

EJERCICIO 2 - SERVIDOR DE BASE DE DATOS

Realizado por Andrea Gómez Fueyo y Sandra Rujas

EJERCICIO 2 - SERVIDOR DE BASE DE DATOS

1. Descarga de imagen.
2. Despliegue de contenedor.
3. Acceso a la base de datos.
4. Borrado del contenedor.
5. Comprobación de la existencia de datos.
6. Creación de nuevo contenedor.
7. Comprobación de la existencia de la tabla en la base de datos.
8. Eliminación de imagen.
9. Eliminación de volumen, imagen y contenedor.

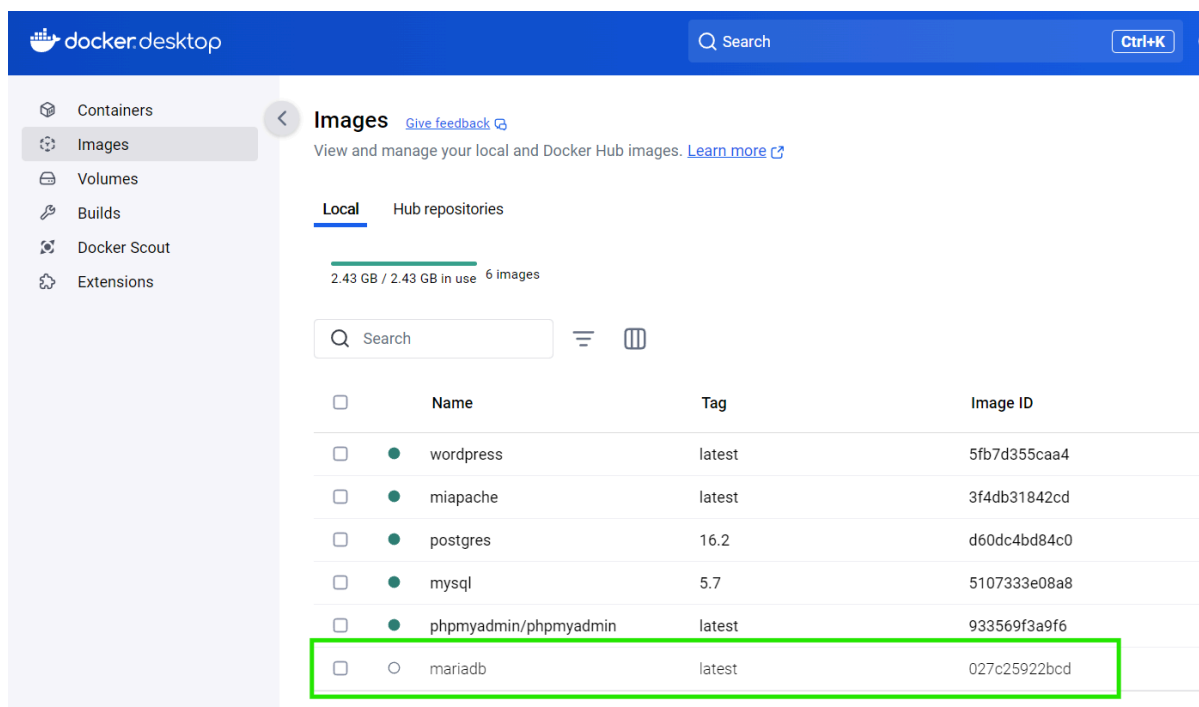
1. Descarga de imagen.

Abre Docker Desktop. Busca mariadb en la sección de imágenes. Selecciona la imagen oficial. Descárgala si no la tienes.

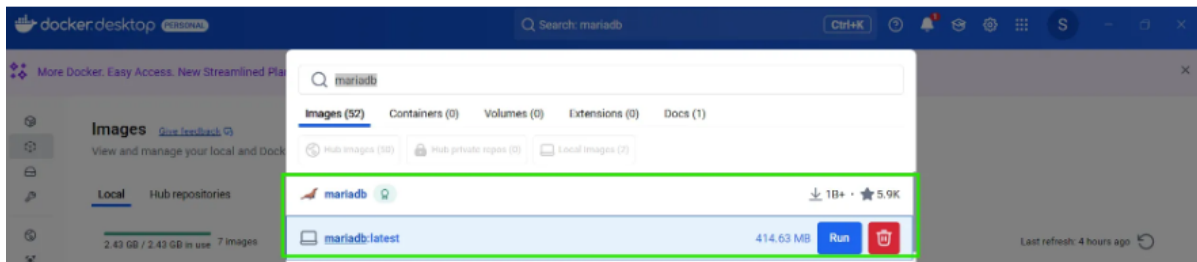
- Si no puedes iniciar sesión, la imagen puedes descargarla mediante comandos.

```
$docker pull mariadb:latest
```

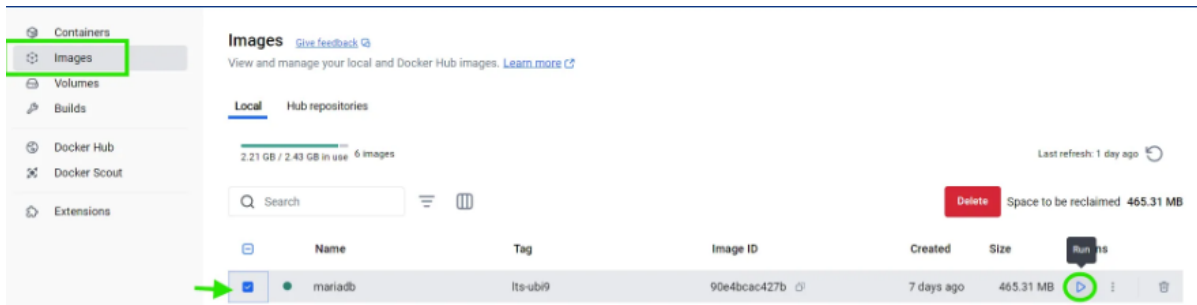
```
C:\Users\34615>docker pull mariadb:latest
latest: Pulling from library/mariadb
5a7813e071bf: Downloading [=====] 25.22MB/29.75MB
f67c6fbc0ef5: Download complete
1f731489858b: Download complete
760f6e3db6bf: Download complete
65dd09f27c61: Waiting
2cbd49ab14b1: Downloading [=====] 11.84MB/89.76MB
640331c2cc76: Waiting
edb426f4a1af: Waiting
```



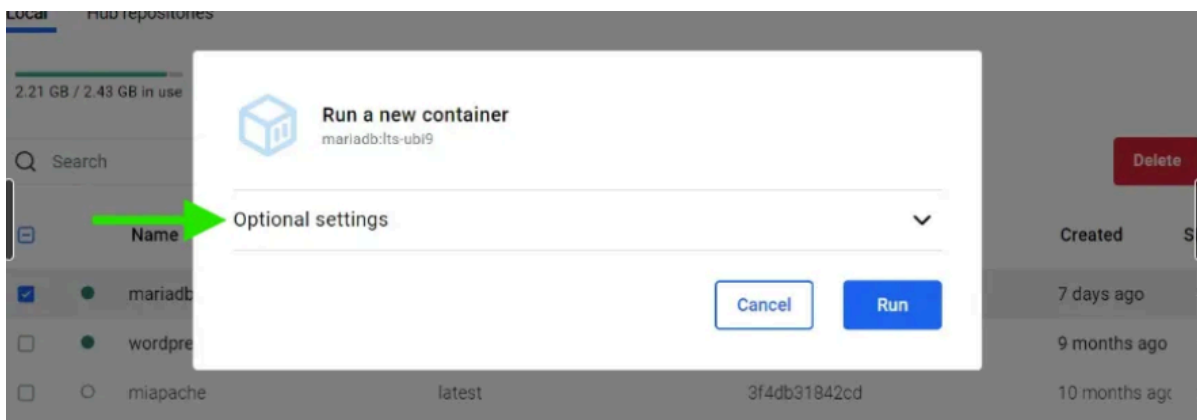
- Si puedes acceder a la búsqueda de imagen, entras en “*Images*”, buscas la última de mariadb y pulsas pull para descargarla.



- En la sección de imágenes, seleccionamos la que queremos utilizar, que en nuestro caso es la de mariadb, y pulsamos el botón de run:



- Al darle al run, nos permite crear un nuevo contenedor:



Run a new container
mariadb:latest-ubi9

Optional settings ^

Container name

A random name is generated if you do not provide one.

Ports
Enter "0" to assign randomly generated host ports.

Host port :3306/tcp

Volumes

Host path ... Container path +

Environment variables

Variable Value +

Cancel Run

2. Despliegue de contenedor.

Al cual le tenemos que poner las siguientes **características**:

- **Nombre del contenedor:** bbdd .
- **Puerto:** 3306 - debe poder conectarse externamente
- Utiliza un **volumen** llamado datos-mariadb .
- Usa las **variables de entorno** necesarias para que el usuario root tenga la password root , la base de datos por defecto sea base , y se cree un usuario daw , con password daw.

Run a new container

mariadb:latest-ubi9

Container name

bbdd

A random name is generated if you do not provide one.

Ports

Enter "0" to assign randomly generated host ports.

Host port

3306

:3306/tcp

Volumes

Host path

datos-mariadb

...

Container path

/var/lib/mysql

+

Environment variables

Variable

MARIADB_ROOT_PASSWORD

Value

root

—

Variable

MARIADB_DATABASE

Value

base

—

Variable

MARIADB_USER

Value

daw

—

Variable

MARIADB_PASSWORD

Value

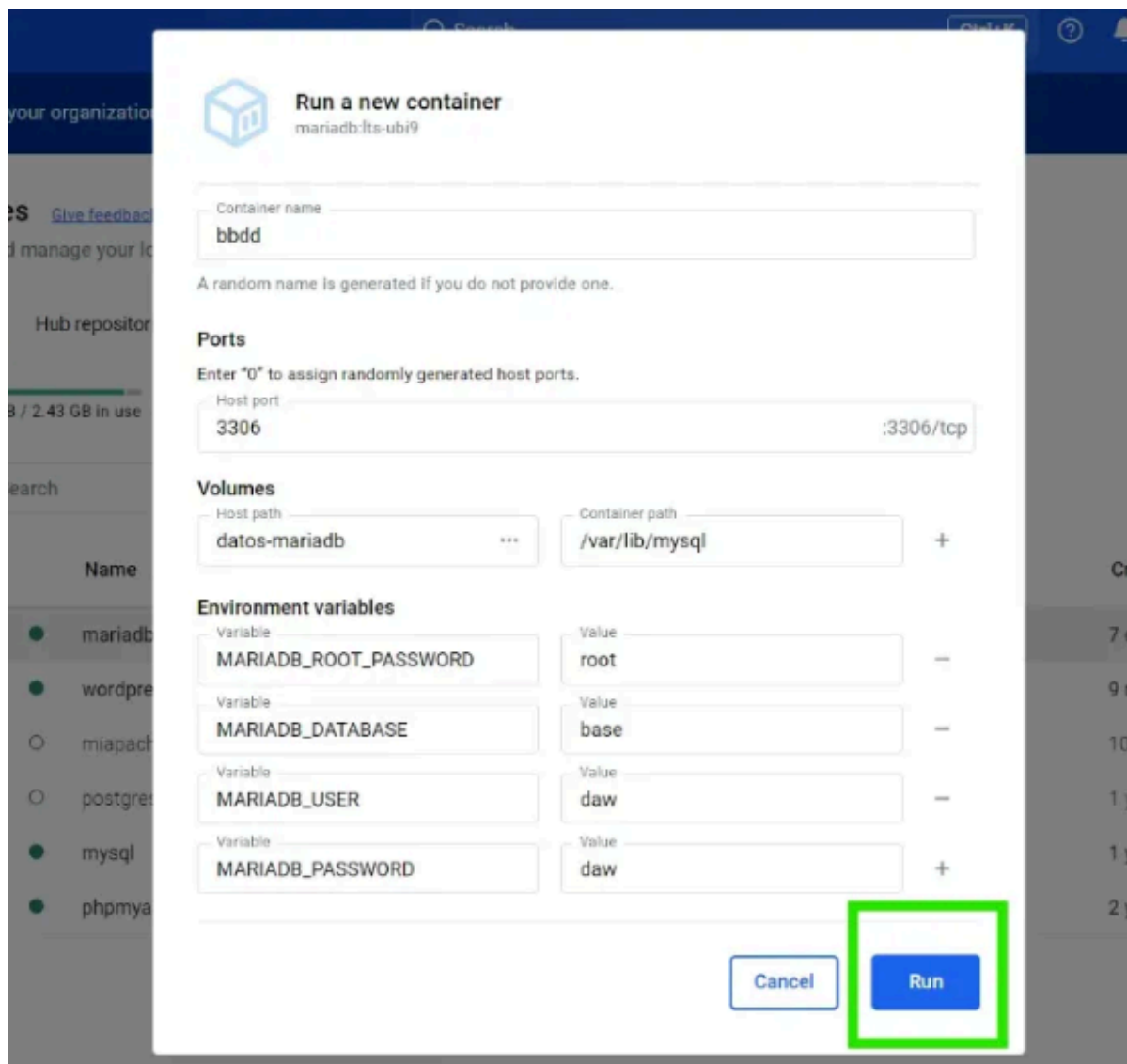
daw

+

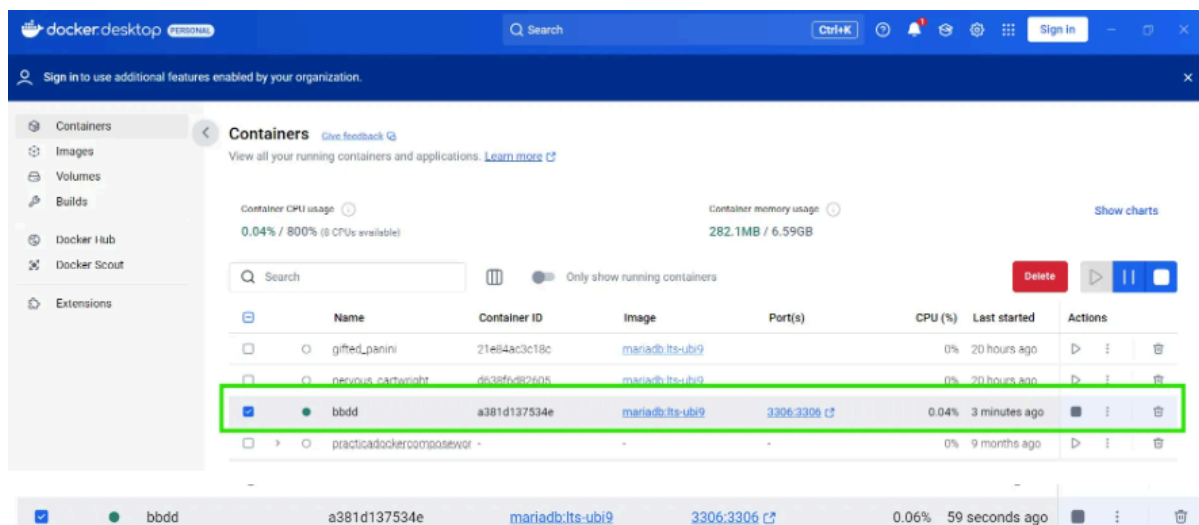
Cancel

Run

- Arrancamos el contenedor:



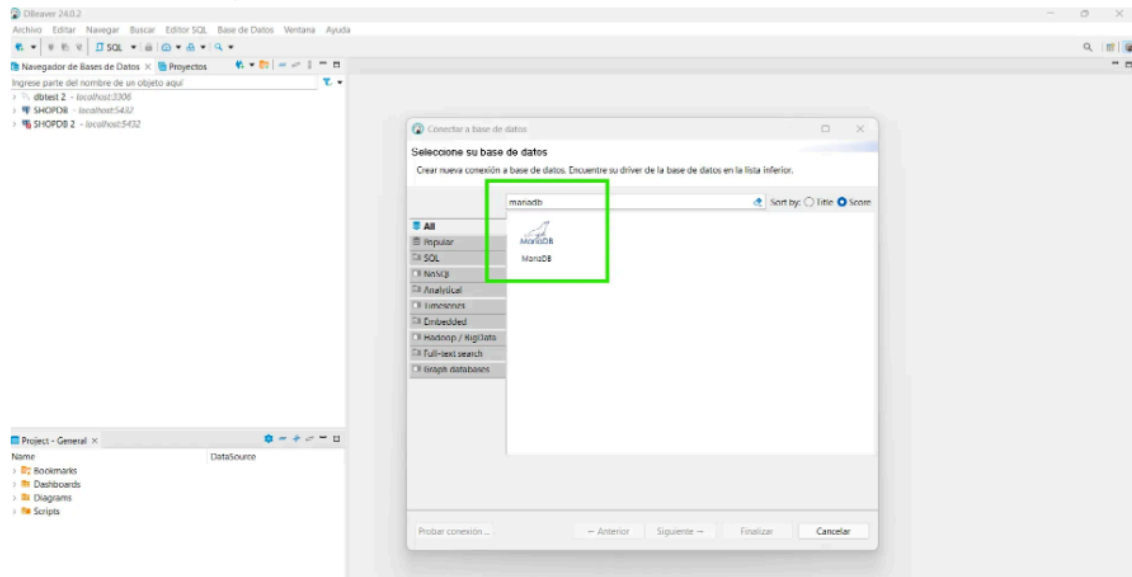
Una vez pulsado el RUN, nos vamos a la sección de contenedores para comprobar que finalmente se haya creado:



3. Acceso a la base de datos.

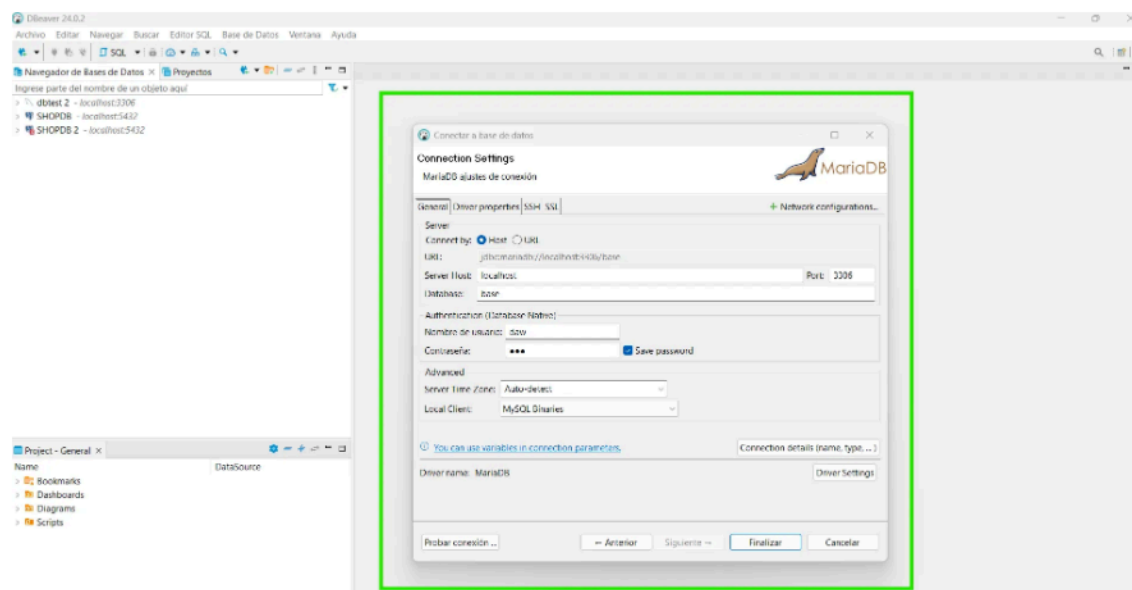
Accede a la base de datos usando una herramienta gráfica, como, por ejemplo dbeaver .
Conéctate con el usuario daw . Crea una base de datos y alguna tabla.

- Abrimos **DBeaver** y crear una nueva conexión a MariaDB:

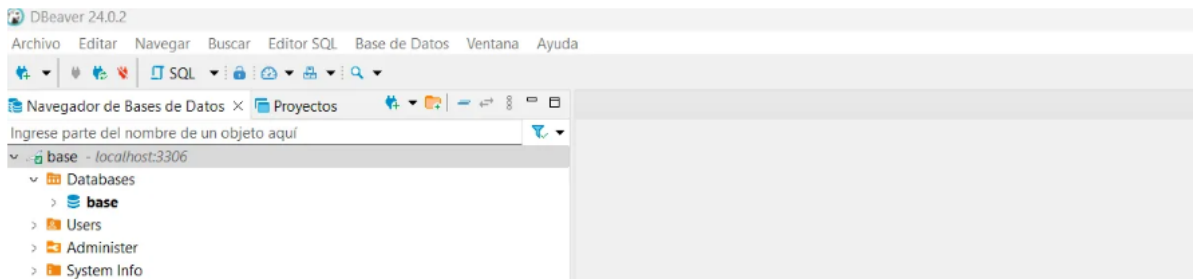


- Configuramos la conexión:

- **Host:** localhost
- **Puerto:** 3306
- **Usuario:** daw
- **Contraseña:** daw
- **Base de datos:** base



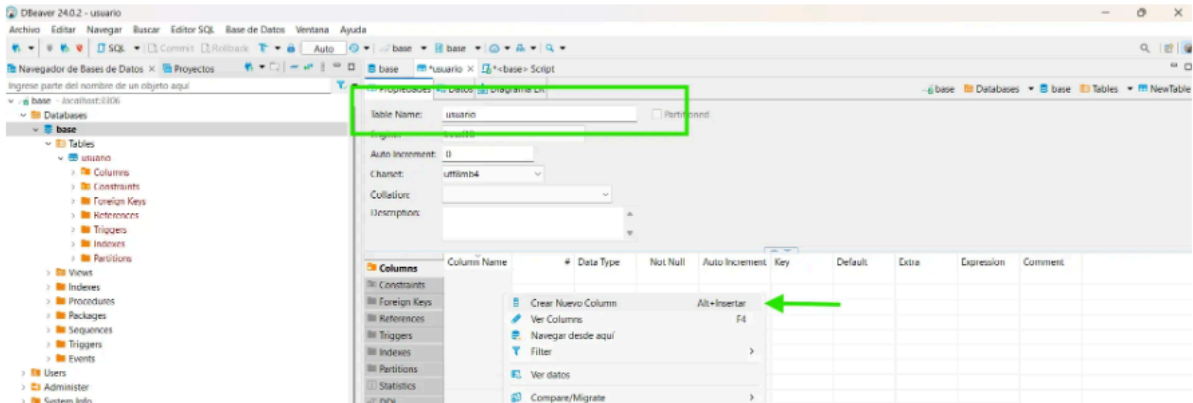
En la columna de la izquierda, podemos comprobar que se ha creado la conexión correctamente:



- Creamos una tabla:

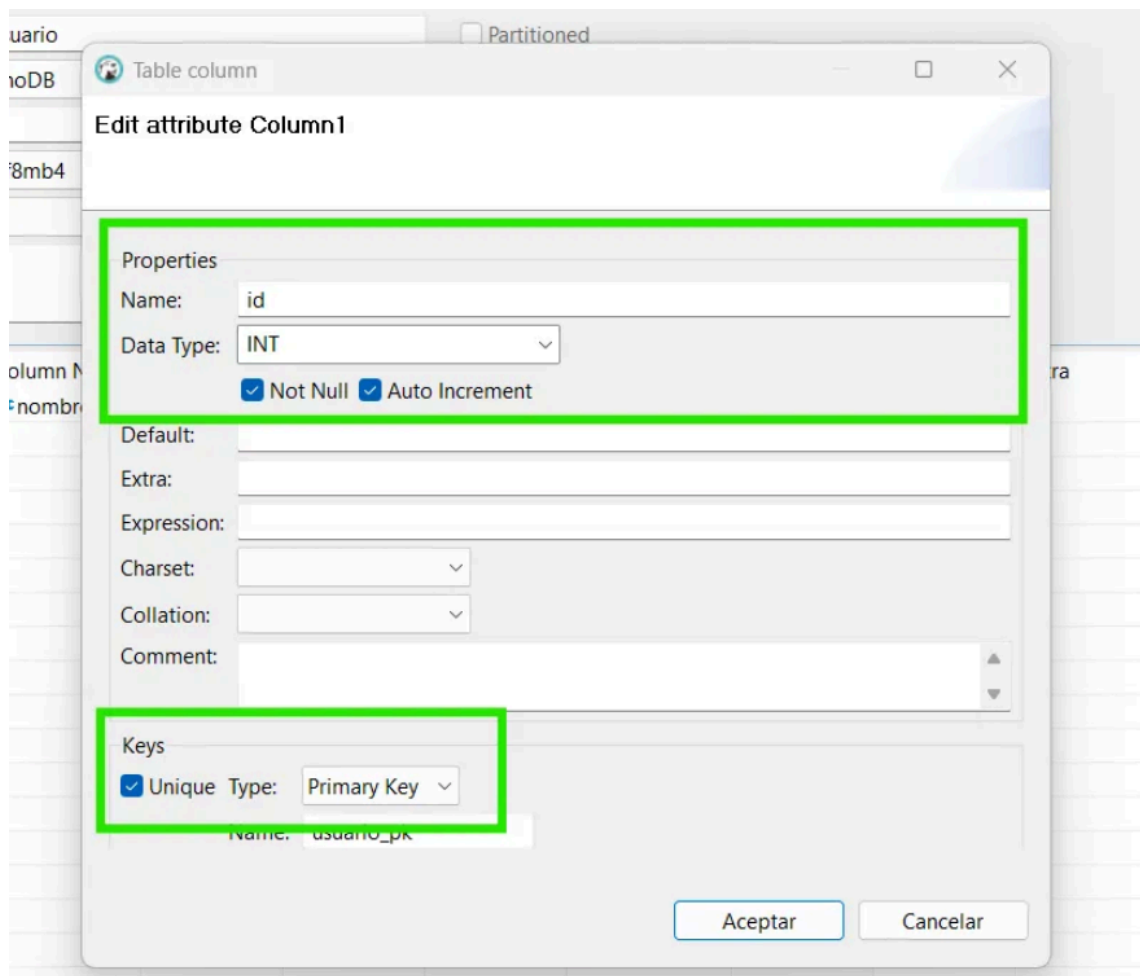
Hacemos click en el botón derecho y pulsamos en “Crear Nueva Columna” para agregar datos:

En *Table Name* introducimos el nombre de la tabla que queramos, en nuestro caso hemos elegido: “usuario”.

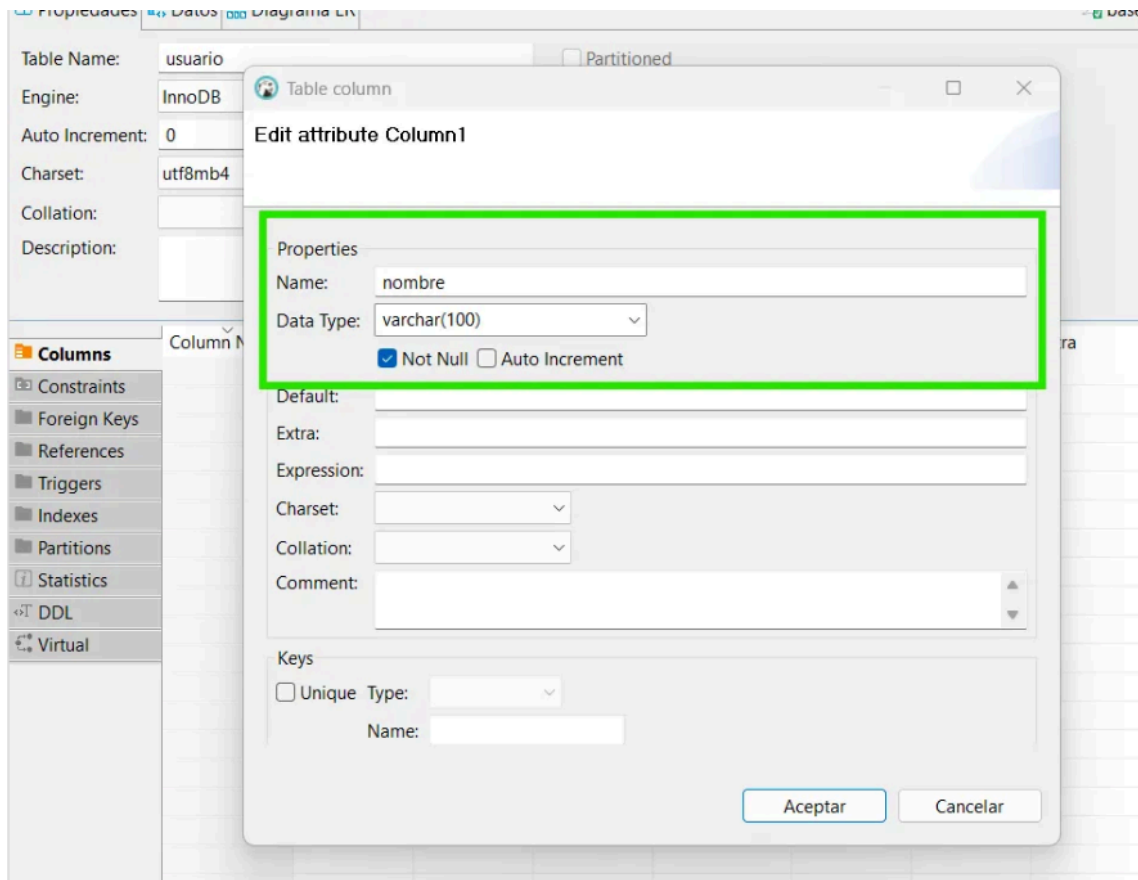


Después hemos creado las columnas para añadir los atributos a nuestra tabla:

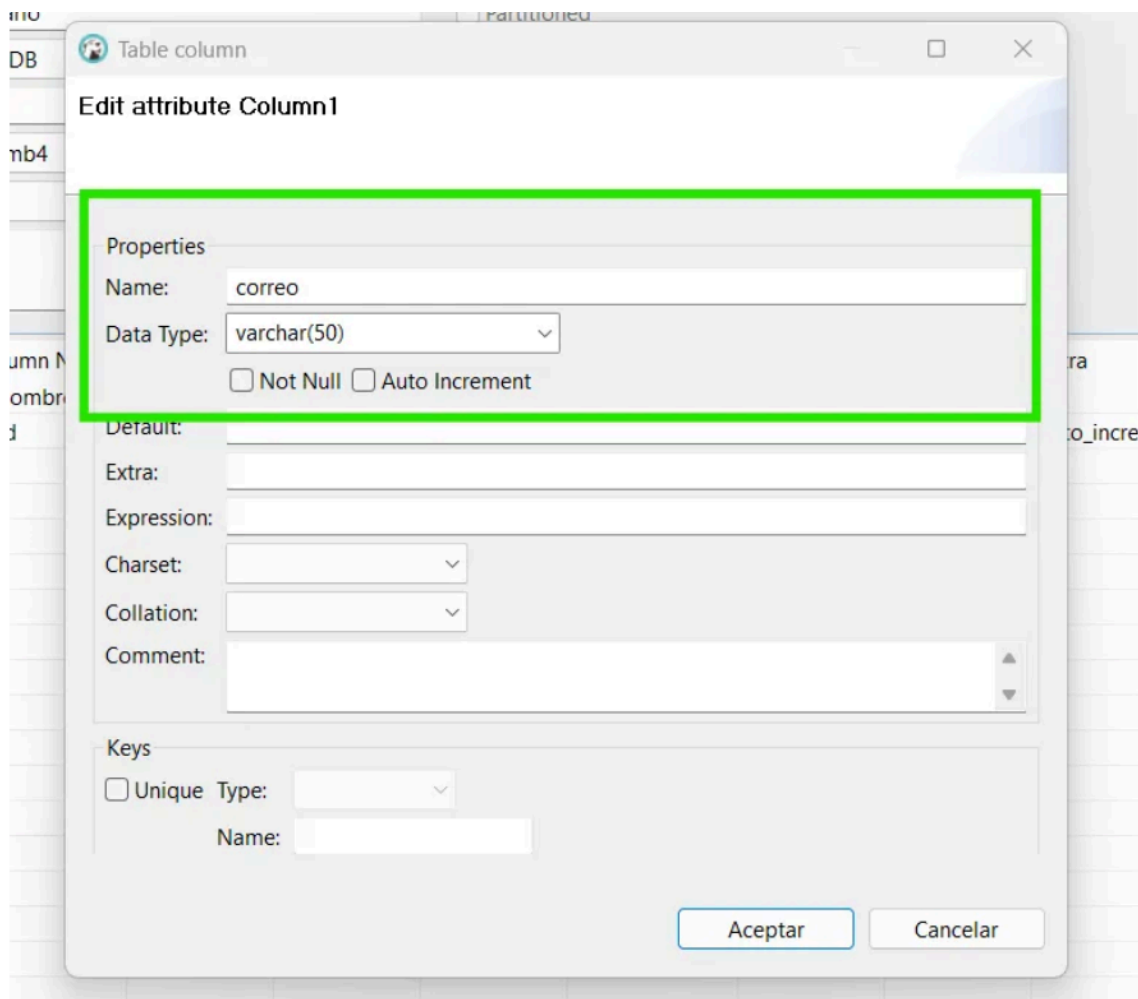
- Un **id**: que es un INT, AUTO_INCREMENT y PK.



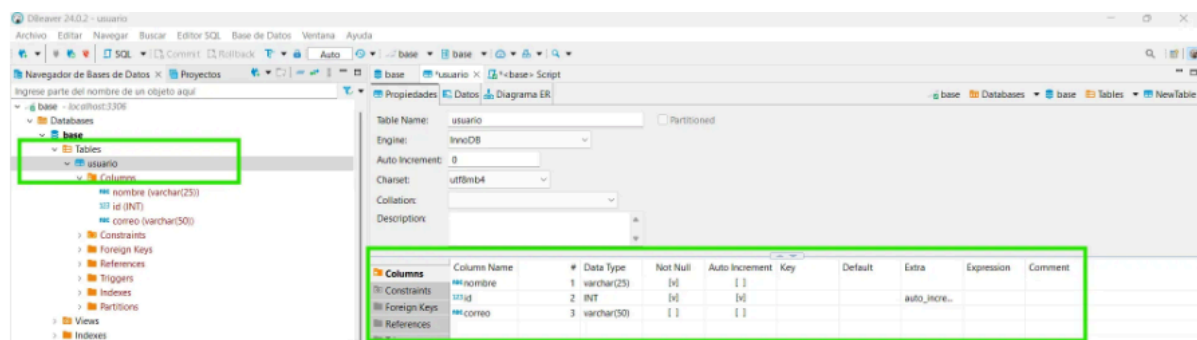
- Un **nombre**: que es un VARCHAR (25) y que no puede ser NULL.



- Un **correo**: que es u VARCHAR de 50.



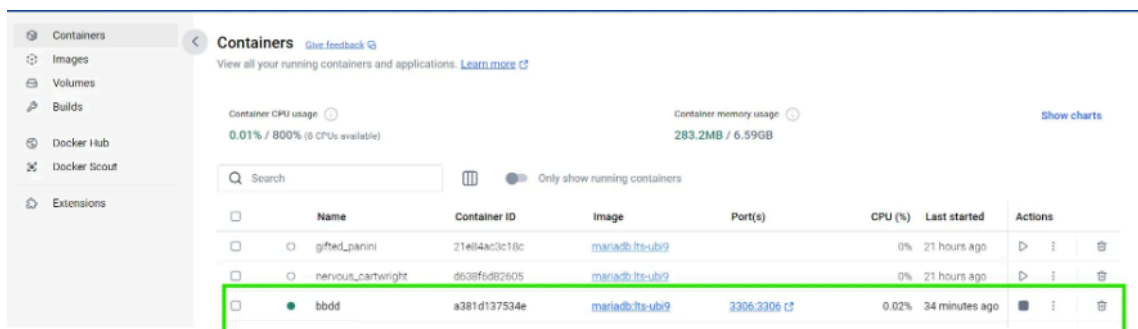
Comprobamos la creación de la tabla:



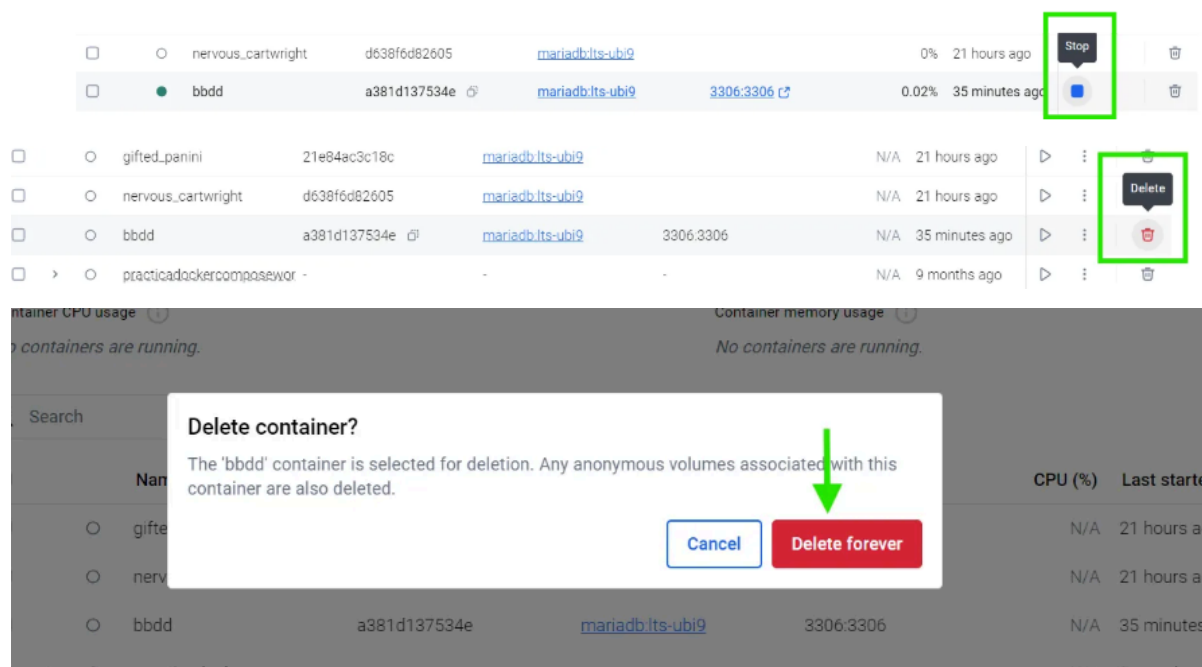
4. Borrado del contenedor.

Una vez creada la tabla, intentamos borrar el contenedor:

- Vamos a la pestaña "**Containers**" y buscamos el contenedor bbdd:



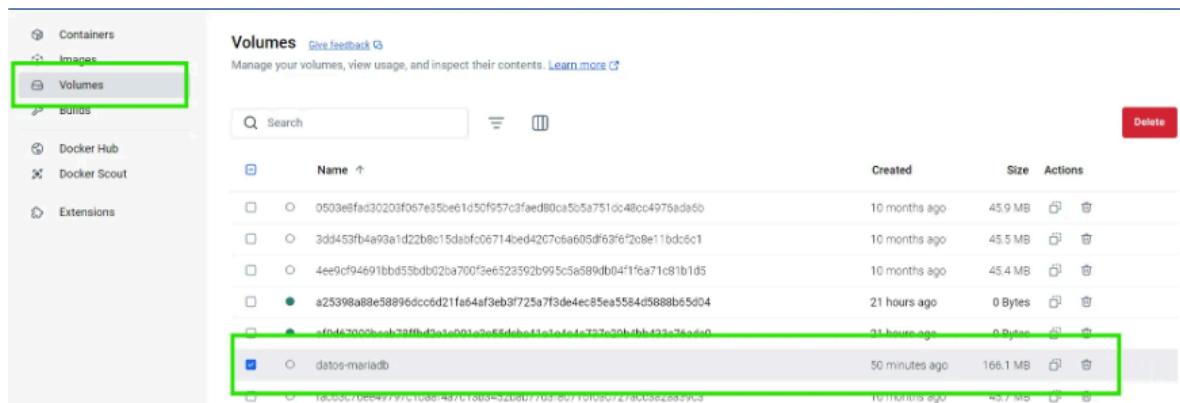
- Lo paramos y le damos a eliminar:



5. Comprobación de la existencia de datos.

Vemos como en Docker Desktop el volumen que contiene los datos no se ha borrado:

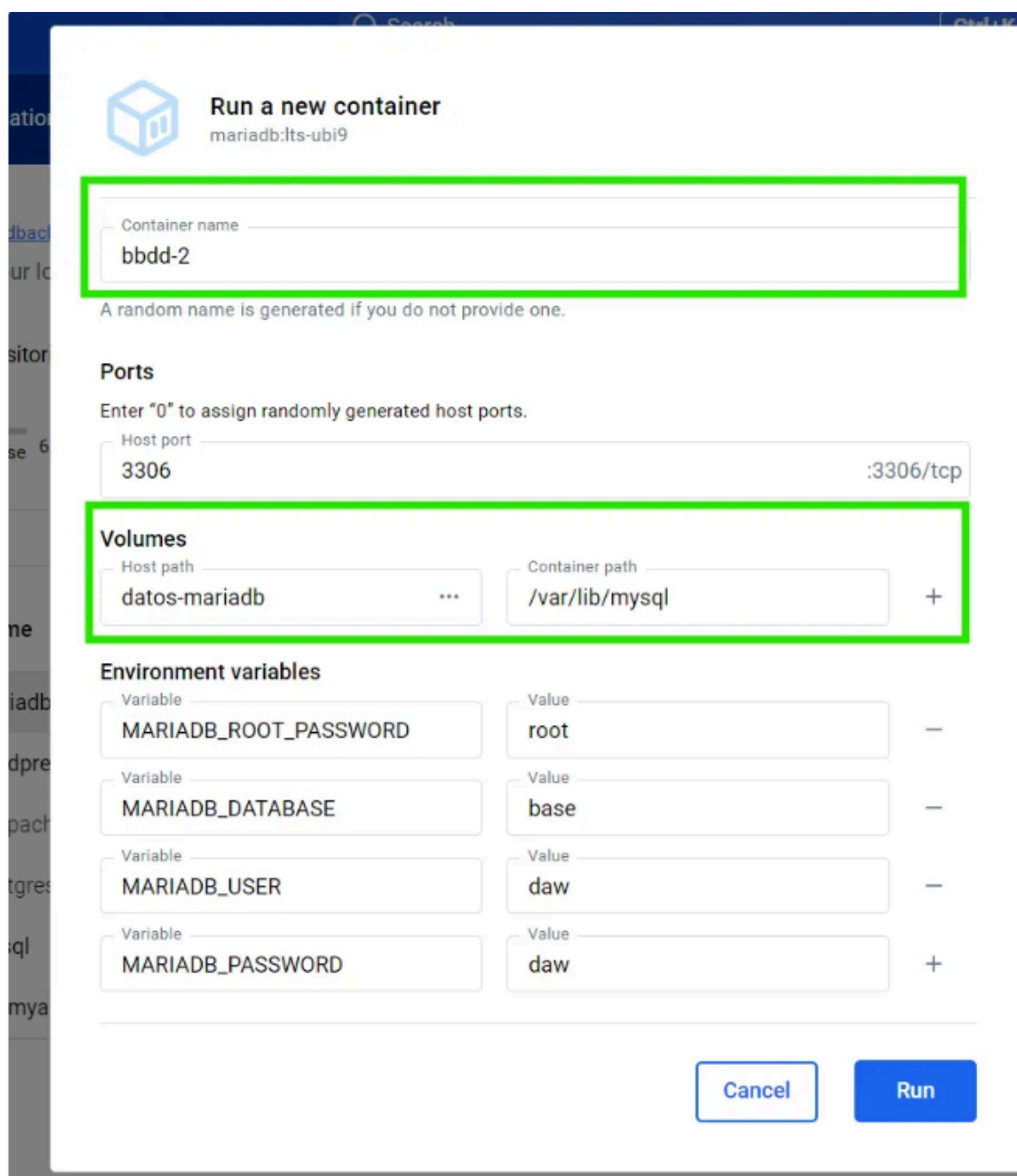
- Vamos a la pestaña "**Volumes**" en Docker Desktop. Buscamos datos-mariadb y verificar que sigue ahí:

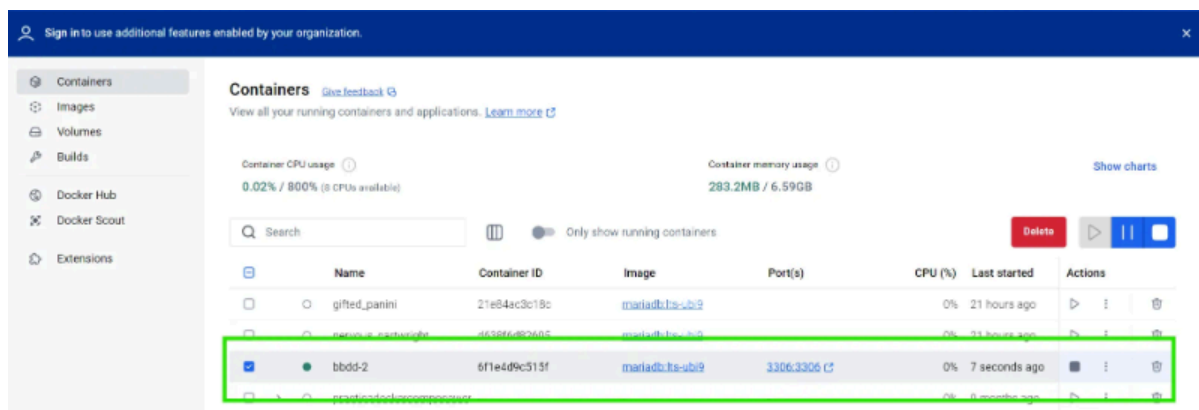


6. Creación de nuevo contenedor.

Creamos otro contenedor con un servidor de base de datos que use el mismo volumen:

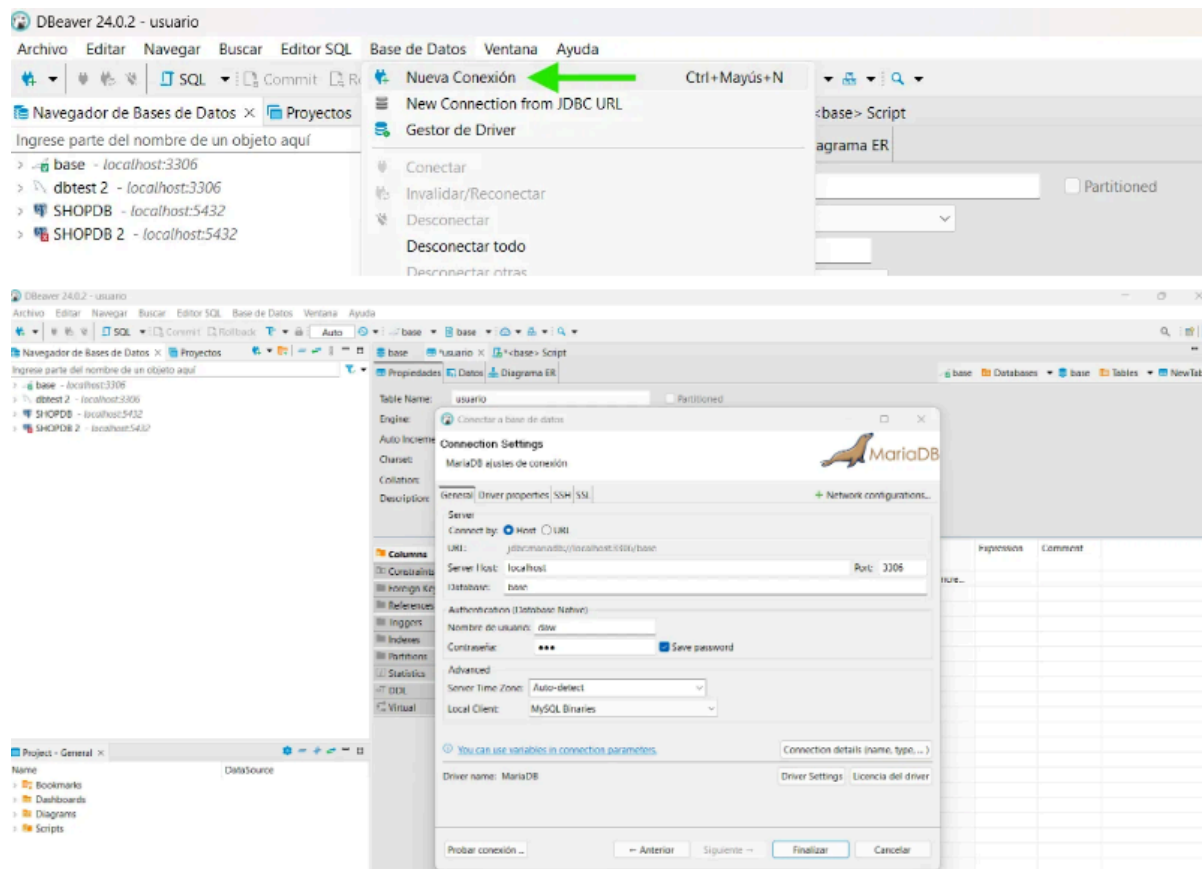
- Llamamos al contenedor **bbdd-2** y utilizamos el mismo volumen que anteriormente:



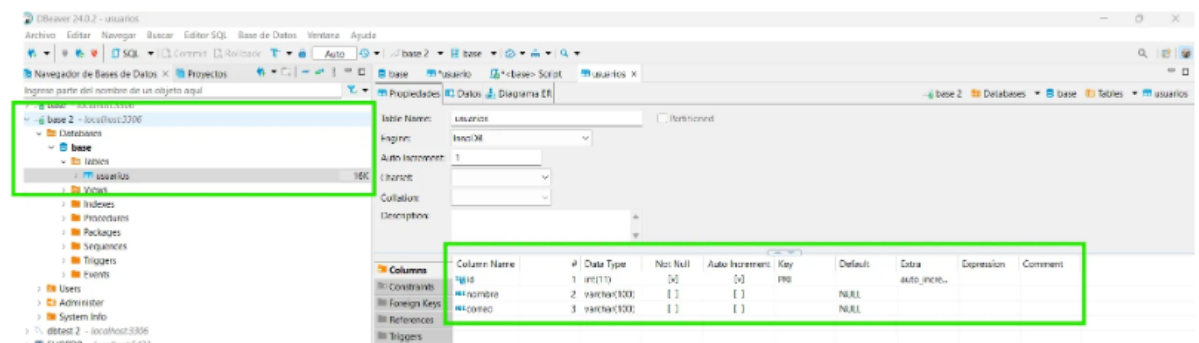


7. Comprobación de la existencia de la tabla en la base de datos.

Volvemos a **DBeaver** y creamos una nueva conexión a **bbdd-2**:

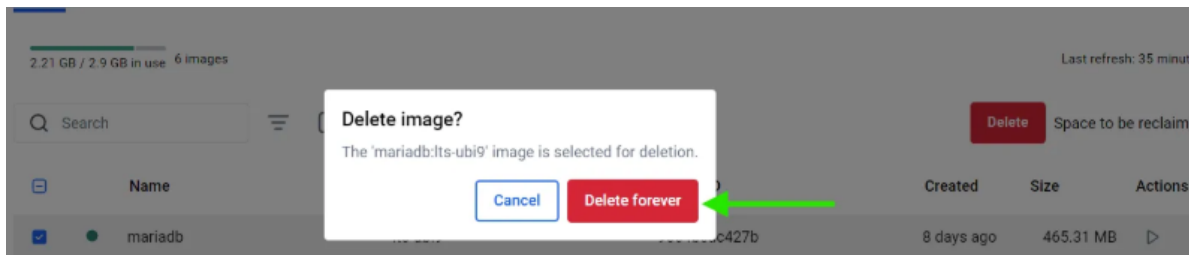
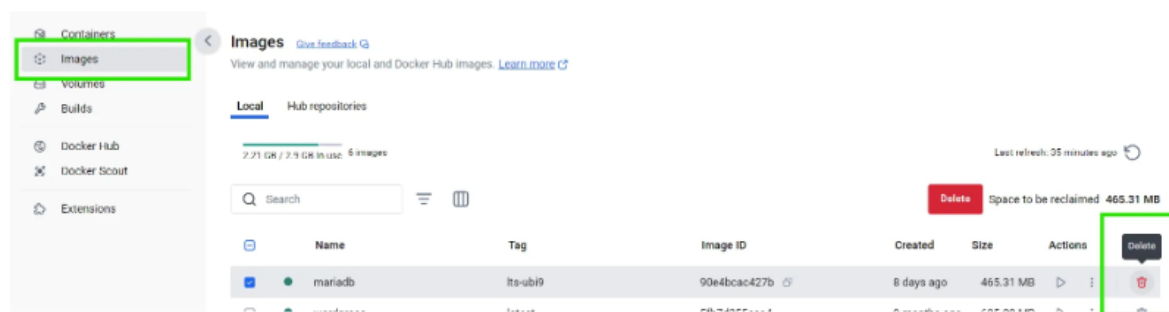


Tras darle al botón de 'Finalizar', en la columna de la izquierda, podemos comprobar, que efectivamente, se ha creado una nueva conexión, que mantiene los datos que hemos introducido en el ejercicio anterior, es decir, la tabla de **usuarios** con sus correspondientes **columnas**:



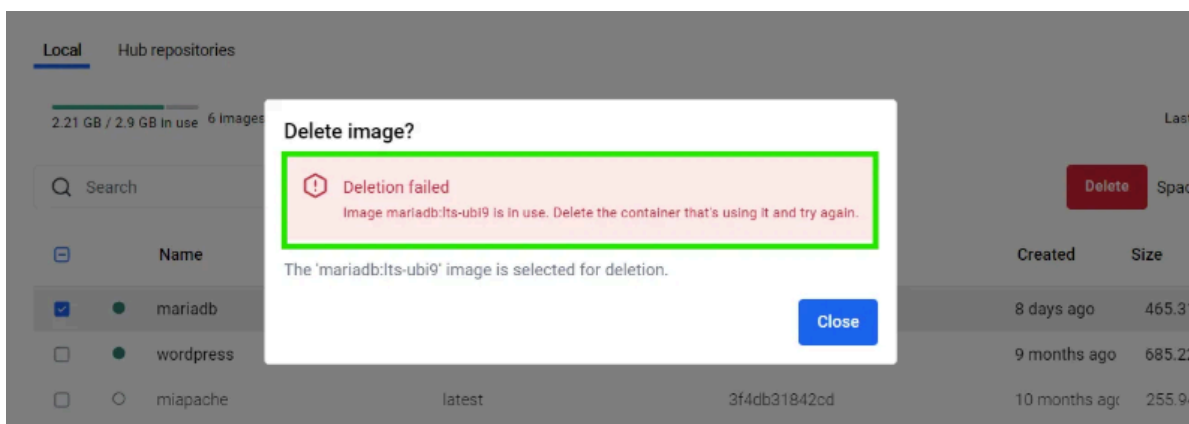
8. Eliminación de imagen.

Seguidamente, vamos a la pestaña de **"Images"** en Docker Desktop y buscamos mariadb e intentamos eliminarla:



¿Qué sucede?

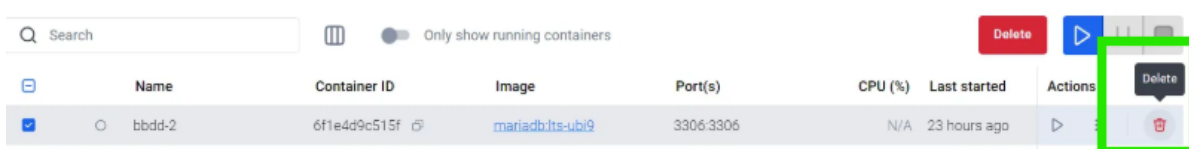
En este caso podemos comprobar que no es posible su eliminación porque hay un contenedor en ejecución.



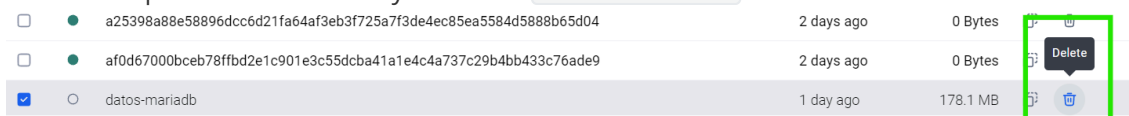
9. Eliminación de volumen, imagen y contenedor.

Borramos todo, volumen, imagen y contenedor. Para ello deberemos seguir el siguiente orden, ya que si no, no te dejaría eliminar la imagen de mariadb, puesto que te saldría que está en uso.

- Eliminamos el contenedor `bbdd-2` en **Docker Desktop**:



- Vamos a la pestaña **"Volumes"** y borramos `datos-mariadb`:



- Y por último vamos a **"Images"** y eliminamos `mariadb`:

The screenshot shows the Docker Desktop interface. At the top, a table lists images with columns: Name, Tag, Image ID, Created, Size, and Actions. The first row is for 'mariadb' with tag 'lts-ubi9' and ID '90e4bcac427b'. A red box highlights the 'Delete' button in the Actions column. Below the table, a 'Delete image?' dialog box is open, asking for confirmation to delete the 'mariadb:lts-ubi9' image. The dialog has 'Cancel' and 'Delete forever' buttons. A green arrow points to the 'Delete forever' button. The background shows the 'Images' section with a search bar and a list of images: 'mariadb', 'wordpress', and 'miapache'.

| Name | Tag | Image ID | Created | Size | Actions |
|-----------|----------|--------------|---------------|-----------|----------|
| mariadb | lts-ubi9 | 90e4bcac427b | 8 days ago | 465.31 MB | [Delete] |
| wordpress | latest | 5fb7d355caa4 | 9 months ago | 685.22 MB | [Delete] |
| miapache | latest | 3f4db31842cd | 10 months ago | 255.94 MB | [Delete] |

Delete image?
The 'mariadb:lts-ubi9' image is selected for deletion.