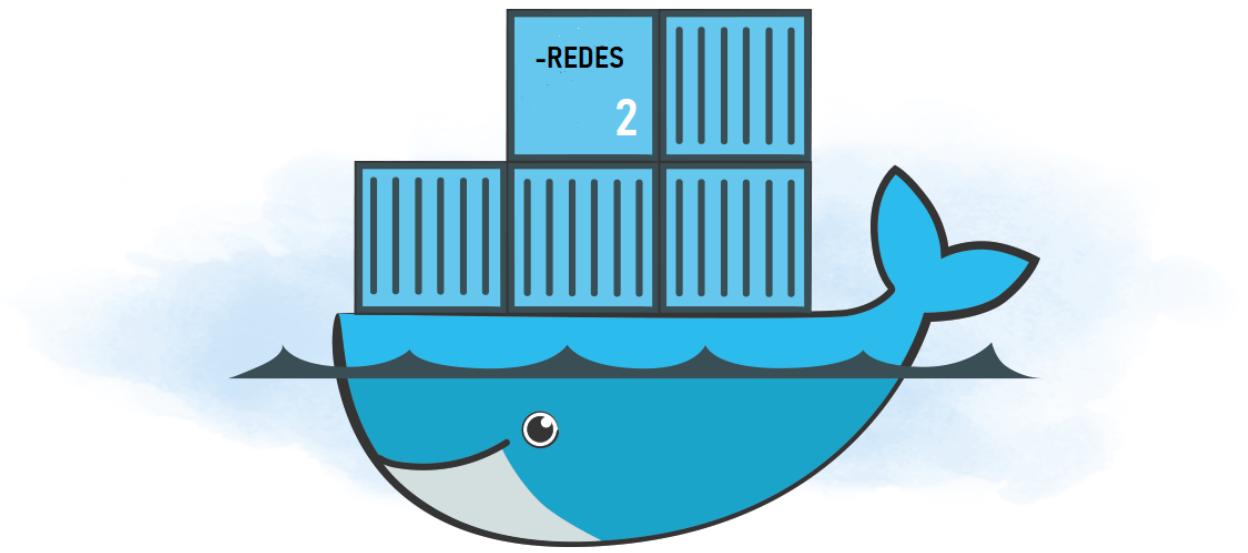


Actividad evaluable 3 - Docker - 2ª Evaluación

Ejercicio 2 - Redes y almacenamiento

- Apartado 1- **Despliegue de contenedores en red: Adminer y MariaDB**

Ejercicio 2 - Redes y almacenamiento

1.- Despliegue de contenedores en red: Adminer y MariaDB

Para esta actividad vamos a trabajar con redes virtuales y almacenamiento. Pasamos a comentar la solución.

- Es importante recordar, que cada vez que creamos un contenedor, este se conecta a *red virtual* y Docker hace una configuración del sistema para que la máquina tenga una ip interna, teniendo así acceso al exterior y podamos mapear. Usa estas tres redes predefinidas: *red tipo bridge, red tipo host, red tipo none.*

- Podemos ver las redes que tenemos cuando hemos instalado Docker.

```
docker network ls
```

```
anacachafeiro@clientelinux:~$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
aa517325e3a8   bridge    bridge    local
48cf0dade1b5   host      host      local
beb867b8b2ac   none      null      local
```

- Comenzamos creando una *red bridge bdnet*

```
docker network create bdnet
docker network ls
```

```
anacachafeiro@clientelinux:~$ docker network create bdnet
docker network ls
0bb52a842023d397725fe802fae2c536387ef2ad131b37286ef70b5b84c218e4
NETWORK ID     NAME      DRIVER    SCOPE
0bb52a842023   bdnet     bridge    local
aa517325e3a8   bridge    bridge    local
48cf0dade1b5   host      host      local
beb867b8b2ac   none      null      local
```

- Continuamos creando un contenedor con la imagen de *mariadb* que nos hemos descargado de Docker Hub y que estará en la *red bdnet* y con las siguientes especificaciones. Así como un volumen de datos persistente.

```
docker pull mariadb
```

```
anacachafeiro@clientelinux:~$ docker pull mariadb
Using default tag: latest
latest: Pulling from library/mariadb
Digest: sha256:9ff479f244cc596aed9794d035a9f352662f2caed933238c533024df64569853
Status: Image is up to date for mariadb:latest
docker.io/library/mariadb:latest
```

```
docker volume create dataDB
docker network ls
```

```
anacachafeiro@clientelinux:~$ docker volume create dataDB
docker volume ls
dataDB
DRIVER    VOLUME NAME
local     69c031ec88599b3119d1a625bd55f066efb667eb0e92affcfc189c1372dac1f5
local     4414cff35538d0aa5d83f6e9fefe6d974985e44167d8ecc1fefe3995fd6e055a
local     dataDB
```

```
docker run -d --network bdnet --name mibase -e MYSQL_ROOT_PASSWORD=root -v
dataDB:/var/lib/mysql mariadb
```

```
anacachafeiro@clientelinux:~$ docker run -d --network bdnet --name mibase -e M
YSQL_ROOT_PASSWORD=root -v dataDB:/var/lib/mysql mariadb
d962d6e669abdcde4fa73d70b7265f084c9770d509625c7474b0b53bb56bc1db
```

- El siguiente paso será crear un contenedor con el programa *Adminer* que se conecte al contenedor de la base de datos.

```
docker run -d --network bdnet --name miadminer -p 8080:8080 -e
ADMINER_DEFAULT_SERVER=myDB adminer
```

```
anacachafeiro@clientelinux:~$ docker run -d --network bdnet --name miadminer -
p 8080:8080 -e ADMINER_DEFAULT_SERVER=myDB adminer
Unable to find image 'adminer:latest' locally
latest: Pulling from library/adminer
3e440a704568: Pull complete
8d6386bc062c: Pull complete
5dbc633dab93: Pull complete
39317196bba2: Pull complete
31ea8e1da1f9: Pull complete
d3ce0ac05636: Pull complete
ac3ee3b23021: Pull complete
Digest: sha256:4203fd6bcd82fe25dceaf8acb4826129cf7a6e93b22a5ab2fc0ec5a7cdaca3f
4
Status: Downloaded newer image for adminer:latest
d9579883cdfb39b48380e8e465fd5c3141280cac3efd9df0e1ca1397af8d9626
```

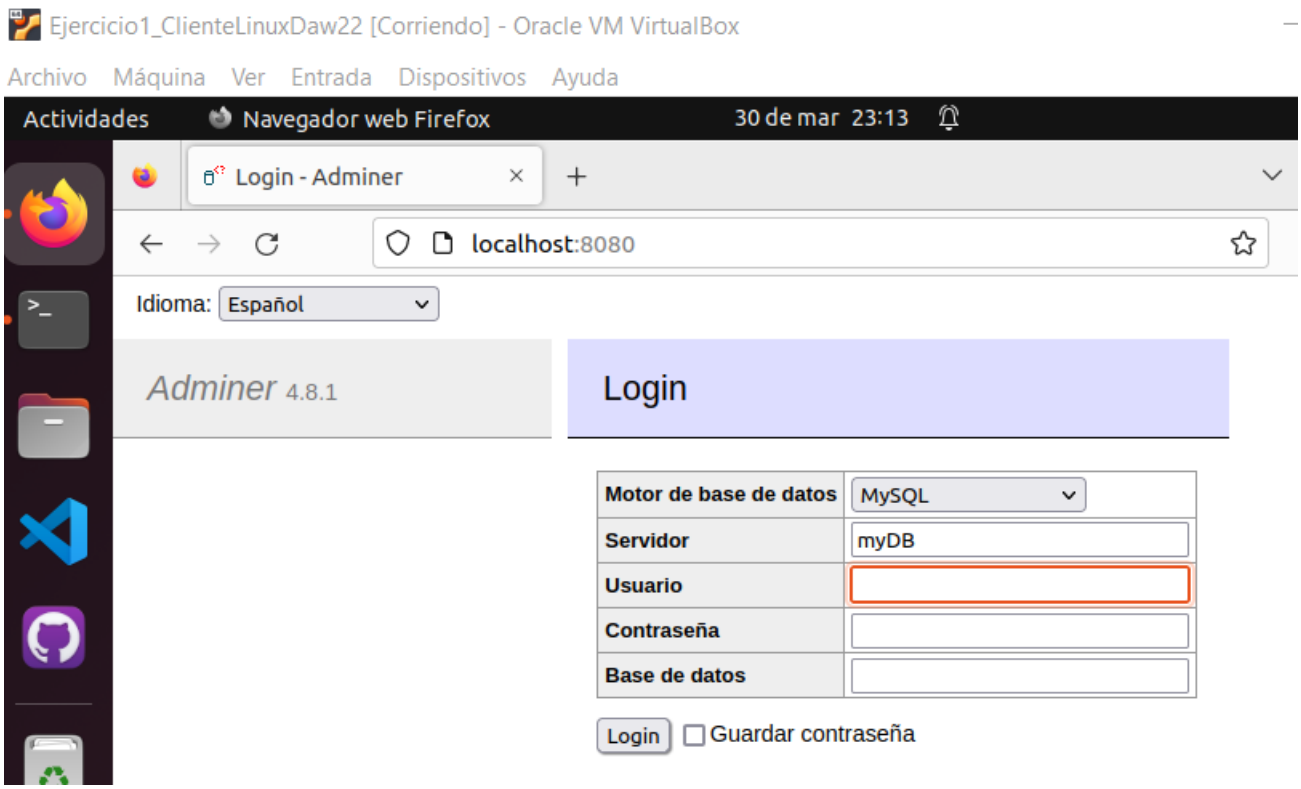
- Mostramos contenedores creados y funcionando.

```
docker ps -a
```

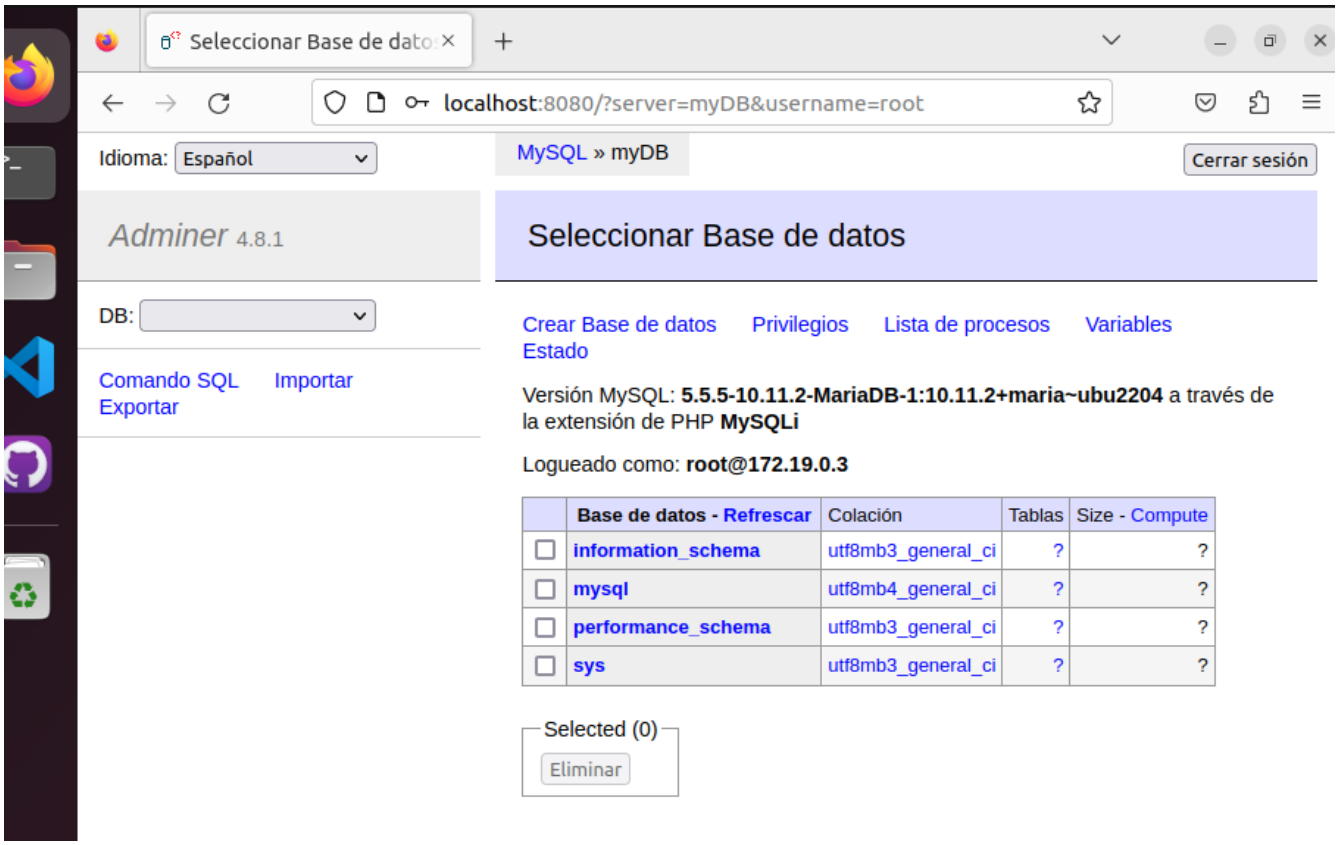
```
anacachafeiro@clientelinux:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS
  PORTS                                     NAMES
d9579883cdfb   adminer   "entrypoint.sh php -…"   About a minute ago   Up About a minute
  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   miadminer
d962d6e669ab   mariadb   "docker-entrypoint.s…"   7 minutes ago        Up 7 minutes
  3306/tcp                                    mibase
```

- Vamos al navegador, introducimos la url: *http://localhost:8080*

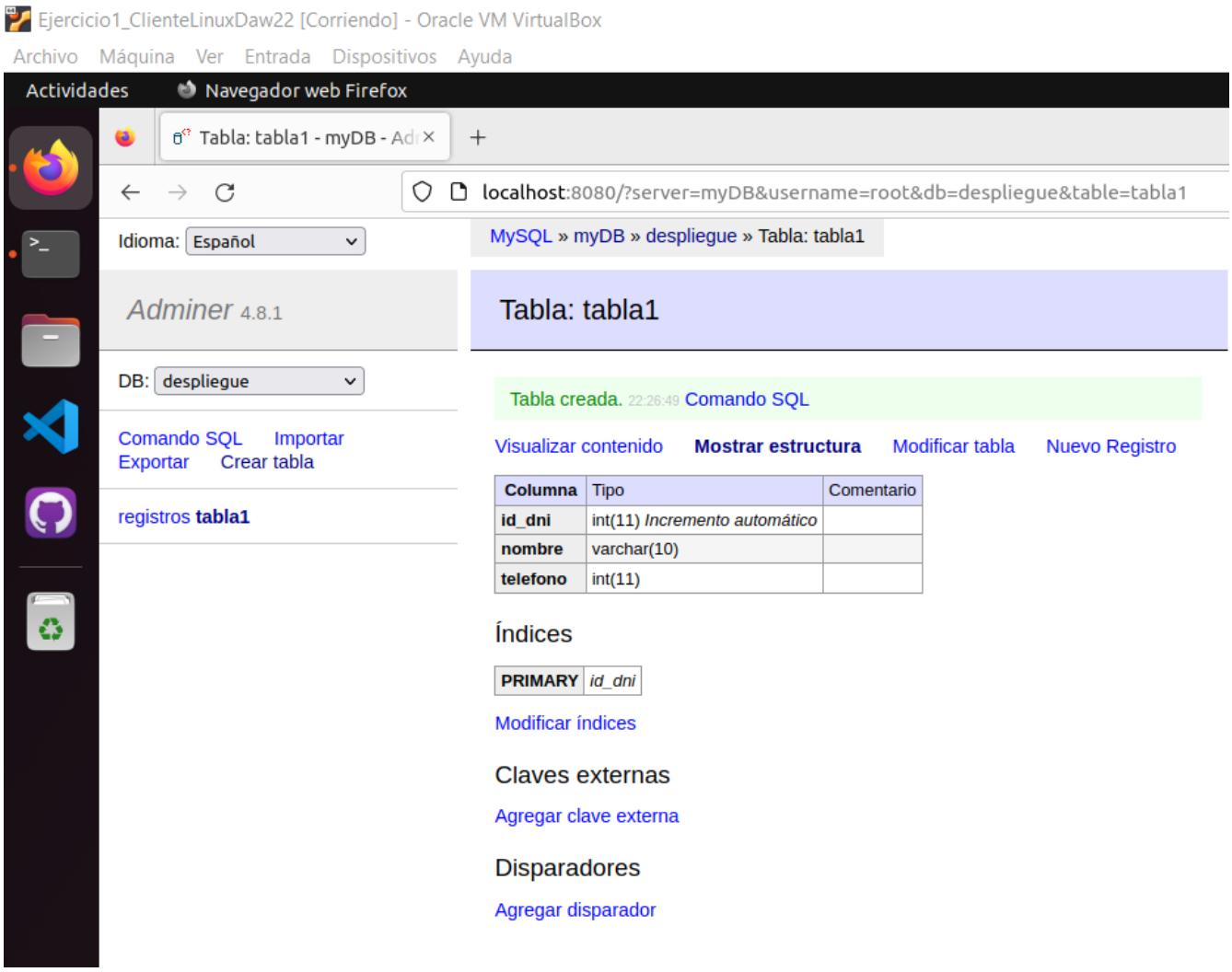
```
http://localhost:8080
```



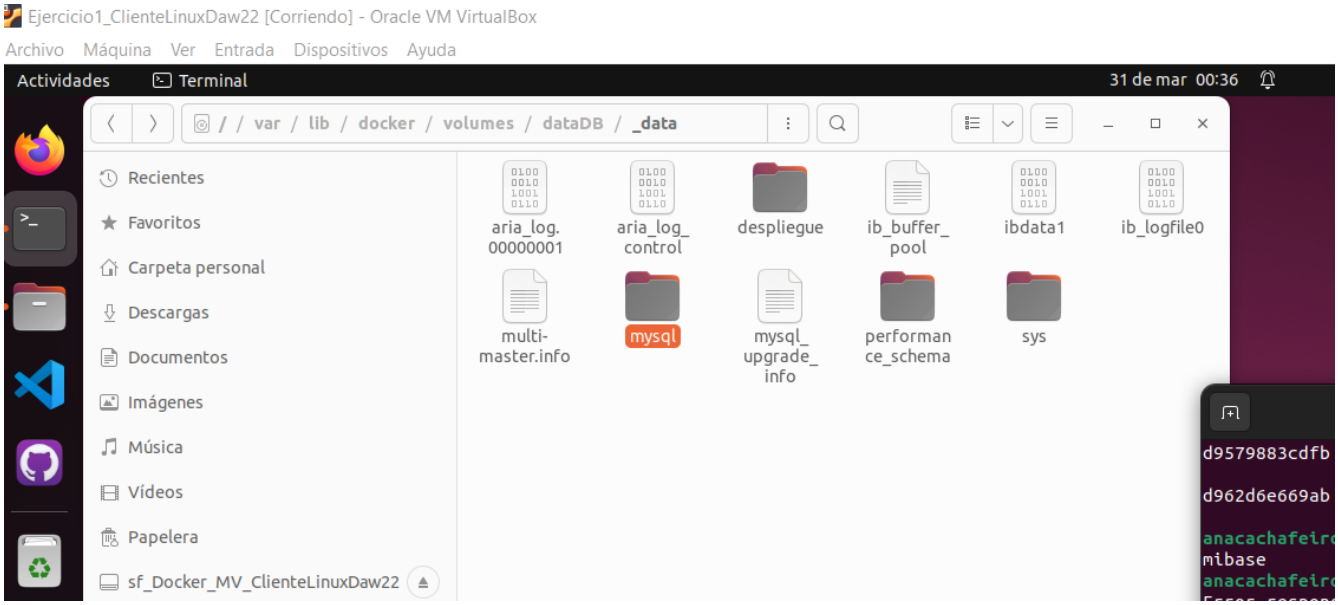
- Introducimos el usuario (root) y la contraseña (root)

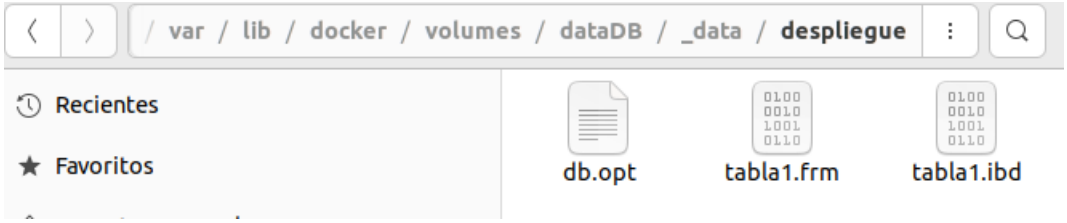


- Una vez dentro de Adminer creamos una base de datos llamada *despliegue*. Aprovechamos para crear una tabla con datos varios.



- Ahora vamos a la ruta */var/lib/docker/volumenes/dataDB/_data*. Comprobamos los datos generados por el contenedor del servidor de base de datos. Confirmamos la base de datos *despliegue* y la *tabla1*.





- Finalizamos la tarea borrando por este orden: los volúmenes, los contenedores y la red creada.

```
docker volume prune
```

```
anacachafeiro@clientelinux:~$ 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:
4414cff35538d0aa5d83f6e9fefe6d974985e44167d8ecc1fefe3995fd6e055a
69c031ec88599b3119d1a625bd55f066efb667eb0e92affcfc189c1372dac1f5

Total reclaimed space: 154.8MB
anacachafeiro@clientelinux:~$ docker volume ls
DRIVER    VOLUME NAME
local     dataDB
```

```
docker rm -f miadminer
```

```
docker rm -f mibase
```

```
anacachafeiro@clientelinux:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED       STATUS                          PORTS     NAMES
d9579883cdfb   adminer   "entrypoint.sh php -…"   2 hours ago   Exited (128) 51 minutes ago               miadminer
d962d6e669ab   mariadb   "docker-entrypoint.s…"   2 hours ago   Exited (1) 27 minutes ago                 mibase
anacachafeiro@clientelinux:~$ docker rm -f miadminer
miadminer
anacachafeiro@clientelinux:~$ docker rm -f mibase
mibase
anacachafeiro@clientelinux:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
anacachafeiro@clientelinux:~$
```

```
docker network rm bdnet
```

```
anacachafeiro@clientelinux:~$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
0bb52a842023   bdnet     bridge    local
e1faf3964778   bridge    bridge    local
48cf0dade1b5   host      host      local
beb867b8b2ac   none      null      local
anacachafeiro@clientelinux:~$ docker network rm bdnet
bdnet
anacachafeiro@clientelinux:~$ dcoker network ls
Orden «dcoker» no encontrada. Quizá quiso decir:
  la orden «docker» del paquete snap «docker (20.10.17)»
  la orden «docker» del paquete deb «podman-docker (3.4.4+ds1-1ubuntu1)»
  la orden «docker» del paquete deb «docker.io (20.10.21-0ubuntu1~22.04.2)»
Consulte «snap info <nombre del snap>» para ver más versiones.
anacachafeiro@clientelinux:~$ docker network ls

NETWORK ID     NAME      DRIVER    SCOPE
e1faf3964778   bridge    bridge    local
48cf0dade1b5   host      host      local
beb867b8b2ac   none      null      local
```

Webgrafía

- Adminer

https://kinsta.com/es/blog/adminer/

https://hub.docker.com/_/adminer/

- Apuntes segundo trimestre DAW
