

Ejercicio 3 - contenedores en red: Adminer y MariaDB

Tarea docker: Gerald y Sara

[Paso 1](#)

[Paso 2](#)

[Paso 3](#)

[Paso 4](#)

[Desde dbeaver](#)

[Desde adminer](#)

[Paso 5](#)

Paso 1

Crea una red bridge **redbd**

```
docker network create --driver bridge redbd
```

```
gerald@clientlinux ~$ docker network create --driver bridge redbd
28902e44aed52479088bc67d9403c6a7bb377ab8b7c148ca4c783cd231154186
```

Parámetros de la red **redbd**

```
docker network inspect redbd
```

```
gerald@clientlinux ~$ docker network inspect redbd
[
  {
    "Name": "redbd",
    "Id": "28902e44aed52479088bc67d9403c6a7bb377ab8b7c148ca4c783cd231154186",
    "Created": "2025-02-13T08:46:50.680733644+01:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.19.0.0/16",
          "Gateway": "172.19.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]
```

Paso 2

Crea un contenedor con una imagen de `mariadb` que estará en la red `redbd` .

Este

contenedor se ejecutará en segundo plano, y será accesible a través del puerto 3306. (Es

necesario definir la contraseña del usuario

`root` y un volumen de datos persistente)

```
docker run -d --name contenedor_mariadb --network redbd \
-e MYSQL_ROOT_PASSWORD=root \
-v mariadb_data:/var/lib/mysql \
```

```
-p 3306:3306 \
mariadb
```

```
gerald@clientlinux ~$ docker run -d --name contenedor_mariadb --network redbd -e MYSQL_ROOT_PASSWORD=root -v mariadb_data:/var/lib/mysql -p 3306:3306 mariadb
852b2875e754cd3efb93917d5ce9780ee9b82075be5130957830730b0ff1dc7e
gerald@clientlinux ~$
```

Paso 3

Crear un contenedor con **Adminer** o con **phpMyAdmin** que se pueda conectar al contenedor de la BD

```
docker run -d --name contenedor_adminer --network redbd -p 8080:8080 ad
```

```
gerald@clientlinux ~$ docker run -d --name contenedor_adminer --network redbd -p 8080:8080 adminer
Unable to find image 'adminer:latest' locally
latest: Pulling from library/adminer
73226dab8db5: Pull complete
ed94e1c95a57: Pull complete
884bce373183: Pull complete
9a4cd7b75371: Pull complete
574dfab7cda2: Pull complete
798a45c9628c: Pull complete
c82cd9b427d9: Pull complete
Digest: sha256:34d37131366c5aa84e1693dbed48593ed6f95fb450b576c1a7a59d3a9c9e8802
Status: Downloaded newer image for adminer:latest
4bafd9643eb5725d017a0781a6c0b576ac928c3b7d389594ebe9e047f0c4bc4c
```

Vemos que los dos contenedores están ejecutandose

```
docker ps
```

```
gerald@clientlinux ~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
4bafd9643eb5   adminer   "entrypoint.sh php -..." 49 seconds ago Up 48 seconds 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp contenedor_adminer
852b2875e754   mariadb   "docker-entrypoint.s..." 7 minutes ago  Up 7 minutes  0.0.0.0:3306->3306/tcp, :::3306->3306/tcp contenedor_mariadb
```

Paso 4

Desde dbeaver

Desde la interfaz gráfica, crear una base de datos y una tabla en el servidor de base de datos

Accedemos a la base de datos con la aplicación `dbeaver`

Conectar a base de datos

Connection Settings
MariaDB ajustes de conexión

General | Driver properties | SSH | SSL | + Network configurations...

Server
Connect by: ☒ Host ☐ URL
URL: jdbc:mariadb://localhost:3306/
Server Host: localhost Port: 3306
Database:

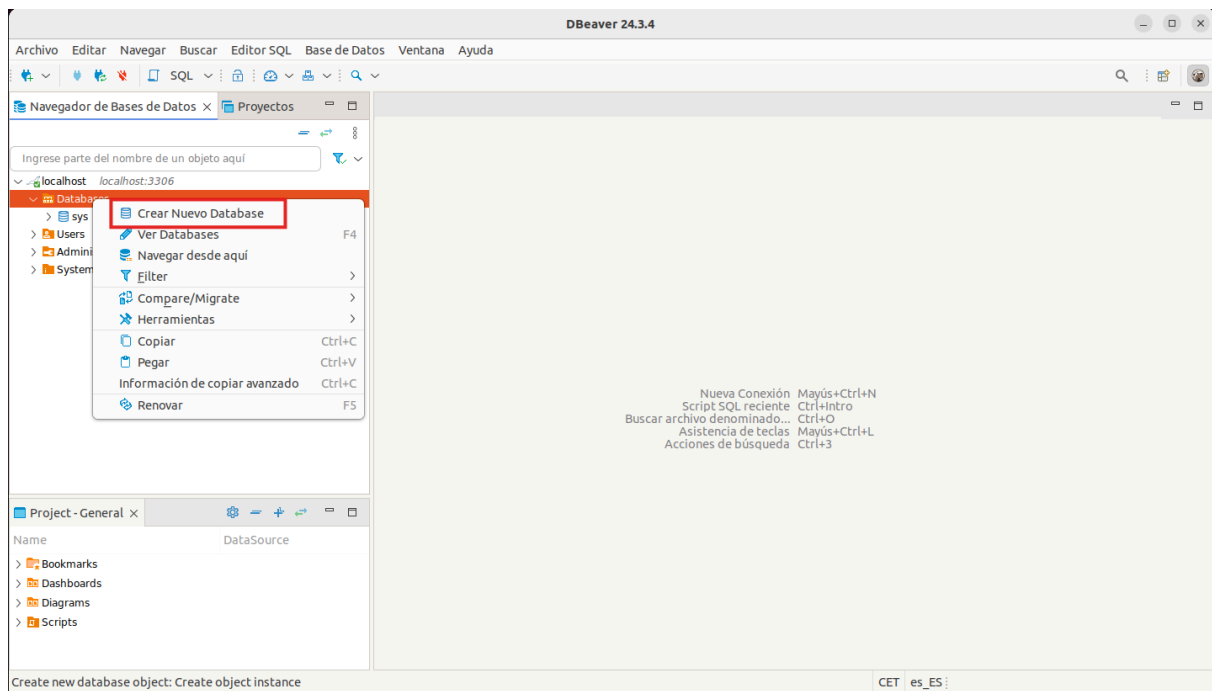
Authentication (Database Native)
Nombre de usuario: root
Contraseña: Save password

Advanced
Server Time Zone: Auto-detect
Local Client: <not present>

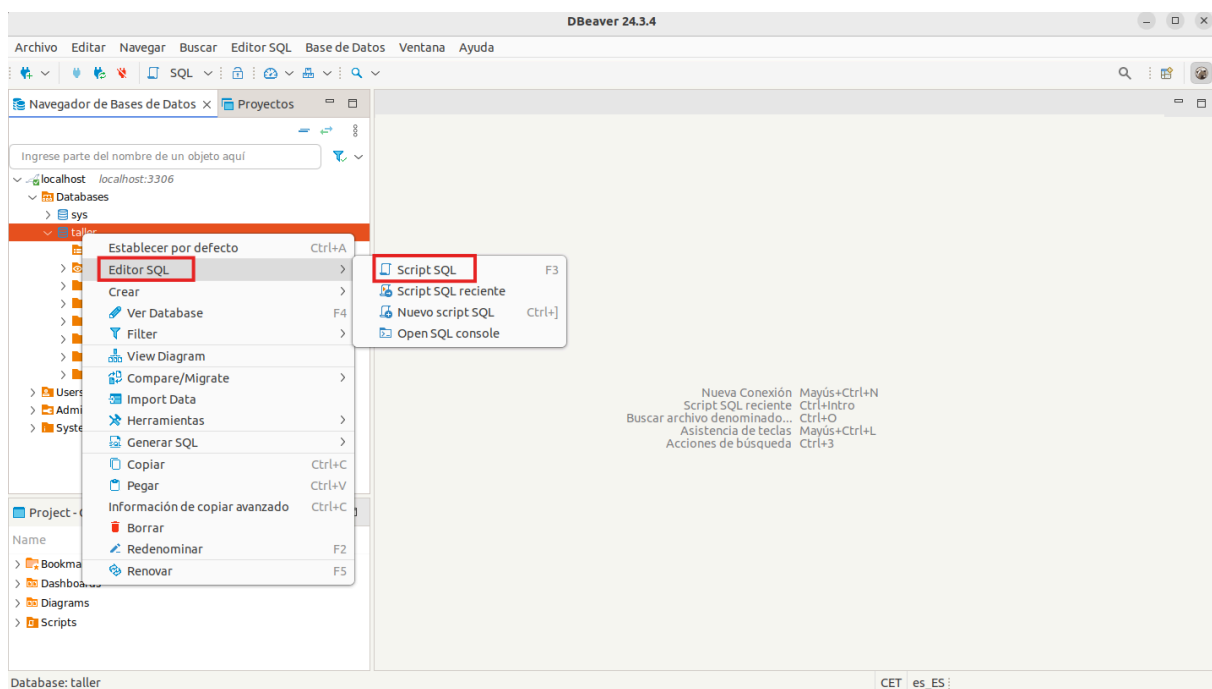
[Connection variables information](#) Connection details (name, type, ...)
Driver name: MariaDB Driver Settings Licencia del driver

Probar conexión ... ← Anterior Siguiendo → Cancelar Finalizar

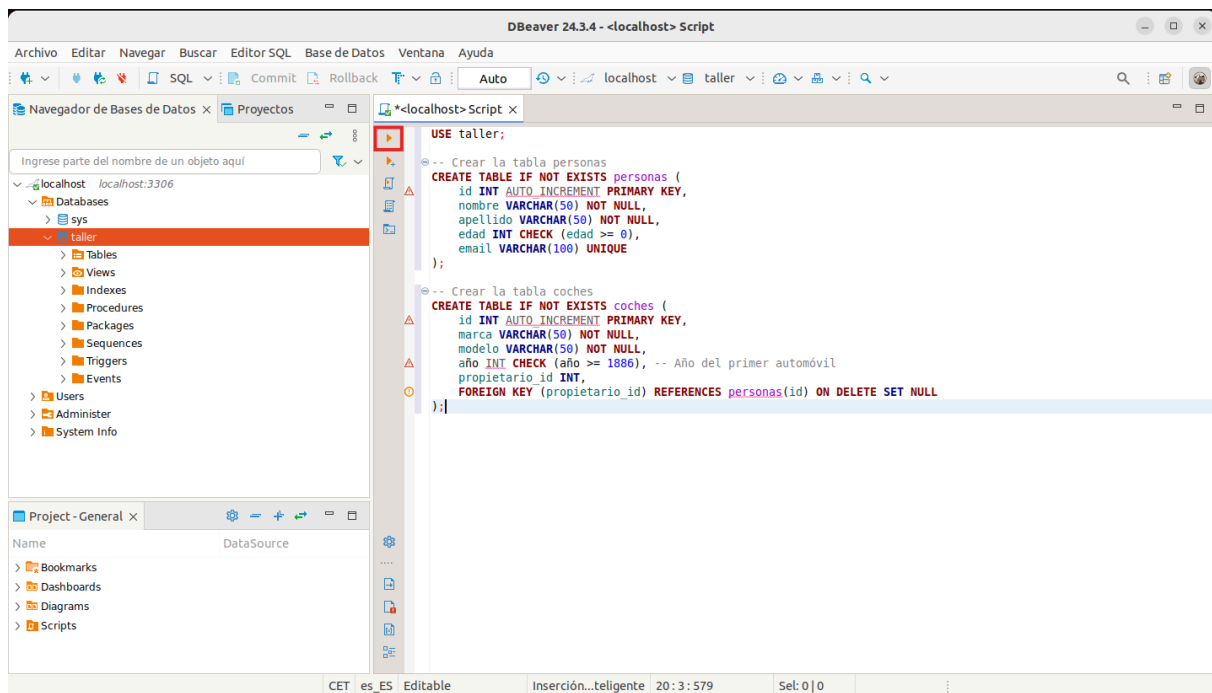
Vamos a crear un base de datos



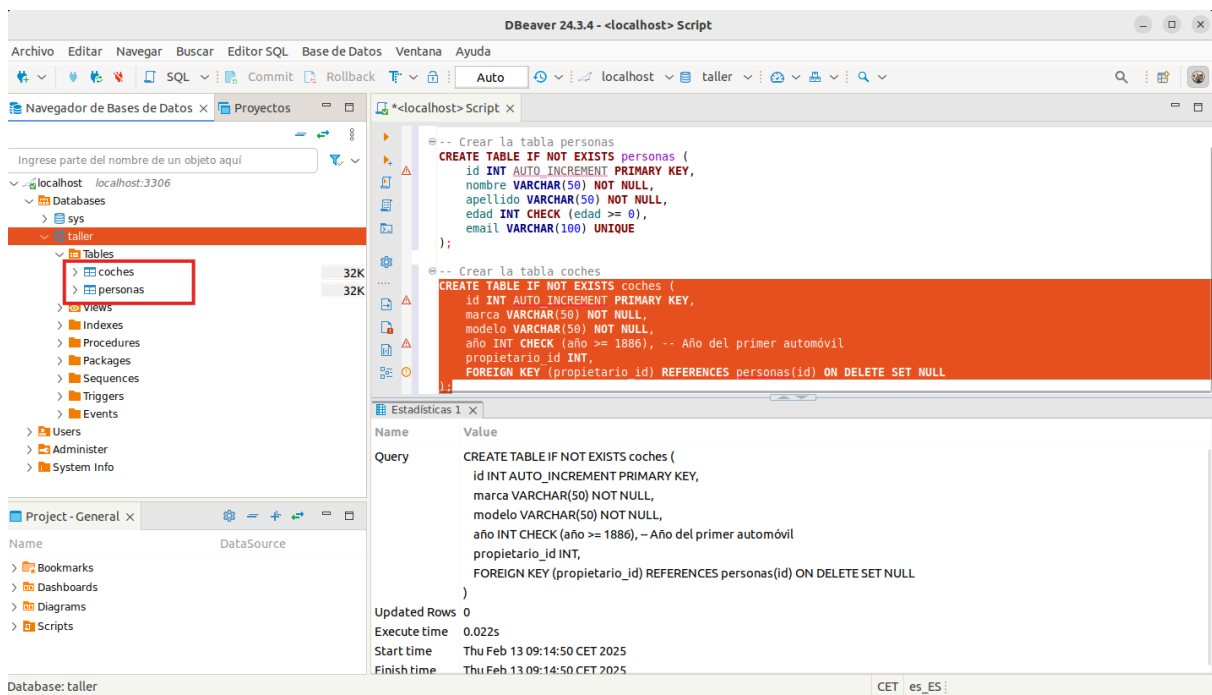
Vamos a crear la tabla **personas** y **coches** mediante un script en la base de datos
Para ello abrimos un script



Escribimos el script y le damos al botón de ejecutar



Vemos que se han creado las tablas



Desde adminer

Ahora vamos a conectar con adminer, para ello en un navegador escribimos

<http://localhost:8080>

Browser tabs: Login - Adminer

Address bar: localhost:8080

Idioma: Español

Adminer 4.8.1

Login

Motor de base de datos	MySQL
Servidor	db
Usuario	
Contraseña	
Base de datos	

☐ Guardar contraseña

Introducimos los datos para acceder en este caso a la base de datos creada previamente

Browser tabs: Login - db - Adminer

Address bar: localhost:8080/?server=db&username=root&db=pruebadminer

Idioma: Español

Adminer 4.8.1

Login

Motor de base de datos	MySQL
Servidor	contenedor_mariadb
Usuario	root
Contraseña
Base de datos	taller

☐ Guardar contraseña

Vemos las tablas creadas anteriormente

Base de datos: taller - con x +

localhost:8080/?server=contenedor_mariadb&username=root&db=taller

Idioma: Español MySQL » contenedor_mariadb » Base de datos: taller Cerrar sesión

Adminer 4.8.1

DB: taller

Comando SQL Importar Exportar Crear tabla

registros coches registros personas

Modificar Base de datos Esquema de base de datos Privilegios

Tablas y vistas

Buscar datos en tablas (2)

<input type="checkbox"/> Tabla	Motor?	Colación?	Longitud de datos?	Longitud de índice?	Espacio libre?	Incremento automático?	Registros?	Comentario?
<input type="checkbox"/> coches	InnoDB	utf8mb4_uca1400_ai_ci	16 384	16 384	0	1	0	
<input type="checkbox"/> personas	InnoDB	utf8mb4_uca1400_ai_ci	16 384	16 384	0	1	0	
2 en total	InnoDB	utf8mb4_uca1400_ai_ci	32 768	32 768	0			

Selected (0)

Analizar Optimizar Comprobar Reparar Vaciar Eliminar

Mover a otra base de datos: taller Mover Copiar ☐ overwrite

Crear tabla Crear vista

Procedimientos

Crear procedimiento Crear función

Eventos

Crear Evento

Vamos a crear una tabla nueva desde **adminer**

Para ello le damos en el botón **crear tabla** e introducimos el nombre de la tabla y los campos que queremos que contenga

Crear tabla - contenedor x +

localhost:8080/?server=contenedor_mariadb&username=root&db=taller&create=

Idioma: Español MySQL » contenedor_mariadb » taller » Crear tabla Cerrar sesión

Adminer 4.8.1

DB: taller

Comando SQL Importar Exportar **Crear tabla**

registros coches registros personas

Crear tabla

Nombre de la tabla: mecanicos (motor) (colación) Guardar

Nombre de columna	Tipo	Longitud	Opciones	NULL	AI?	+
dni	int			<input type="checkbox"/>	<input type="radio"/>	+ ↑ ↓ ×
nombre	varchar	20	(colación)	<input type="checkbox"/>	<input type="radio"/>	+ ↑ ↓ ×
telefono	int	9		<input type="checkbox"/>	<input type="radio"/>	+ ↑ ↓ ×
	int			<input type="checkbox"/>	<input type="radio"/>	+ ↑ ↓ ×

Incremento automático: ☐ Valores predeterminados ☐ Comentario

Guardar

Particionar por

Vemos como se ha creado la tabla **mecánicos**

Base de datos: taller - cor x

localhost:8080/?server=contenedor_mariadb&username=root&db=taller

Idioma: Español MySQL » contenedor_mariadb » Base de datos: taller Cerrar sesión

Adminer 4.8.1

DB: taller

Comando SQL Importar Exportar Crear tabla

registros coches
registros mecanicos
registros personas

Base de datos: taller

Modificar Base de datos Esquema de base de datos Privilegios

Tablas y vistas

Buscar datos en tablas (3)

Tabla	Motor	Colación	Longitud de datos	Longitud de índice	Espacio libre	Incremento automático	Registros	Comentario
<input type="checkbox"/> coches	InnoDB	utf8mb4_uca1400_ai_ci	16 384	16 384	0	1	0	
<input type="checkbox"/> mecanicos	InnoDB	utf8mb4_uca1400_ai_ci	16 384	0	0		0	
<input type="checkbox"/> personas	InnoDB	utf8mb4_uca1400_ai_ci	16 384	16 384	0		1	0
3 en total	InnoDB	utf8mb4_uca1400_ai_ci	49 152	32 768	0			

Selected (0)

Analizar Optimizar Comprobar Reparar Vaciar Eliminar

Mover a otra base de datos: taller Mover Copiar overwrite

Crear tabla Crear vista

Procedimientos

Crear procedimiento Crear función

Eventos

Crear Evento

Ahora vamos a crear una base de datos nueva, para ello pinchamos en el nombre del servidor, en este caso **contenedor_mariadb** y le damos al botón **crear base de datos**

Selecciónar Base de datos x

localhost:8080/?server=contenedor_mariadb&username=root

Idioma: Español MySQL » contenedor_mariadb

Adminer 4.8.1

DB:

Comando SQL Importar Exportar

Seleccionar Base de datos

Crear Base de datos Privilegios Lista de procesos Variables Estado

Versión MySQL: 11.6.2-MariaDB-ubu2404 a través de la extensión de PHP MySQLi

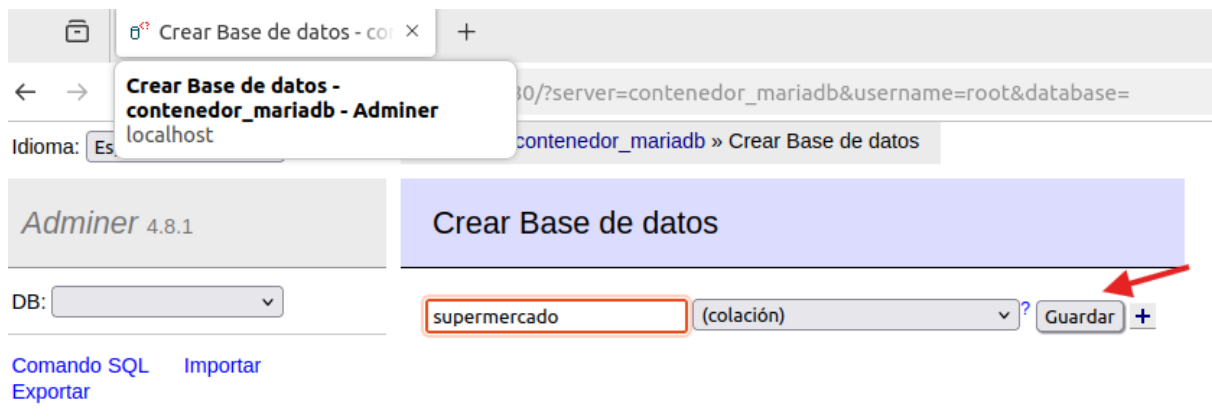
Logueado como: root@172.19.0.3

Base de datos - Refrescar	Colación	Tablas	Size - Compute
<input type="checkbox"/> information_schema	utf8mb3_general_ci	?	?
<input type="checkbox"/> mysql	utf8mb4_uca1400_ai_ci	?	?
<input type="checkbox"/> performance_schema	utf8mb3_general_ci	?	?
<input type="checkbox"/> sys	utf8mb3_general_ci	?	?
<input type="checkbox"/> taller	utf8mb4_uca1400_ai_ci	?	?

Selected (0)

Eliminar

Introducimos el nombre de la base de datos y le damos a guardar



Vemos como se ha creado



Paso 5

Borrar los contenedores, la red y los volúmenes utilizados - desde comando o utilizando

Docker Desktop

Vamos a empezar borrando los contenedores, para ello primero vamos a tener que pararlos

```
docker stop contenedor_adminer
docker stop contenedor_mariadb
```

```
gerald@clientlinux ~$ docker stop contenedor_adminer
contenedor_adminer
gerald@clientlinux ~$ docker stop contenedor_mariadb
contenedor_mariadb
```

Ahora ya podemos borrarlos

```
docker rm contenedor_adminer
docker rm contenedor_mariadb
```

```
gerald@clientlinux ~$ docker rm contenedor_adminer
contenedor_adminer
gerald@clientlinux ~$ docker rm contenedor_mariadb
contenedor_mariadb
```

Comprobamos que se han borrado los contenedores

```
docker ps -a
```

```
gerald@clientlinux ~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
98113e77ca1c   php:7.4-apache  "docker-php-entrypoi..." 13 days ago   Exited (255) 8 days ago           0.0.0.0:80->80/tcp, :::80->80/tcp   bindmount_php
9167294e13bb   mariadb        "docker-entrypoint.s..." 2 weeks ago   Exited (255) 8 days ago           0.0.0.0:3336->3306/tcp, [::]:3336->3306/tcp   bbdd
44ab1aa951d6   php:7.4-apache  "docker-php-entrypoi..." 2 weeks ago   Exited (255) 13 days ago          0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   contenedor_demonio
57af7b16970e   nextcloud      "/entrypoint.sh apac..." 3 weeks ago   Exited (255) 2 weeks ago          0.0.0.0:8082->80/tcp, [::]:8082->80/tcp   nextcloud
145d5e57679d   nginx         "/docker-entrypoint...." 3 weeks ago   Exited (255) 3 weeks ago          0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   nginx-server
24f5fa76b1c2   debian        "/bin/bash"              3 weeks ago   Exited (127) 3 weeks ago                                     elastic_aryabhata
9e36663c111f   fedora        "bash"                   3 weeks ago   Exited (137) 3 weeks ago                                     trusting_ramanujan
e8d250be1db0   ubuntu       "/bin/bash"              3 weeks ago   Exited (130) 3 weeks ago                                     objective_elion
8084dc7f8d5a   busybox      "sh"                     3 weeks ago   Exited (0) 3 weeks ago                                     focused_dubinsky
ed9a38f5c49b   hello-world   "/hello"                 3 weeks ago   Exited (0) 3 weeks ago                                     friendly_hellman
89410034ffac   ubuntu       "bash"                   3 weeks ago   Exited (0) 3 weeks ago                                     mi_ubuntu
5ab32356f558   mariadb:10.5  "docker-entrypoint.s..." 3 weeks ago   Exited (255) 3 weeks ago          0.0.0.0:3306->3306/tcp, :::3306->3306/tcp   mimariadb
8d45b87d30aa   httpd:2.4     "httpd-foreground"       3 weeks ago   Exited (255) 3 weeks ago          0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   my-apache-app
8d5944c7d2b7   ubuntu       "bash -c 'while true..." 3 weeks ago   Exited (255) 3 weeks ago                                     hora-container2
99e40a3d8ece   ubuntu       "bash -c 'while true..." 3 weeks ago   Exited (137) 3 weeks ago                                     contenedor5
5795cebeab3f   ubuntu       "bash"                   4 weeks ago   Exited (255) 3 weeks ago                                     contenedor2
7bf387cec76a   ubuntu       "bash"                   4 weeks ago   Exited (127) 4 weeks ago                                     contenedor1
59ff16dc23dc   ubuntu       "hostname contenedor..." 4 weeks ago   Exited (1) 4 weeks ago                                     quizzical_ptolemy
83343096e824   ubuntu       "echo 'Hello world'"     4 weeks ago   Exited (0) 4 weeks ago                                     contenedor_ubuntu
870680fb430   ubuntu       "echo 'Hello world'"     4 weeks ago   Exited (0) 4 weeks ago                                     jolly_tharp
51178db562fd   hello-world   "/hello"                 4 weeks ago   Exited (0) 4 weeks ago
```

Vemos que ya no aparecen

Ahora vamos a eliminar la red

```
docker network rm redbd
```

```
gerald@clientlinux ~$ docker network rm redbd  
redbd
```

Comprobamos que se ha borrado la red

```
docker network ls
```

```
gerald@clientlinux ~$ docker network ls  
NETWORK ID      NAME      DRIVER      SCOPE  
d28c0940818b    bridge    bridge      local  
8c688f8cbbc5    host      host        local  
efbe511b3a2c    none      null        local
```

Por último eliminamos el volumen

```
docker volume rm mariadb_data
```

```
gerald@clientlinux ~$ docker volume rm mariadb_data  
mariadb_data
```

Comprobamos que se ha eliminado el volumen

```
docker volume ls
```

```
gerald@clientlinux ~$ docker volume ls
DRIVER      VOLUME NAME
local       35d6705e1c9edb93aa9e150bf5bab404cfb8b2ce031694f77adc53b887c24de5
local       a8c712434eca3a053aac57ac4a9bcf1ad040144e2364030cc2fb861b2cec9084
local       f9c2cf5c5b6522e37ac9dc6717f9f938e8cfd0c66b106f194de6c7e3b79b5ee3
local       miweb
local       nextcloud-data
```