



Ejercicio 2

CIFP La Laboral - Módulo Despliegue de Aplicaciones Web

 **Autores:** Pelayo Rodríguez e Iker Pérez

 **Fecha de entrega y exposición:** Viernes, 21 de febrero de 2025.

 **Repositorio GitHub:**



ÍNDICE

1. Descargar la imagen de MariaDB
2. Verificación de la descarga
3. Crear un contenedor con MariaDB
4. Conectarse a la base de datos con DBeaver
5. Crear una nueva base de datos
6. Crear una tabla

7. Detener y eliminar contenedores

8. Verificación del volumen

9. Crear nuevo contenedor con volumen existente

10. Comprobación de persistencia de datos

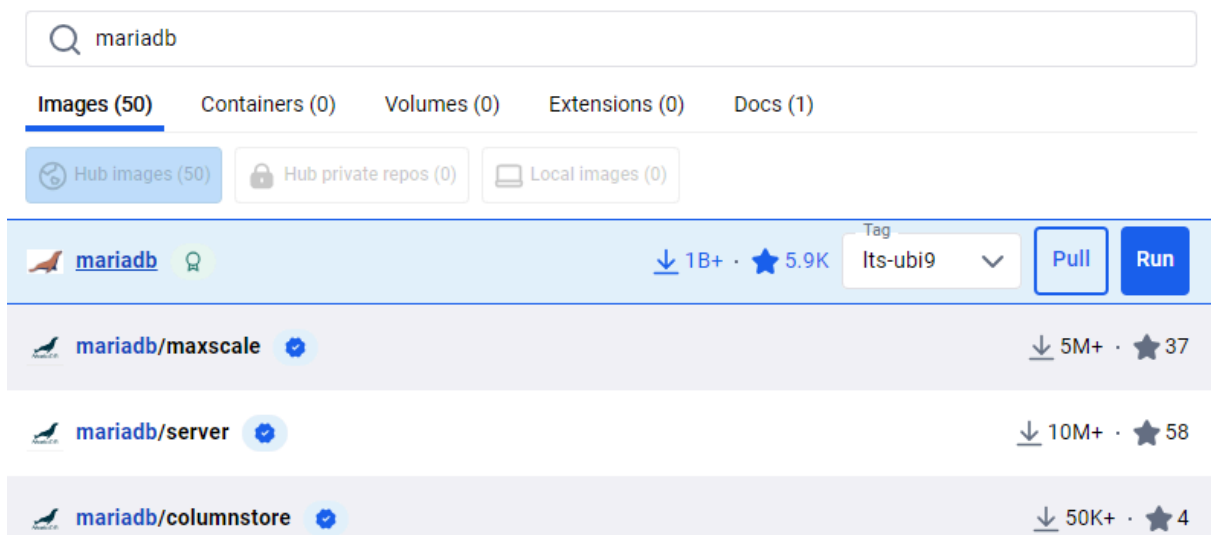
11. Intentar borrar la imagen

12. Limpieza final (Volumen, Imagen y Contenedor)

Ejercicio 2: Servidor de Base de Datos

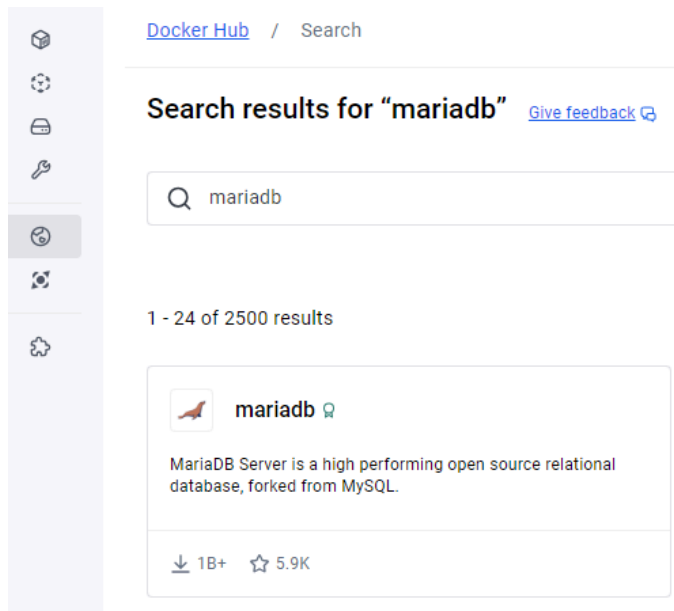
Descargar la imagen de MariaDB

- Buscamos la imagen de **mariadb** y la descargamos
- En el menú de Docker Desktop, seleccionamos la pestaña **"Images"** (Imágenes).
- En la barra de búsqueda, escribimos **"mariadb🐧"**.
- Aparecen varias opciones relacionadas con MariaDB, seleccionamos la primera.



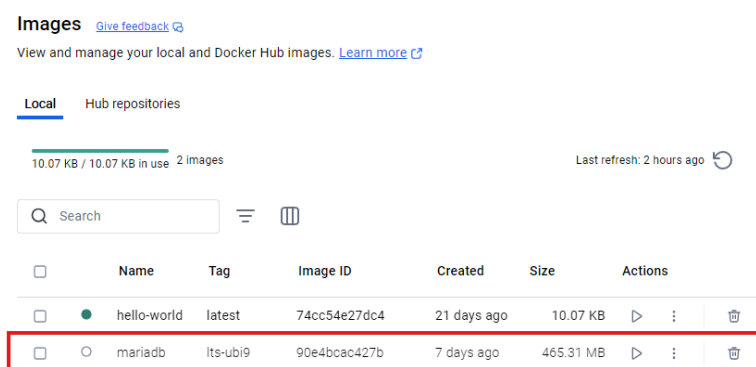
- Identificamos la imagen correcta con el nombre **"mariadb🐧"** y el ícono oficial.

- Verificamos que tiene **más de 1B+ descargas** y **5.9K estrellas** (esto confirma que es la imagen más utilizada y confiable).
- Hacemos clic en la imagen para seleccionarla.



Verificación que la imagen se descargó correctamente

- Vamos nuevamente a la pestaña "Images".
- Ahora, la imagen de MariaDB nos aparece en la lista de imágenes disponibles en tu sistema.



Crear un contenedor con MariaDB

- Vamos a la pestaña **Images** y haz clic en **Run** con la imagen de **MariaDB** que descargamos.
- Configuramos los siguientes parámetros en **Optional Settings**:

- **Nombre del contenedor:** `bbdd`
- **Puerto: 3306** (nos aseguramos de exponerlo para conexiones externas)
- **Volumen:** Creamos un volumen llamado `datos-mariadb`.
 - En **Host path**, escribimos el nombre del volumen `datos-mariadb`.
 - En **Container path**, asignamos la ruta donde MariaDB guarda los datos dentro del contenedor `/var/lib/mysql` que es la ruta por defecto donde MariaDB almacena sus datos en el contenedor.
- **Variables de entorno:**
 - `MYSQL_ROOT_USER=root`
 - `MYSQL_ROOT_PASSWORD=root`
 - `MYSQL_DATABASE=base`
 - `MYSQL_USER=daw`
 - `MYSQL_PASSWORD=daw`
- Guardamos y le damos a **Run**.



Run a new container

mariadb:its-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



Run a new container

mariadb:latest-ubi9

Optional settings

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

MYSQL_ROOT_USER

Value

root

—

Variable

MYSQL_ROOT_PASSWORD

Value

root

—

Variable

MARIADB_DATABASE

Value

base

—

Variable

MARIADB_USER

Value

daw

—

Variable

MARIADB_PASSWORD

Value

daw

+

Cancel

Run

Containers / bbdd

7ed85cbb851 mariadb:latest-ubi9
3306:3306

STATUS
Running (0 seconds ago)



Logs Inspect Bind mounts Exec Files Stats

```
2025-02-12 12:06:00 2025-02-12 11:06:00+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 11.4.5 started.
2025-02-12 12:06:00 2025-02-12 11:06:00+00:00 [Note] [Entrypoint]: Initializing database files
2025-02-12 12:06:01 2025-02-12 11:06:01+00:00 [Note] [Entrypoint]: Database files initialized
2025-02-12 12:06:01 2025-02-12 11:06:01+00:00 [Note] [Entrypoint]: Starting temporary server
2025-02-12 12:06:01 2025-02-12 11:06:01+00:00 [Note] [Entrypoint]: Waiting for server startup
2025-02-12 12:06:02 2025-02-12 11:06:02 0 [Note] Starting MariaDB 11.4.5-MariaDB source revision 077110266f5c04216af4bf1243c55f8c67ccf4 server_uid K3yTHmzkxwRQBEIhSvut04gey= as process 74
2025-02-12 12:06:02 2025-02-12 11:06:02 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
2025-02-12 12:06:02 2025-02-12 11:06:02 0 [Note] InnoDB: Number of transaction pools: 1
2025-02-12 12:06:02 2025-02-12 11:06:02 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
2025-02-12 12:06:02 2025-02-12 11:06:02 0 [Note] mariadb: O_TMPFILE is not supported on /tmp (disabling future attempts)
2025-02-12 12:06:02 2025-02-12 11:06:02 0 [Note] InnoDB: Using liburing
2025-02-12 12:06:02 2025-02-12 11:06:02 0 [Note] InnoDB: Initializing buffer pool, total size = 128.000MB, chunk size = 2.000MB
2025-02-12 12:06:02 2025-02-12 11:06:02 0 [Note] InnoDB: Completed initialization of buffer pool
2025-02-12 12:06:02 2025-02-12 11:06:02 0 [Note] InnoDB: File cache buffer for log disabled (block size=4096 bytes)
```



Conectarse a la base de datos con DBeaver

- Abrimos **DBeaver**
- Creamos una nueva conexión:

- Configuración general:

- **Host:** localhost

- **Puerto:** 3306

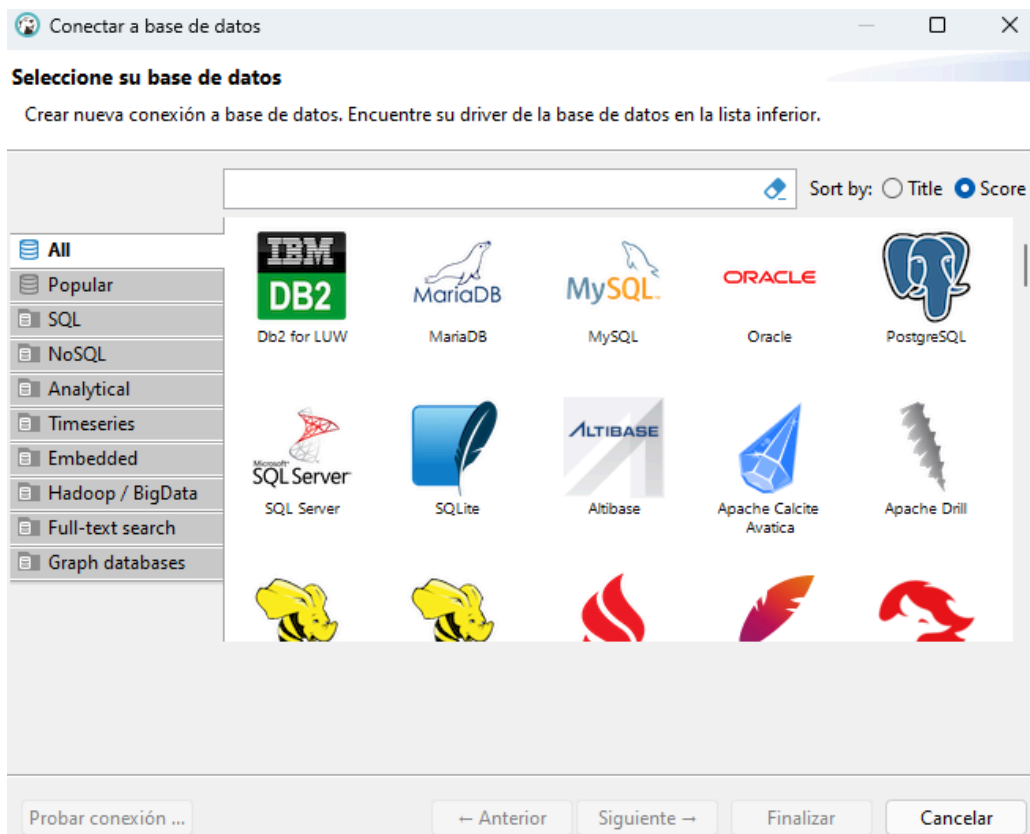
- Credenciales:

- **Usuario:** daw

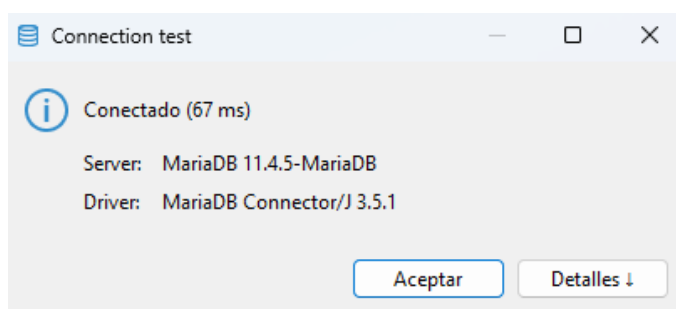
- **Contraseña:** daw

- Base de datos:

- **Base de datos:** base



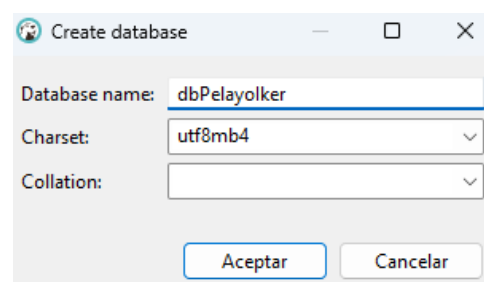
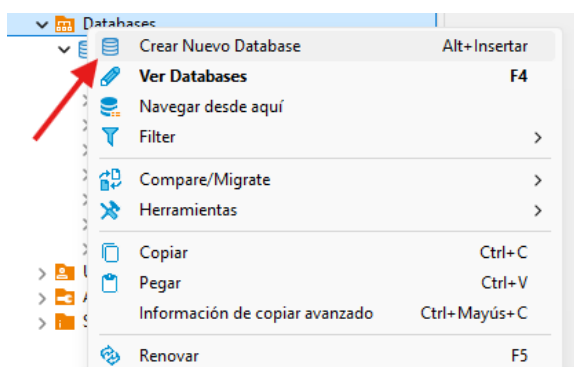
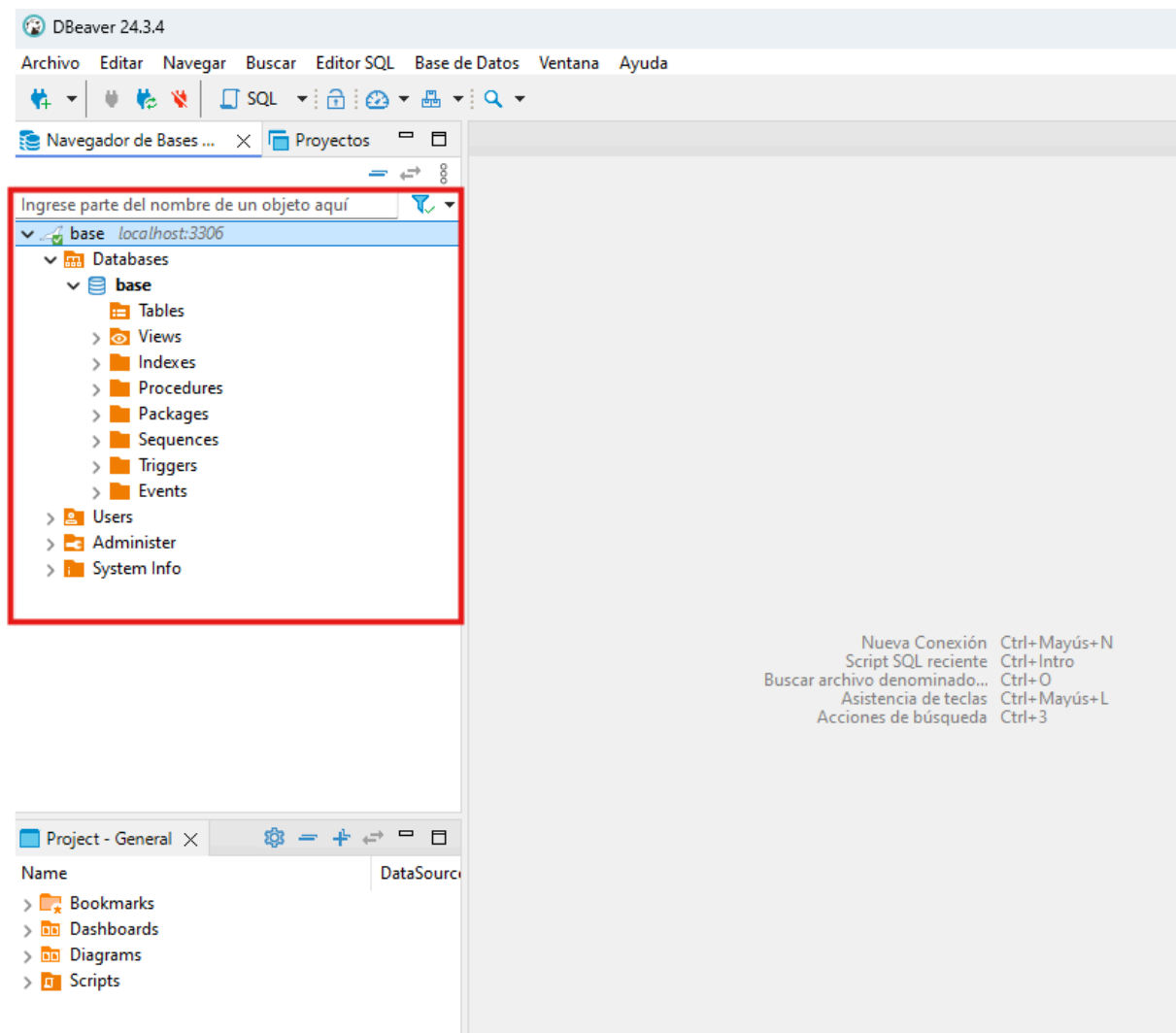
Probamos la conexión y nos aseguramos de que funciona.



Crear una nueva base de datos

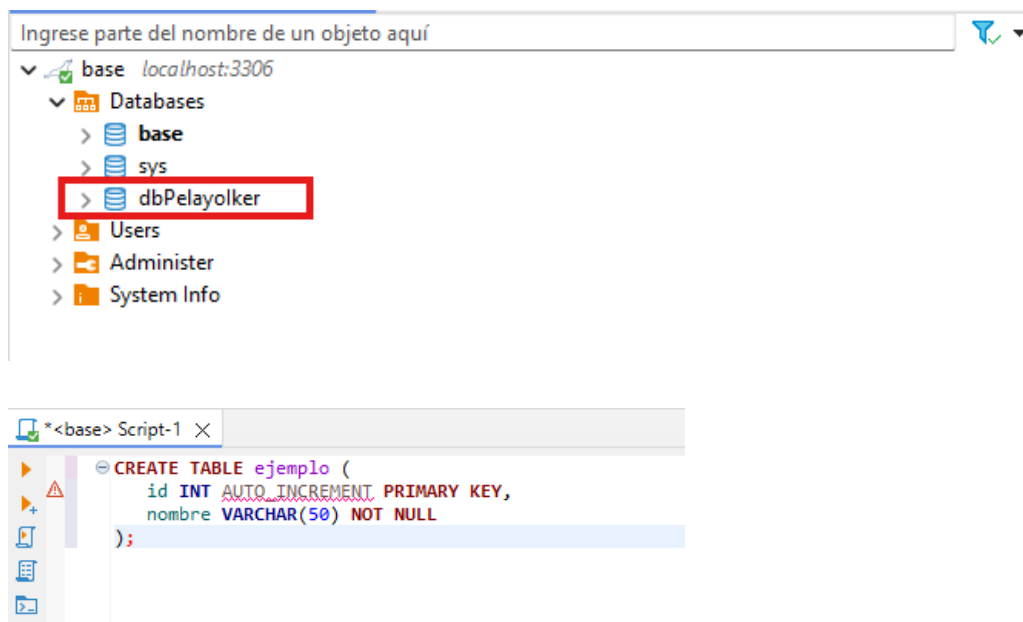
- Hacemos clic derecho en la conexión con **MariaDB** y seleccionamos **"Crear Nuevo Database"**.
- Se nos abre una ventana llamada **"Create database"**.
- En **Database name**, escribimos el nombre de la base de datos, en este caso la llamamos **dbPelayolker**.

- En **Charset**, seleccionamos **utf8mb4** (para compatibilidad con caracteres especiales y emojis).
- Opcionalmente, podíamos configurar **Collation**, pero en este caso lo dejamos vacío para la opción por defecto.
- Hacemos clic en **"Aceptar"** para crear la base de datos.



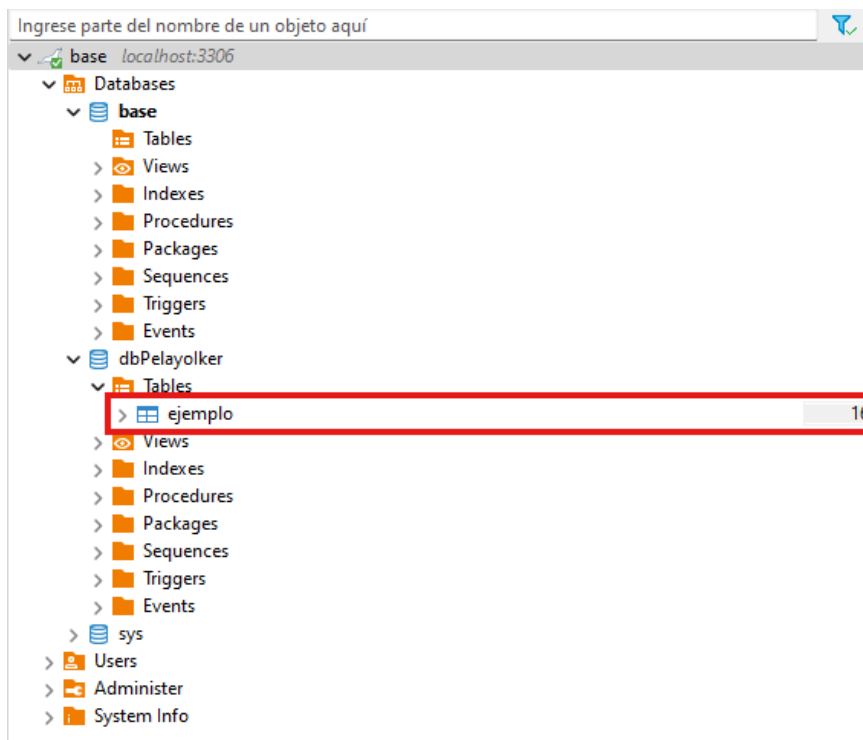
Seleccionamos la base de datos y crear una tabla

- Hacemos clic derecho sobre la base de datos `dbPelayolker` que acabamos de crear.
- Luego, hacemos clic derecho en **"Tables"** → **"Create New Table"**.
- Asignamos un nombre a la tabla , le ponemos por ejemplo, `ejemplo` .
- Agregamos las columnas necesarias:
 - `id` → **INT**, los marcamos como **Primary Key**, opción **Auto Increment**.
 - `nombre` → **VARCHAR(50)**, lo marcamos como **NOT NULL**.
- Guardamos los cambios y presionamos en **"Aceptar"**.



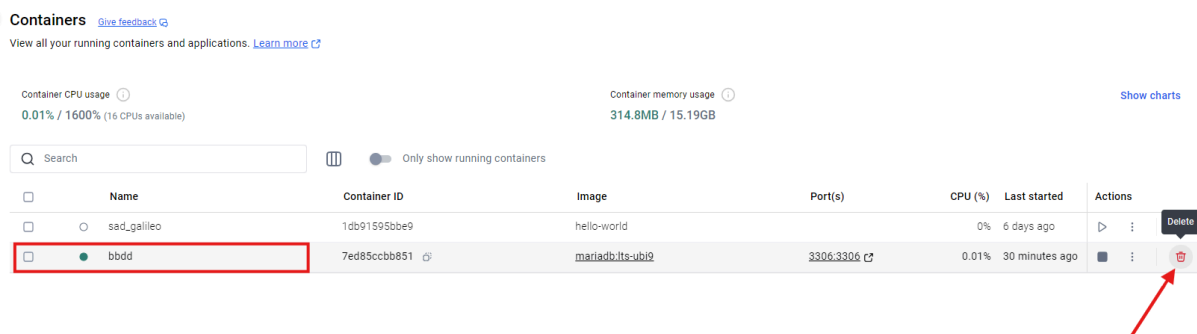
Verificamos que la tabla se creo correctamente

- Expandimos la base de datos `dbPelayolker` en el panel izquierdo y hacemos clic en **"Tables"**.
- Vemos la tabla `ejemplo` , con lo cual se ha creado correctamente.



Detener y eliminar los contenedores

- Abrimos **Docker Desktop**.
- En la pestaña **Containers**, buscamos el contenedores **bbdd**.
- Detenemos cada contenedor presionando el botón **Stop** (■).
- Luego, eliminamos los contenedores con el botón **Delete** (🗑️).



Verificamos que el volumen "datos-mariadb" sigue existiendo

- En Docker Desktop, vamos a la pestaña **Volumes**.
- Buscamos el volumen **datos-mariadb**.
- Como se ve el volumen sigue ahí, lo que significa que los datos no se han perdido.

Volumes [Give feedback](#)

Manage your volumes, view usage, and inspect their contents. [Learn more](#)

Q Search					Create
<input type="checkbox"/>	Name ↑	Created	Size	Actions	
<input type="checkbox"/>	<input type="radio"/> datos-mariadb	4 minutes ago	0 Bytes		

Creamos otro contenedor con el mismo volumen

- En Docker Desktop, vamos a **Images** y buscamos **MariaDB**.
- Hacemos clic en **Run** para desplegar un nuevo contenedor.
- Configuramos:
 - **Container name:** `bbdd-2`
 - **Port:** `3306`
 - **Volume:** `datos-mariadb`
 - **Environment variables** (como antes):
 - `MYSQL_ROOT_USER=root`
 - `MYSQL_ROOT_PASSWORD=root`
 - `MARIADB_DATABASE=base`
 - `MARIADB_USER=daw`
 - `MARIADB_PASSWORD=daw`
- Hacemos clic en **Run**.



Run a new container

mariadb:latest-ubi9

Optional settings

Container name

bbdd-2

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

MYSQL_ROOT_USER

Value

root

—

Variable

MYSQL_ROOT_PASSWORD

Value

root

—

Variable

MARIADB_DATABASE

Value

base

—

Variable

MARIADB_USER

Value

daw

—

Variable

MARIADB_PASSWORD

Value

daw

+

Cancel

Run

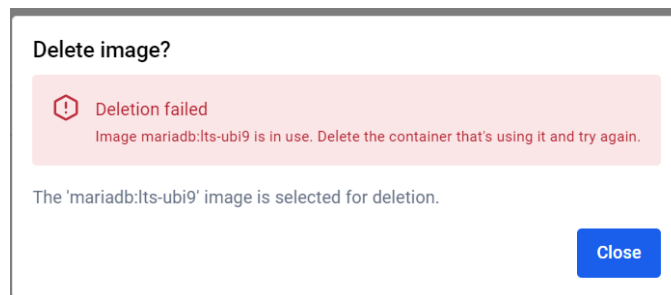
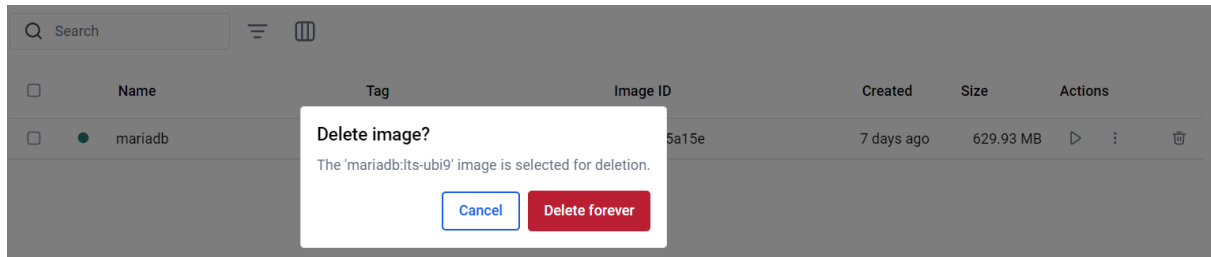
Comprobamos que los datos siguen en el nuevo contenedor

- Abrimos **DBeaver** y nos **conectamos de nuevo a MariaDB** (`localhost:3306` , usuario `daw`).
- Verificamos que `dbPelayolker` y la tabla `ejemplo` siguen existiendo.
- Como todo está correcto, los datos se han conservado gracias al volumen persistente.



Intentar borrar la imagen de MariaDB

- En Docker Desktop, vamos a la pestaña **Images**.
- Intentamos eliminar la imagen de **MariaDB** (**Delete**).
- **¿Qué sucede?**
 - No podremos eliminar si hay contenedores en ejecución usándola (**bbdd-2**).
 - Tenemos que eliminar **bbdd-2** antes de poder borrar la imagen.



Borrar todo (Volumen, Imagen y Contenedor)

- Eliminamos los contenedor **bbdd** y **bbdd-2** desde Docker Desktop.

Q	sha256:05f189b5a15eb5cc990ea3;	☰	Only show running containers						
<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions	Delete	
<input type="checkbox"/>	bbdd	94d4bd08116c	mariadb:its-ubi9	3306:3306	N/A	1 hour ago	▶ ⋮		
<input type="checkbox"/>	bbdd-2	ff1bf68bdf5d	mariadb:its-ubi9	3306:3306	N/A	16 minutes ago	▶ ⋮		

- Eliminamos el volumen **datos-mariadb** desde la pestaña **Volumes**.

<input type="checkbox"/>	datos-mariadb	17 minutes ago	0 Bytes		
--------------------------	---------------	----------------	---------	--	--

- Eliminamos la imagen de MariaDB desde la pestaña **Images**.

Q Search

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions	Delete
<input type="checkbox"/>	<div><div></div><div>mariadb</div></div>	lts-ubi9	05f189b5a15e <div></div>	7 days ago	629.93 MB	<div><div></div><div></div></div>	<div></div>