Ejercicio 1:
A) Bajar la imagen latest de Debian.

Con docker pull se pueden bajar imágenes que se encuentren en docker hub.

```
victor@PROMETHEUS:~/docker/ControlDocker$ docker pull debian:latest
latest: Pulling from library/debian
bc0734b949dc: Pull complete
Digest: sha256:bac353db4cc04bc672b14029964e686cd7bad56fe34b51f432c1a1304b9928da
Status: Downloaded newer image for debian:latest
docker.io/library/debian:latest
```

B) Ejecutar en modo interactivo un terminal root en la imagen Debian previamente descargada donde instalaremos nmap.

Con docker run -it iniciamos la terminal y le indicamos la imagen que vamos a usar, y por último, /bin/bash para indicar que vamos a iniciar como root.

```
victor@PROMETHEUS:~/docker/ControlDocker$ docker run -it debian:latest /bin/bash
root@eda82d5dedce:/# apt-get update
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [52.1 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8787 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [11.3 kB]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [128 kB]
Fetched 9177 kB in 1s (6163 kB/s)
Reading package lists... Done
root@eda82d5dedce:/# apt-get install -y nmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dbus dbus-bin dbus-daemon dbus-session-bus-common dbus-system-bus-common libapparmor1 libblas3 libdbus-1-3
  libexpat1 liblinear4 liblua5.3-0 libpcap0.8 libpcre3 libssh2-1 libssl3 lua-lpeg nmap-common
Suggested packages:
  default-dbus-session-bus | dbus-session-bus liblinear-tools liblinear-dev ncat ndiff zenmap
The following NEW packages will be installed:
  dbus dbus-bin dbus-daemon dbus-session-bus-common dbus-system-bus-common libapparmor1 libblas3 libdbus-1-3
  libexpat1 liblinear4 liblua5.3-0 libpcap0.8 libpcre3 libssh2-1 libssl3 lua-lpeg nmap nmap-common
0 upgraded, 18 newly installed, 0 to remove and 0 not upgraded.
Need to get 9979 kB of archives.
After this operation, 37.2 MB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 libdbus-1-3 amd64 1.14.10-1~deb12u1 [201 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 dbus-bin amd64 1.14.10-1~deb12u1 [105 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 dbus-session-bus-common all 1.14.10-1~deb12u1 [78.2 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 libapparmor1 amd64 3.0.8-3 [41.2 kB]
Get:5 http://deb.debian.org/debian bookworm/main amd64 libexpat1 amd64 2.5.0-1 [99.3 kB]
Get:6 http://deb.debian.org/debian bookworm/main amd64 dbus-daemon amd64 1.14.10-1~deb12u1 [184 kB]
Get:7 http://deb.debian.org/debian bookworm/main amd64 dbus-system-bus-common all 1.14.10-1~deb12u1 [79.3 kB]
Get:8 http://deb.debian.org/debian bookworm/main amd64 dbus amd64 1.14.10-1~deb12u1 [97.4 kB]
```

C) Eliminar el contenedor y la imagen descargada en los pasos a y b.

Con docker rm se pueden eliminar contenedores que no estén en uso y por docker image rm se pueden eliminar imágenes las cuáles no estén siendo usadas por un contenedor.

```
victor@PROMETHEUS:~/docker/ControlDocker$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS                       PORTS
                     NAMES
eda82d5dedce   debian:latest    "/bin/bash"              2 minutes ago   Exited (127) 26 seconds ago
                     kind_rubin
a86dffee004e   practica27-app   "java -jar /app.jar"     2 weeks ago     Exited (255) 13 days ago     0.0.0.0:8069->808
0/tcp, :::8069->8080/tcp   practica27-app-1
victor@PROMETHEUS:~/docker/ControlDocker$ docker images
REPOSITORY              TAG           IMAGE ID       CREATED       SIZE
debian                  latest        2a033a8c6371   2 hours ago   117MB
practica27-app          latest        4886c14a85fa   2 weeks ago   339MB
efriit/practica27-app   v1            4886c14a85fa   2 weeks ago   339MB
eclipse-temurin         21-jre-jammy  62416ec91e3a   2 weeks ago   292MB
victor@PROMETHEUS:~/docker/ControlDocker$ docker rm eda
eda
victor@PROMETHEUS:~/docker/ControlDocker$ docker image rm debian:latest
Untagged: debian:latest
Untagged: debian@sha256:bac353db4cc04bc672b14029964e686cd7bad56fe34b51f432c1a1304b9928da
Deleted: sha256:2a033a8c63712da54b5a516f5d69d41606cfb5c4ce9aa1690ee55fc4f9babb92
Deleted: sha256:ae134c61b154341a1dd932bd88cb44e805837508284e5d60ead8e94519eb339f
victor@PROMETHEUS:~/docker/ControlDocker$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED       STATUS                     PORTS
                    NAMES
a86dffee004e   practica27-app   "java -jar /app.jar"     2 weeks ago   Exited (255) 13 days ago   0.0.0.0:8069->8080/tcp
, :::8069->8080/tcp   practica27-app-1
victor@PROMETHEUS:~/docker/ControlDocker$ docker images
REPOSITORY              TAG           IMAGE ID       CREATED       SIZE
efriit/practica27-app   v1            4886c14a85fa   2 weeks ago   339MB
practica27-app          latest        4886c14a85fa   2 weeks ago   339MB
eclipse-temurin         21-jre-jammy  62416ec91e3a   2 weeks ago   292MB
victor@PROMETHEUS:~/docker/ControlDocker$
```

D) Crear un volumen interno de Docker llamado DatosExamen. Otro volumen interno llamado HTMLExamen. Muestra las características de los dos mediante los comandos necesarios

Con docker volume create se crean los volumenes y para mostrar sus características se usa docker volume inspect.

```
victor@PROMETHEUS:~/docker/ControlDocker$ docker volume create DatosExamen
DatosExamen
victor@PROMETHEUS:~/docker/ControlDocker$ docker volume create HTMLExamen
HTMLExamen
```

```
victor@PROMETHEUS:~/docker/ControlDocker$ docker volume inspect DatosExamen
[
    {
        "CreatedAt": "2023-12-19T03:59:02+01:00",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/snap/docker/common/var-lib-docker/volumes/DatosExamen/_data",
        "Name": "DatosExamen",
        "Options": {},
        "Scope": "local"
    }
]
victor@PROMETHEUS:~/docker/ControlDocker$ docker volume inspect HTMLExamen
[
    {
        "CreatedAt": "2023-12-19T03:59:12+01:00",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/snap/docker/common/var-lib-docker/volumes/HTMLExamen/_data",
        "Name": "HTMLExamen",
        "Options": {},
        "Scope": "local"
    }
]
```

Ejercicio 2. Realiza un cluster-nodo con docker compose y dos contenedores: uno con la imagen de node última y otra con la imagen mysql última.

```yaml
version: "3.8"
services:
  mysql:
    image: mysql
    restart: unless-stopped
    environment:
      MYSQL_DATABASE: appdb
      MYSQL_ROOT_PASSWORD: control1234
      MYSQL_USER: nodeUser
      MYSQL_PASSWORD: nodePassword
      MYSQL_PORT: 3310
    volumes:
      - "./data:/var/lib/mysql"
    ports:
      - "3310:3310"
  node:
    depends_on:
      - mysql
    image: node
    user: node
    restart: unless-stopped
    ports:
      - "8080:3000"
    environment:
      DB_HOST: mysql
      DB_USER: nodeUser
      DB_PASSWORD: nodePassword
      DB_NAME: appdb
      DB_PORT: localhost:3310
    volumes:
      - "./www:/home/node/app"
    stdin_open: true
    tty: true
```