local/apache2/htdocs/index.html de mi contenedor llamado web que es un servidor Apache (httpd)

> docker cp prueba.html web:/usr/local/apache2/htdocs/index.html

```
root@1d5f0ae9b56d:/usr/local/apache2# touch prueba.html
root@1d5f0ae9b56d:/usr/local/apache2# ls
bin build cgi-bin conf error htdocs icons include logs modules prueba.html
root@1d5f0ae9b56d:/usr/local/apache2# cp prueba.html /usr/local/apache2/htdocs
root@1d5f0ae9b56d:/usr/local/apache2# ls /usr/local/apache2/htdocs
index.html  prueba.html
root@1d5f0ae9b56d:/usr/local/apache2#
```

**NOTA: LOS CONTENEDORES VIENEN CON SOLO LO IMPRESCINDIBLE INSTALADO. SI QUIERO INSTALAR ALGO DEBO NORMALMENTE HACER ANTES UN APT UPDATE (ya que la mayoría son basados en Debian).**

### EJERCICIOS

1. Crear un contenedor con el servicio mysql versión 8.0.22 con el nombre mysql8 y que se pueda acceder desde el puerto 3306 en el host. Además ejecútalo en segundo plano de forma iterativa.

```
PS C:\Users\jcnave> docker run -d --name mysql8 -p 3306:30 mysql:8.0.22
3603c5101cb5e288f835e6d6a4e365a6ba402c81f251ab94fa03f46d5bd13d48
PS C:\Users\jcnave> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
3603c5101cb5	mysql:8.0.22	"docker-entrypoint.s..."	15 seconds ago	Exited (1) 11 seconds ago

2. Lista las imágenes descargadas

```
PS C:\Users\jcnave> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd         latest    f3cffeea581b   2 hours ago    138MB
debian        9         acf04b8ccc37   8 hours ago    101MB
mariadb       latest    992bce5ed710   2 weeks ago    401MB
mysql         8.0.24    0627ec6901db   3 weeks ago    556MB
mysql         latest    0627ec6901db   3 weeks ago    556MB
mysql         8.0.22    d4c3cafb11d5   3 months ago    545MB
PS C:\Users\jcnave>
```

3. ¿Qué información nos muestra el comando Docker images?

Nos muestra las imágenes descargadas.

4. ¿Que realiza la opción -p sobre Docker pull o Docker run

Le asigna un puerto al contenedor.

5. Borra la imagen mysql8 usando su image id.

```
PS C:\Users\jcnave> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd         latest    f3cffeea581b   2 hours ago    138MB
debian        9         acf04b8ccc37   8 hours ago    101MB
mariadb       latest    992bce5ed710   2 weeks ago    401MB
mysql         8.0.24    0627ec6901db   3 weeks ago    556MB
mysql         latest    0627ec6901db   3 weeks ago    556MB
mysql         8.0.22    d4c3cafb11d5   4 months ago    545MB
PS C:\Users\jcnave> docker rmi -f d4c3cafb11d5
Untagged: mysql:8.0.22
Untagged: mysql@sha256:0fd2898dc1c946b34dceaccc3b80d38b1049285c1dab70df7480de62265d6213
Deleted: sha256:d4c3cafb11d573699728f9e7de10d1b976089b01298c0360e03f0afd9a1a8b36
PS C:\Users\jcnave>
```

6. ¿Cuál de las siguientes instrucciones son correctas?

- > docker rmi mysql:8.0.22, es correcta
- > docker rmi dd7265748b5d, es correcta
- > docker image rm mysql:8.0.2, no es correcta
- > docker image rm dd7265748b5d, no es correcta
- > docker rmi mysql:8.0.22 mysql:5.7, es correcta

7. ¿Es posible borrar una imagen si un contenedor la esta usando?

Es posible forzándola a borrarse, es decir, utilizando -f en el comando.

8. Borra todas las imágenes

```
PS C:\Users\jcnave> docker rmi $(docker images -q)
Untagged: httpd:latest
Untagged: httpd@sha256:e4c2b93c04762468a6cce6d507d94def02ef4dc285278d0d926e09827f4857db
Deleted: sha256:f3cffeea581b3306a13d80b25a437f73f767b8f27afdb29393b3764b9dfacea69
Deleted: sha256:887d0217b7251e4d5ad46486c19c5e2c82ae8aa6d53c3ffb729f032028a5f1c9
Deleted: sha256:e04f6a8ae4f2939d3b73fd06085169c625486a81d3823fba3fdad029b2017347
Deleted: sha256:85643304b5484a30ffe669196e71d6b763661800e12beb7ff131615040906adb
Deleted: sha256:7524155083fa19d15b8902105916a8da484d1ff5d362d3530d871511c1f13f2a
Deleted: sha256:02c055ef67f5904019f43a41ea5f099996d8e7633749b6e606c400526b2c4b33
Untagged: mariadb:latest
Untagged: mariadb@sha256:36288c675a192bd0a8a99cd6ba0780e31df85f0bfd0cbb204837cd108be3d236
Deleted: sha256:992bce5ed7107642a27af59c2a2ddb7e589a44b639342b9501044511ab00aca0
Deleted: sha256:597a23bb8470db04ddb72ed1ff1e529baa63f69dc1960365742c6e17d227d732
Deleted: sha256:109848e3c68751ab413706f51c358c8fb4d4b0ff623a59611592ebf2225c093e
Deleted: sha256:39c9c394374ff05519631d5864019dd69b7468dc1362e0eb65e2b134cd95c7f7
Deleted: sha256:526e2e851bcefe0644e7d045ffab54d1379bca36dafc377eb8bb8d1b035e603d
Deleted: sha256:a9de7edab48166f3f5b17e990f080d76514d804e98fd2919cac95ce616cb0fbb
Deleted: sha256:40473878582bd910c06b2efd2356c183d39ef12b0b570622778c4cb2ad83025e
Deleted: sha256:bf3b32e6f91ea6141c6c6cddd96e0302dde03595ef51dd79e9b403c4683a0bd9
Deleted: sha256:51200e23239dee1b33bfce273843416b9ce71829a03d3b1c705a92e1d8ee386c
Deleted: sha256:cc5b4ec409e0f83b6d66cb6bdc04d5092ee3525862fa6d035606ad222fb08b1f
Deleted: sha256:3dd8c8d4fd5b59d543c8f75a67cdfaab30aef5a6d99aea3fe74d8cc69d4e7bf2
Deleted: sha256:8d8dceacec7085abcab1f93ac1128765bc6cf0caac334c821e01546bd96eb741
Deleted: sha256:ccdbb80308cc5ef43b605ac28fac29c6a597f89f5a169bbbedbb8dec29c987439
PS C:\Users\jcnave> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
PS C:\Users\jcnave>
```