

# docker

**PILAR GUZMÁN**  
**2ºDAW**

**1- Lee el documento contenedores que hay en Moodle y responde razonadamente las siguientes cuestiones en este mismo documento y entrégalo por correo electrónico**

**- ¿Qué es un contenedor?**

Empaquetado portable normalizado para sus aplicaciones.

Un paquete de software estándar (conocido como “contenedor”) agrupa el código de una aplicación con las bibliotecas y los archivos de configuración asociados, junto con las dependencias necesarias para que la aplicación se ejecute. Esto permite a los desarrolladores y profesionales de TI implementar aplicaciones sin problemas en todos los entornos.

**- ¿Qué es una imagen de docker?**

Una imagen es una especie de plantilla, una captura del estado de un contenedor. Hay muchas imágenes públicas con elementos básicos como Java

**- ¿Qué relación/diferencia hay entre un contenedor y una imagen?**

Las imágenes se utilizan para crear contenedores, y nunca cambian.

**- ¿Qué es una máquina virtual? ¿Que diferencias hay entre una máquina virtual y un contenedor? ¿Que ventajas / inconvenientes presentan ambas soluciones?**

Las máquinas virtuales son ordenadores de software que proporcionan la misma funcionalidad que los ordenadores físicos. Como ocurre con los ordenadores físicos, ejecutan aplicaciones y un sistema operativo.

Una máquina virtual requiere combinar su propio sistema operativo y el hipervisor con el sistema operativo del host. En cambio, un contenedor no necesita un sistema operativo para funcionar y aprovechan el sistema operativo del host para ejecutarse.

Los contenedores permiten desplegar aplicaciones más rápido, arrancarlas y pararlas más rápido y aprovechar mejor los recursos de hardware. Las máquinas virtuales nos permiten crear sistemas completos totalmente aislados, con mayor control sobre el entorno y mezclando sistemas operativos host y huésped.

Cada máquina virtual contiene una imagen de sistema operativo, bibliotecas, aplicaciones, etc., por lo que puede ser bastante grande.

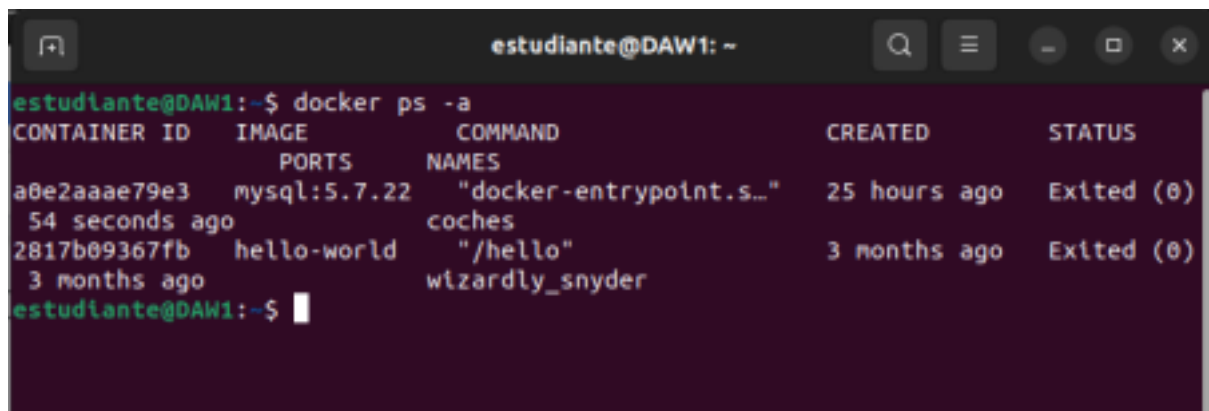
Aunque los contenedores son portables, están limitados al sistema operativo para el que se han definido. Por ejemplo, un contenedor para Linux no se puede ejecutar en Windows y

viceversa.

- ¿Busca información e indica qué es docker compose?

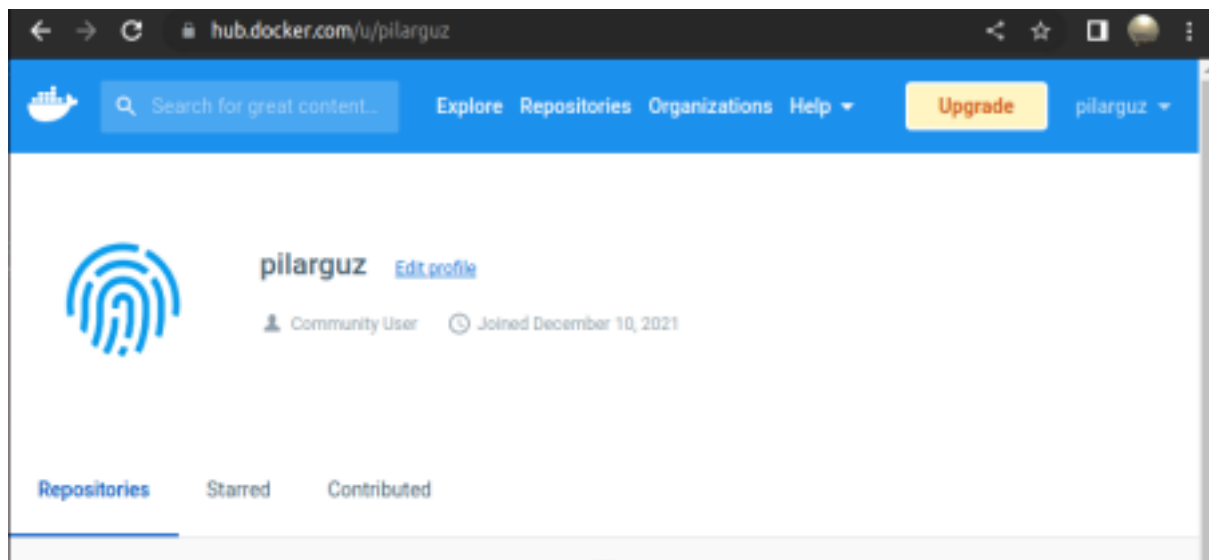
Docker Compose es una herramienta para definir y ejecutar aplicaciones de Docker de varios contenedores. En Compose, se usa un archivo YAML para configurar los servicios de la aplicación. Después, con un solo comando, se crean y se inician todos los servicios de la configuración.

**2- Comprueba si tienes instalado docker (haz captura de todo lo que hagas y lo envías por e-mail) y en caso de no tenerlo, instálalo.**



```
estudiante@DAW1: ~  
estudiante@DAW1:~$ docker ps -a  
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS  
a8e2aaae79e3   mysql:5.7.22   "docker-entrypoint.s..." 25 hours ago   Exited (0)  
54 seconds ago  
2817b09367fb   hello-world    "/hello"                 3 months ago   Exited (0)  
3 months ago  
estudiante@DAW1:~$
```

**3- Créate una cuenta de usuario en la web oficial de docker.**



**4- Busca en el repositorio los 4 primeros contenedores que aparezcan en el listado y anótalos aquí.**

alpine, Ubuntu, Busybox, Python

**5- Descarga los dos primeros contenedores, y carga el segundo. Anota la ID de este.**

```
estudiante@DAW1: ~$ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
213ec9aee27d: Pull complete
Digest: sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d9730fc2ad
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
estudiante@DAW1: ~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
2b55860d4c66: Pull complete
Digest: sha256:20fa2d7bb4de7723f542be5923b06c4d704370f0390e4ae9e1c833c8785644c1
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
estudiante@DAW1: ~$ docker run ubuntu
estudiante@DAW1: ~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
ca2d1cbb8135   ubuntu        "bash"                  12 seconds ago Exited (0) 12 seconds ago
a0e2aaae79e3   youthful_bhaskara
coches        "docker-entrypoint.s..." 25 hours ago   Exited (0) 12 minutes ago
2817b09367fb   hello-world   "/hello"                3 months ago   Exited (0) 3 months ago
wizardly_snyder
estudiante@DAW1: ~$
```

ID: ca2d1cbb8135

6- Obtén un listado de los contenedores que hay descargados y para el que esté arrancado

```
estudiante@DAW1: ~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
ca2d1cbb8135   ubuntu        "bash"                  About a minute ago Exited (0) About a minute ago
a0e2aaae79e3   youthful_bhaskara
coches        "docker-entrypoint.s..." 25 hours ago   Exited (0) 13 minutes ago
2817b09367fb   hello-world   "/hello"                3 months ago   Exited (0) 3 months ago
wizardly_snyder
estudiante@DAW1: ~$
```

7- Borra el primer contenedor, y arranca el 3 y 4 que pusiste en la lista de la cuestión 4.

```
estudiante@DAW1: ~$ docker rm ca2d1cbb8135
ca2d1cbb8135
estudiante@DAW1: ~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORT
5 NAMES
a0e2aaae79e3   youthful_bhaskara
coches        "docker-entrypoint.s..." 25 hours ago   Exited (0) 16 minutes ago
2817b09367fb   hello-world   "/hello"                3 months ago   Exited (0) 3 months ago
wizardly_snyder
estudiante@DAW1: ~$
```

```

estudiante@DAW1:~$ docker run python
Unable to find image 'python:latest' locally
latest: Pulling from library/python
23858da423a6: Pull complete
326f452ade5c: Pull complete
a42821cd14fb: Pull complete
8471b75885ef: Pull complete
8ffa7aaef404: Pull complete
15132af73342: Pull complete
aaf3b07565c2: Pull complete
736f7bc16867: Pull complete
94da21e53a5b: Pull complete
Digest: sha256:e9c35537103a2801a30b15a77d4a56b35532c964489b125ec1ff24f3d5b53409
Status: Downloaded newer image for python:latest
estudiante@DAW1:~$ docker run busybox
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
729ce43e2c91: Pull complete
Digest: sha256:ad9bd57a3a57cc95515c537b89aaa69d83a6df54c4050fcf2b41ad367bec0cd5
Status: Downloaded newer image for busybox:latest
estudiante@DAW1:~$

```

8- Muestra las salidas de la imagen de alguno de los contenedores que haya activos.

```

estudiante@DAW1:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
51e311220cec   busybox        "sh"                    3 minutes ago   Exited (0) 3 minutes ago
sad_haslett
b9854f095c87   python        "python3"              4 minutes ago   Exited (0) 4 minutes ago
festive_boyd
a0e2aaae79e3   mysql:5.7.22  "docker-entrypoint.s..." 25 hours ago    Exited (0) 22 minutes ago
coches
2817b09367fb   hello-world    "/hello"               3 months ago    Exited (0) 3 months ago
wizardly_snyder
estudiante@DAW1:~$

```

9- Busca en la web la sintaxis para crear un contenedor / para arrancar un contenedor  
 docker run -d --name nombreContenedor

10 – Busca en la web la sintaxis para trabajar con una imagen

concreta. docker exec -it nombreContenedor /bin/bash

## Principales instrucciones con Docker

docker run imagen..... para arrancar una imagen (si no existe, la descarga también)

docker run 'imagen:versión'.....para arancar una imagen con una versión concreta.

docker pull imagen.....descarga la imagen pero no la arranca (por defecto la mas reciente)

docker images.....muestra las imágenes que tenemos descargadas

docker images | head.....muestra las primeras líneas de las imágenes descargadas.

docker ps .....muestra las imágenes que están arrancadas, activas

docker ps -a .....muestra las últimas imágenes utilizadas control

C.....frena, para una imagen arrancada

docker start ID..... si queremos reiniciar-recuperar el contenedor que hemos paralizado

**docker log**

**docker log -f ID.....muestra las salidas de la imagen**

**docker exec ID.....ejecuta un comando dentro de un contenedor que está  
arrancado**

**docker exec -it ID sh.....( -i crea una sesión interactiva, -t emula una terminal... sh  
es una shell)**

**docker stop ID..... Para un contenedor**

**docker run -d imagen ..... para arrancar una imagen “background”**

**docker rm ID.....borra un contenedor**