

Actividad 3 Volúmenes NextCloud y MariaDb

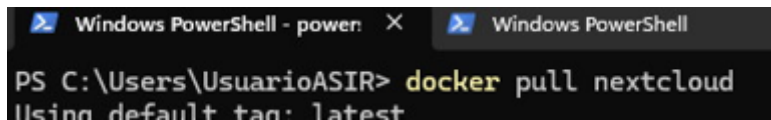
Cristóbal Suárez Abad
OPTATIVA – 2º ASIR

NextCloud

Se desea desplegar un servicio de **almacenamiento en la nube** para un pequeño centro educativo utilizando Nextcloud

1. Descarga la **imagen oficial de Nextcloud** desde Docker Hub.
2. Crea un **contenedor** llamado nextcloud-server a partir de esa imagen.
3. Configura un **volumen persistente** para que los datos almacenados por los usuarios (archivos subidos) se mantengan, aunque el contenedor se detenga o elimine.
4. El servicio debe estar disponible en el **puerto 8080** de la máquina anfitriona.
5. Comprueba desde el navegador que la interfaz web de Nextcloud está disponible y añade algún fichero de prueba.
6. Elimina el contenedor y vuélvelo a crear para comprobar que el archivo sigue existiendo en la nube.

Primero descargamos la imagen: **docker pull nextcloud**



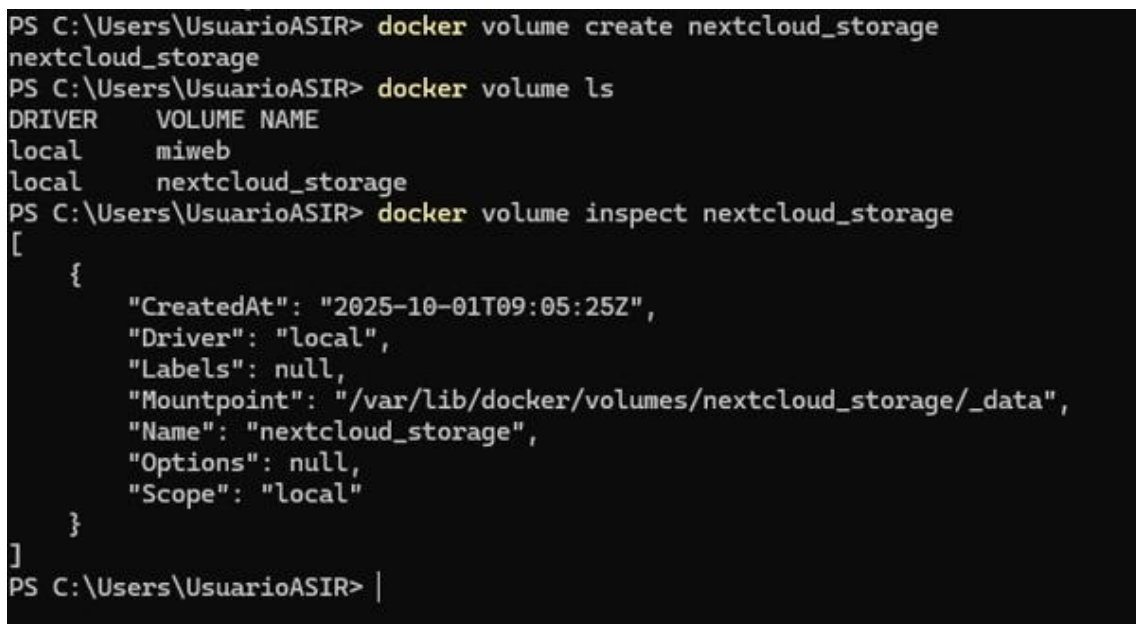
```
Windows PowerShell - power: X Windows PowerShell
PS C:\Users\UsuarioASIR> docker pull nextcloud
Using default tag: latest
```

Antes de generar el contenedor, debemos crear el volumen que se le añade durante su creación.

docker volumen create nextcloud_storage

Comprobamos su creación: **docker volumen ls**

Podemos inspeccionarlo: **docker volumen inspect nextcloud_storage**

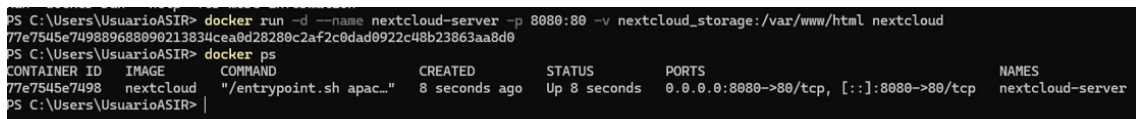


```
PS C:\Users\UsuarioASIR> docker volume create nextcloud_storage
nextcloud_storage
PS C:\Users\UsuarioASIR> docker volume ls
DRIVER      VOLUME NAME
local       miweb
local       nextcloud_storage
PS C:\Users\UsuarioASIR> docker volume inspect nextcloud_storage
[
  {
    "CreatedAt": "2025-10-01T09:05:25Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/nextcloud_storage/_data",
    "Name": "nextcloud_storage",
    "Options": null,
    "Scope": "local"
  }
]
PS C:\Users\UsuarioASIR> |
```

Ahora procedemos a crear el contenedor.

docker run -d --name nextcloud-serer -p 8080:80 -v nextcloud_storage:/var/www/html nextcloud

Y usamos **docker ps** para comprobar que está activo.



```
PS C:\Users\UsuarioASIR> docker run -d --name nextcloud-server -p 8080:80 -v nextcloud_storage:/var/www/html nextcloud
77e7545e749889688090213834cea0d28280c2af2c0dad0922c48b23863aa8d0
PS C:\Users\UsuarioASIR> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
77e7545e7498   nextcloud   "/entrypoint.sh apac..." 8 seconds ago  Up 8 seconds  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp  nextcloud-server
PS C:\Users\UsuarioASIR> |
```

“-d” es para que arranque en segundo plano y no nos llene la terminal con todos los datos de inicio.

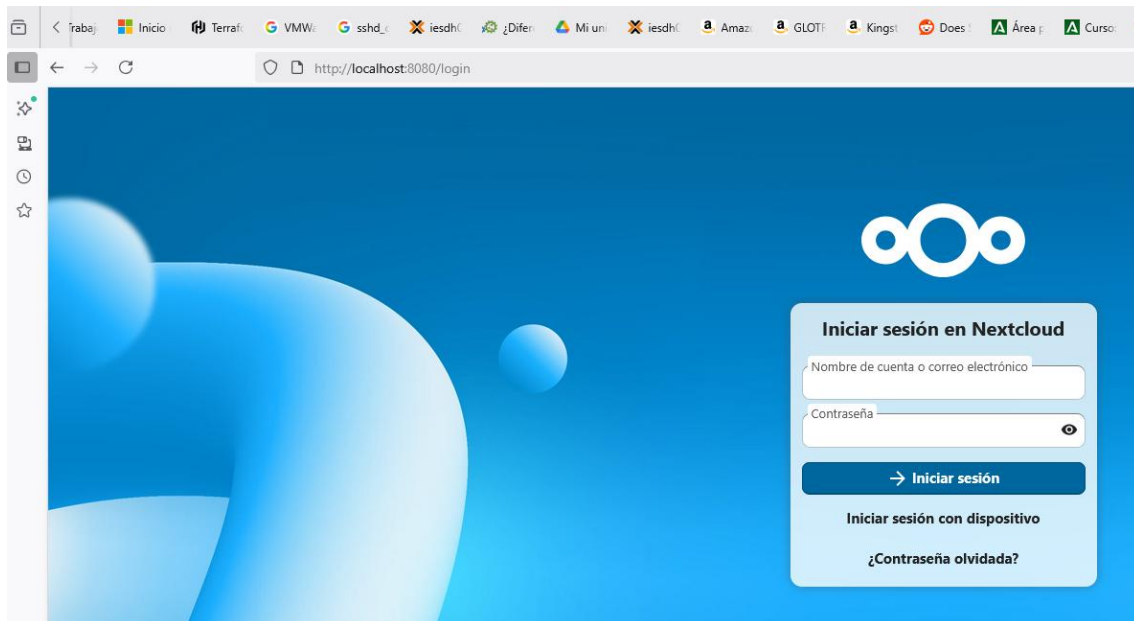
“--name”: establece el nombre del contenedor.

“-p”: los puertos de acceso. El primero es del equipo anfitrión y el segundo del contenedor.

“-v”: el del volumen. Primero el nombre del volumen y luego la ruta dentro del contenedor. Usamos esa ruta dentro del contenedor, porque es la que se indica en los foros¹.

Y por último la imagen que se usa para la creación del contenedor.

Para comprobar que funciona, abrimos el navegador web en el equipo anfitrión y usamos “localhost:8080” (o la IP_del_anfitrión:8080).



Ahora vamos a crear un documento para comprobar que se guarda en el volumen. En este caso sirve uno de texto.



Podemos verlo también desde dentro del contenedor.

```
root@77e7545e7498:/var/www/html/data/cristobal_suarez/files# cat Documents/Probando\ volúmenes\ en\ Docker.md
Soy Cristóbal Suárez Abad, de 2º ASIR y estoy en Optativa. Probando volúmenes en Docker.root@77e7545e7
```

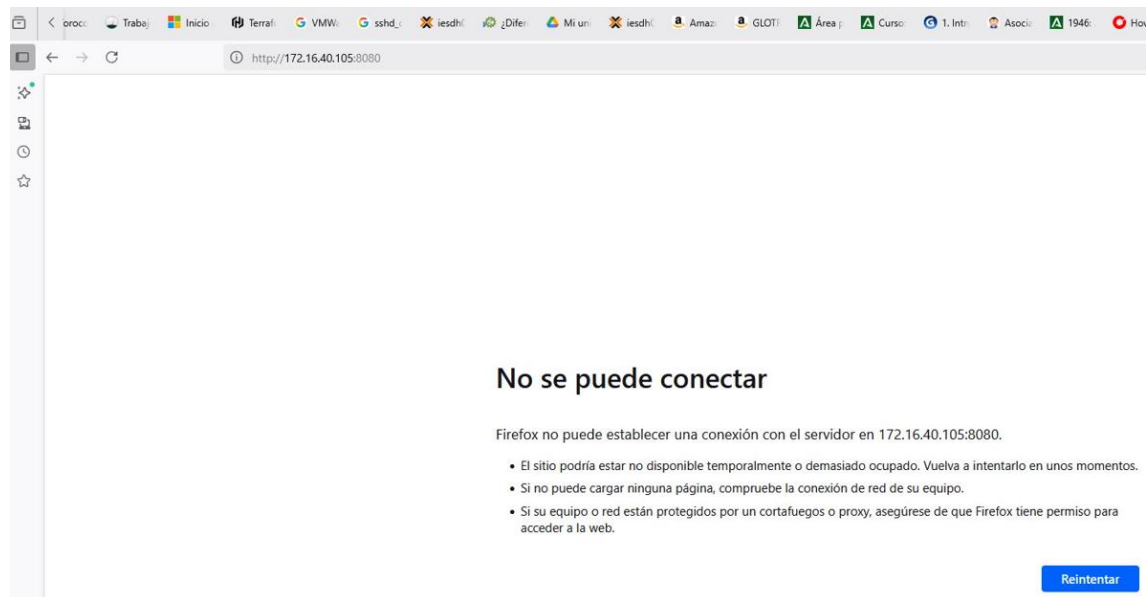
¹ <https://help.nextcloud.com/t/volume-management-in-docker-still-no-working-nextcloud/104585/3>

Procedemos a borrar el contenedor y a comprobar que ya no existe. Usamos:

docker rm -f nextcloud-server

```
PS C:\Users\UsuarioASIR> docker rm -f nextcloud-server
nextcloud-server
PS C:\Users\UsuarioASIR> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
PS C:\Users\UsuarioASIR> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
80f654bd7dd7   mediawiki:1.39.6   "docker-php-entrypoi..."   5 days ago    Exited (0) 5 days ago           mediawiki3
8c1d00fe6f09   mediawiki:1.40.2   "docker-php-entrypoi..."   5 days ago    Exited (0) 5 days ago           mediawiki2
a9a4c52cfe73   mediawiki          "docker-php-entrypoi..."   5 days ago    Exited (0) 5 days ago           mediawiki1
PS C:\Users\UsuarioASIR>
```

Como se puede ver, ya no se tiene acceso, porque no existe.



Volvemos a crear el contenedor.

```
Windows PowerShell - power... x Windows PowerShell x + v
PS C:\Users\UsuarioASIR> docker run -d --name nextcloud-server -p 8080:80 -v nextcloud_storage:/var/www/html nextcloud
f93f8395b1d20cb79d414820f15653dcce826890bf621989be3e72e55931acf4
PS C:\Users\UsuarioASIR>
```

Volvemos a acceder a NextCloud desde el navegador web y comprobamos que el archivo se encuentra donde antes. Es más, los usuarios siguen funcionando y no hay que crear uno nuevo.

The screenshot shows the NextCloud web interface in a browser window. The address bar indicates the URL: `http://172.16.40.105:8080/apps/files/files/47dir=/Documents`. The interface is in Spanish and shows the 'Documents' folder. A sidebar on the left contains navigation options like 'Todos los archivos', 'Archivos personales', 'Reciente', 'Favoritos', 'Compartidos', 'Etiquetas', and 'Árbol de carpetas'. The main content area displays a list of files in the 'Documents' folder. The file 'Probando volúmenes en Docker.md' is highlighted in yellow. Below the file list, it shows '5 archivos' and a total size of '1,1 MB'. The bottom of the screen shows the Windows taskbar with the search bar and system tray.

Tipo	Modificado	Personas	Nombre	Tam...	Modific...
			Example.md	1 KB	hace 10 minutos
			Nextcloud flyer.pdf	1 MB	hace 10 minutos
			Probando volúmenes en Docker.md	< 1 KB	hace 4 minutos
			Readme.md	< 1 KB	hace 10 minutos
			Welcome to Nextcloud Hub.docx	23 KB	hace 10 minutos
5 archivos				1,1 MB	

MariaDB

Un cliente necesita un sistema de gestión de bases de datos para una aplicación web. Se pide desplegar **MariaDB** en Docker con persistencia de datos.

1. Descarga la **imagen oficial de MariaDB** desde Docker Hub.
2. Crea un **contenedor** llamado mariadb-server a partir de esa imagen.
3. Configura un **volumen persistente** que guarde los datos de las bases de datos fuera del contenedor.
4. El contenedor debe iniciar con una **contraseña de root** definida mediante variables de entorno.
5. Expón el servicio en el **puerto 3306** de la máquina anfitriona.
6. Comprueba la conexión desde el cliente mysql (usa dbeaver). Realiza algún cambio
7. Elimina el contenedor y vuélvelo a crear para comprobar que el archivo sigue existiendo en la nube.

Lo primero que hacemos es descargar la imagen:

docker pull mariadb:latest

```
Windows PowerShell - power: X Windows PowerShell X + v
PS C:\Users\UsuarioASIR> docker pull mariadb:latest
latest: Pulling from library/mariadb
953cdd413371: Pull complete
52364fc3563c: Pull complete
5a1051e81802: Pull complete
eeb61de4991e: Pull complete
3bf2e947a240: Pull complete
32cbc8a9ec6b: Pull complete
0c69925a3f99: Pull complete
650ac881f9b7: Pull complete
Digest: sha256:8a061ef9813cf960f94a262930a32b190c3fbe5c8d3ab58456ef1df4b90fd5dc
Status: Downloaded newer image for mariadb:latest
docker.io/library/mariadb:latest
PS C:\Users\UsuarioASIR> |
```

Luego, creamos el volumen que vamos a usar cuando creemos el contenedor:

docker volume create mariadb_storage

```
PS C:\Users\UsuarioASIR> docker volume create mariadb_storage
mariadb_storage
PS C:\Users\UsuarioASIR> docker volume l
docker: unknown command: docker volume l

Usage:  docker volume COMMAND

Run 'docker volume --help' for more information
PS C:\Users\UsuarioASIR> docker volume ls
DRIVER      VOLUME NAME
local       mariadb_storage
local       miweb
local       nextcloud_storage
PS C:\Users\UsuarioASIR> docker volume inspect mariadb_storage
[
  {
    "CreatedAt": "2025-10-01T10:06:30Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/mariadb_storage/_data",
    "Name": "mariadb_storage",
    "Options": null,
    "Scope": "local"
  }
]
PS C:\Users\UsuarioASIR> |
```


Ahora vamos a crear el contenedor:

docker run -d --name mariadb-server -e MYSQL_ROOT_PASSWORD=12345 -v mariadb_storage:/var/lib/mysql -p 3306:3306 mariadb:latest

```
PS C:\Users\UsuarioASIR> docker run -d --name mariadb-server -e MYSQL_ROOT_PASSWORD=12345 -v mariadb_storage:/var/lib/mysql -p 3306:3306 mariadb:latest
a4f05d3e1c2c24687162e79a7938ce5e90be4bdf4c16536940b6320176e99a0f
PS C:\Users\UsuarioASIR> |
```

En este caso se incluye una variable para indicar la contraseña del ROOT de MySQL. En la documentación de MariaDB se indica que se pueden usar tres posibles variantes²:

- **MARIADB_ROOT_PASSWORD_HASH**
- **MARIADB_ROOT_PASSWORD**
- **MYSQL_ROOT_PASSWORD**

En ningún momento se indica que haya alguna diferencia entre una y otra.

La ruta que se indica dentro del contenedor se ha conseguido de docker³.

² <https://mariadb.com/docs/server/server-management/install-and-upgrade-mariadb/installing-mariadb/binary-packages/automated-mariadb-deployment-and-administration/docker-and-mariadb/mariadb-server-docker-official-image-environment-variables>

³ <https://forums.docker.com/t/persistent-mariadb-storage/100946>

Ahora vamos a crear una instancia en MariaDB. Para ello entramos con el “root” (aún no hay ningún otro usuario). Creamos una base de datos, un usuario y le damos permisos a ese usuario sobre la base de datos.

```
MariaDB [(none)]> create database devops01;  
Query OK, 1 row affected (0.001 sec)
```

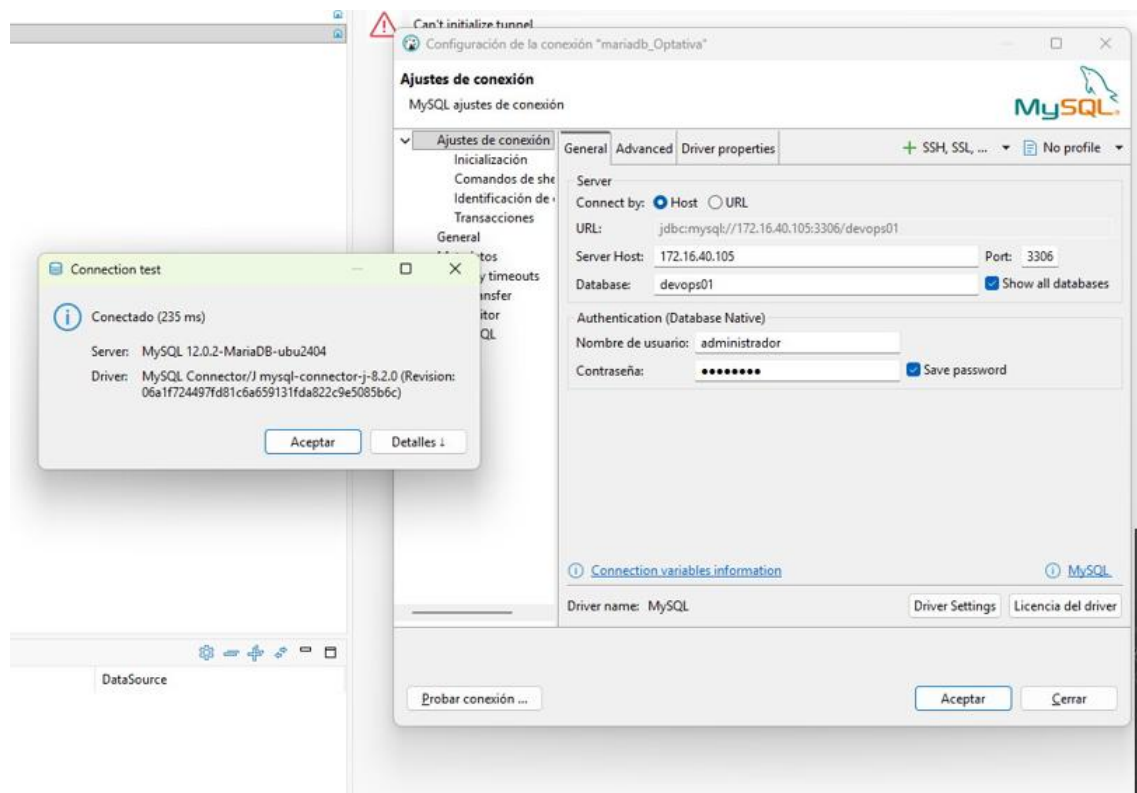
```
MariaDB [(none)]> CREATE USER 'administrador'@'%' IDENTIFIED BY 'pcjabela';  
Query OK, 0 rows affected (0.004 sec)
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON devops01.* TO 'administrador'@'%';  
Query OK, 0 rows affected (0.002 sec)
```

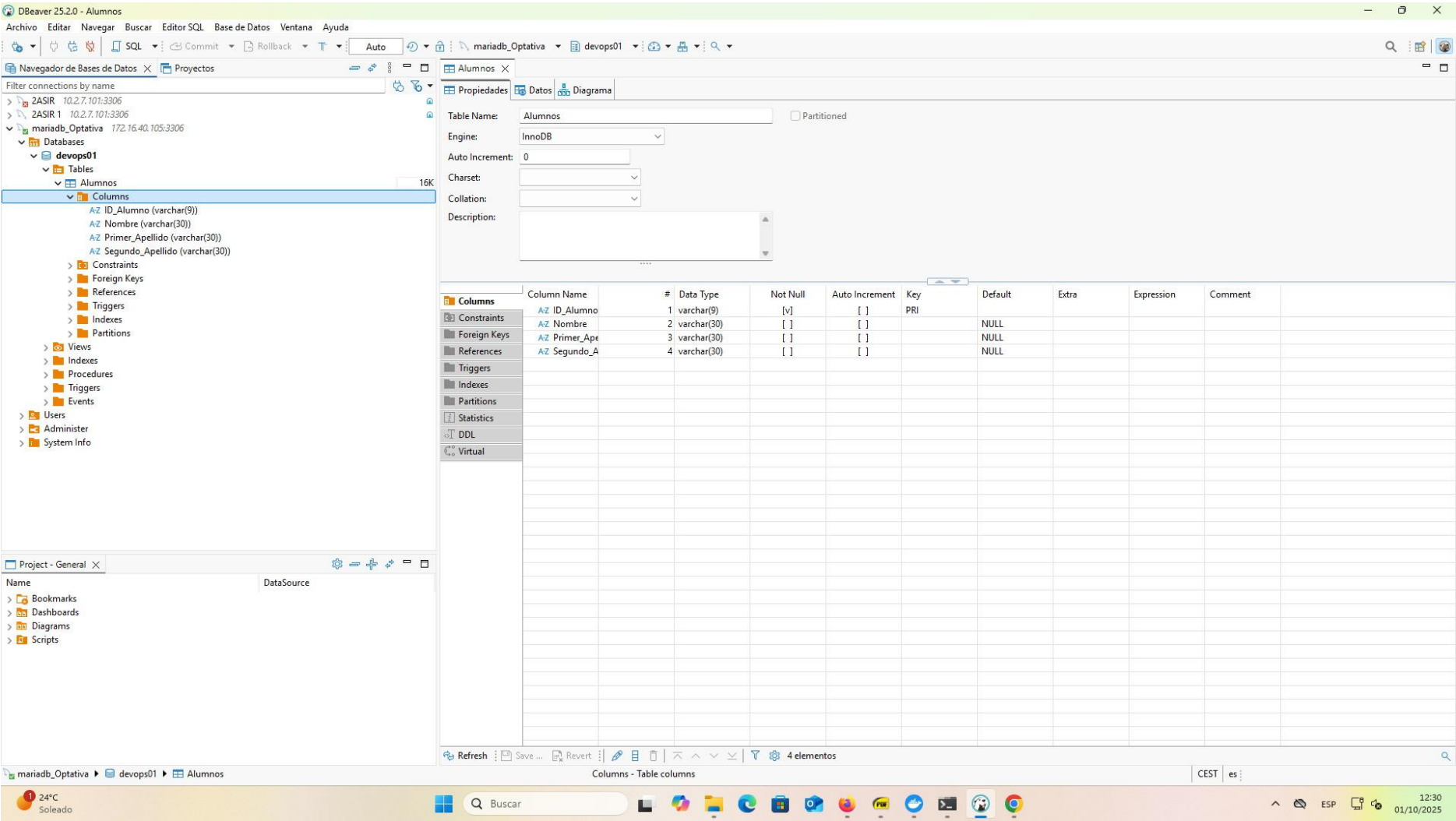
```
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.001 sec)
```

```
MariaDB [(none)]> |
```

Ahora procedemos a conectarnos desde un cliente con DBeaver: para ello usamos una configuración parecida a la de ejercicios anteriores en la asignatura ASGBD. Pero en esta ocasión no hemos tenido que configurar la pestaña de SSH.



Desde el usuario, procedemos a hacer ciertas modificaciones: en la base de datos hemos creado una tabla y en ella varias columnas.



Ahora borramos el contenedor.

docker rm -f mariadb-server

```
PS C:\Users\UsuarioASIR> docker rm -f mariadb-server
mariadb-server
PS C:\Users\UsuarioASIR> |
```

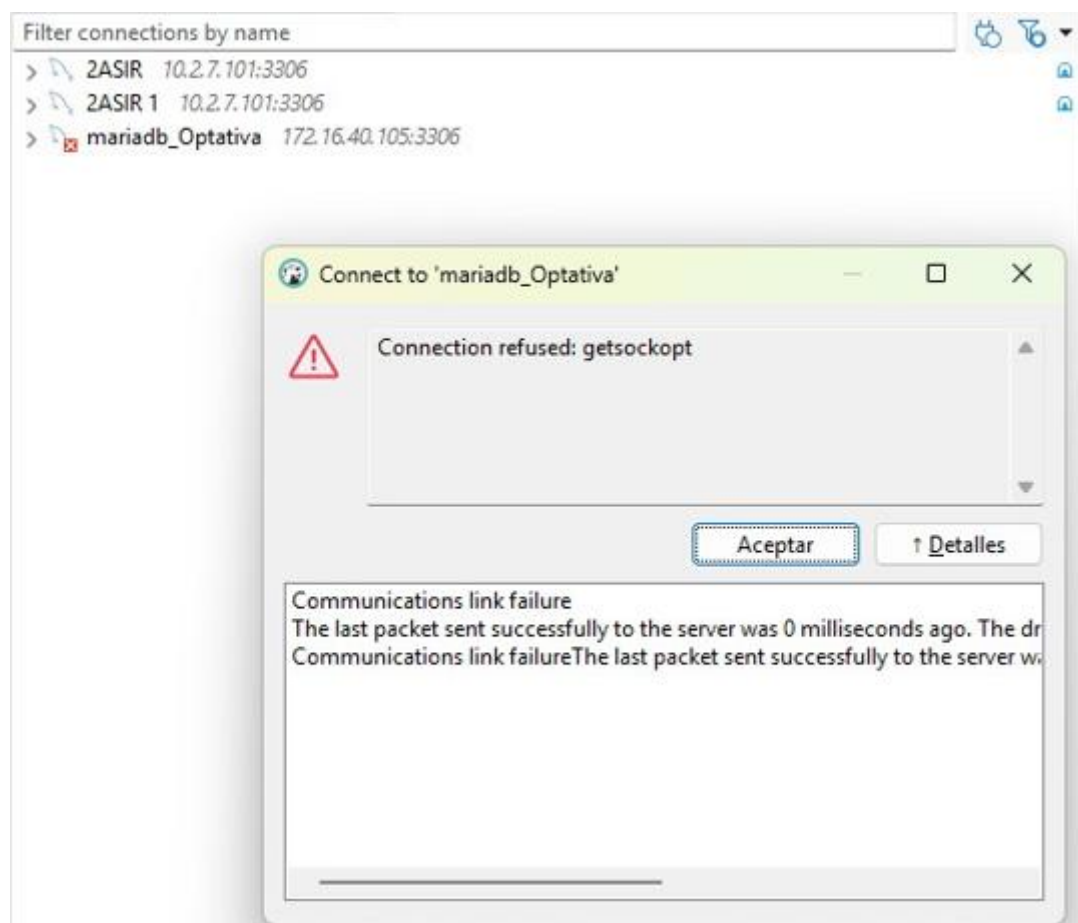
Para comprobar que efectivamente está borrada, usamos:

docker ps: muestra cuales están activas.

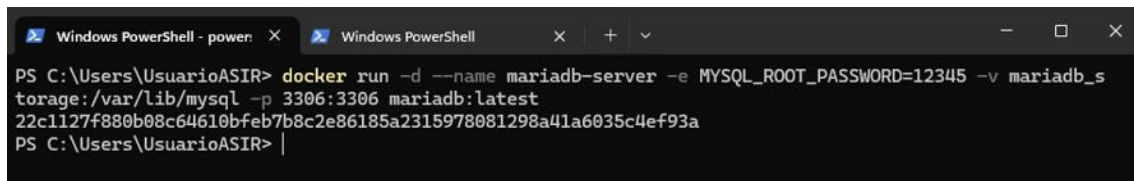
docker ps -a: muestra todas las que tengamos.

```
PS C:\Users\UsuarioASIR> docker rm -f mariadb-server
mariadb-server
PS C:\Users\UsuarioASIR> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
f93f8395b1d2   nextcloud "/entrypoint.sh apac..." About an hour ago Up About an hour 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp nextcloud-server
PS C:\Users\UsuarioASIR> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
f93f8395b1d2   nextcloud "/entrypoint.sh apac..." About an hour ago Up About an hour 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp nextcloud-server
80f654bd7dd7   mediawiki:1.39.6 "docker-php-entrypoi..." 5 days ago    Exited (0) 5 days ago                                mediawiki3
8c1d09fe6f09   mediawiki:1.40.2 "docker-php-entrypoi..." 5 days ago    Exited (0) 5 days ago                                mediawiki2
a9a4c52cfe73   mediawiki "docker-php-entrypoi..." 5 days ago    Exited (0) 5 days ago                                mediawiki1
PS C:\Users\UsuarioASIR> |
```

Además, que el cliente no puede conectarse.



Procedemos a volver a crear el contenedor:



```
Windows PowerShell - power: X Windows PowerShell X + v
PS C:\Users\UsuarioASIR> docker run -d --name mariadb-server -e MYSQL_ROOT_PASSWORD=12345 -v mariadb_s
torage:/var/lib/mysql -p 3306:3306 mariadb:latest
22c1127f880b08c64610bfeb7b8c2e86185a2315978081298a41a6035c4ef93a
PS C:\Users\UsuarioASIR> |
```

Ahora, si volvemos a DBeaver, podremos conectarnos de nuevo y vemos que la configuración de conexión, la base de datos, la tabla y las columnas están tal y como las dejamos.

