Ejercicios Docker

```
Ejercicios Docker
Redes
```

Realizada por Daniel Pérez

Redes

Lo primero es crear una red bridge que tenga de nombre redbd para que se puedan conectar a ella los contenedores correspondientes:

```
docker network create redbd
```

Tenemos descargado la imagen MariaDB así que solo crearemos el contenedor:

```
docker run --name bbddredes --network redbd -v
/home/docker/datadir:/var/lib/mysql -e
MARIADB_ROOT_PASSWORD=root -p 3306:3306 -d mariadb
```

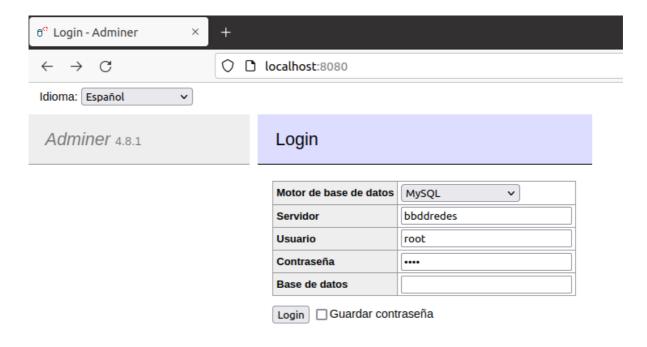
Tras crear el contenedor de base de datos ahora hay que usar el de Adminer para que se conecte a la base de datos:

```
docker run --name adminer --network redbd --link
bbddredes:mariadb -p 8080:8080 -d adminer
```

Comprobamos que los contenedores están corriendo:

```
| CREATED | STATUS | PORTS | P
```

Accedemos a Adminer desde localhost:8080:



Procedemos a crear una base de datos mediante Adminer:



Para comprobar que se ha creado accedemos desde consola usando el siguiente comando para acceder al contenedor interactivamente con la consola bash y el de después para ver la efectiva creación de la base de datos prueba:

```
docker exec -itm bbddredes bash
show databases;
```

Para terminar borramos los contenedores y los volúmenes:

```
docker rm bbddred
docker rm adminer
docker volume ls
docker volume prune
```

```
ocker@docker-VirtualBox:~$ docker rm bbddredes
bbddredes
docker@docker-VirtualBox:~$ docker rm adminer
adminer
docker@docker-VirtualBox:~$ docker volume ls
DRIVER VOLUME NAME
local
        9d6b8f3715a460325789111b11eb21d789b9ffe58fdef05fbf71ce24d22fcd46
local
        81cf2850de8ad83ad390c07547a1fa2904152b94c4d504de8e8ea6ae573a19f0
local
        235e5755469704e3aea3bdc0dd9990d7c54baa64941fa219039d8c24d074fb9b
local
         3587a702db242f1c1a0eb757e24b584c300a7391da701e8e68996df99380e45e
        96393fec7b9dc2633356e9d96cd4c37b7af595b62047b9dc165c635a4fdd2f87
local
        a1cf03a1459229c096cfe3d86c079cc65ec06a07d2e0ace87cb2fe25fc2aaae0
local
         a23adba5b80361d654ad91839373482476cfaddd310c7862a508427ab52c188a
local
        b3006563dc6e44d986b69bf2155b947f9c8bc81942ff6d8e490e3fbdb04689f9
local
```

```
docker@docker-VirtualBox:~$ docker volume prune
WARNING! This will remove all local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
3587a702db242f1c1a0eb757e24b584c300a7391da701e8e68996df99380e45e
81cf2850de8ad83ad390c07547a1fa2904152b94c4d504de8e8ea6ae573a19f0
96393fec7b9dc26333356e9d96cd4c37b7af595b62047b9dc165c635a4fdd2f87
9d6b8f3715a460325789111b11eb21d789b9ffe58fdef05fbf71ce24d22fcd46
a1cf03a1459229c096cfe3d86c079cc65ec06a07d2e0ace87cb2fe25fc2aaae0
a23adba5b80361d654ad91839373482476cfaddd310c7862a508427ab52c188a
b3006563dc6e44d986b69bf2155b947f9c8bc81942ff6d8e490e3fbdb04689f9
235e5755469704e3aea3bdc0dd9990d7c54baa64941fa219039d8c24d074fb9b
Total reclaimed space: 640.1MB
```

Con docker volume prune borramos todos los volúmenes que no están siendo usados (como habíamos creado otros anteriores al crear contenedores y posteriormente borrarlos, el brune también los ha eliminado).