

“Bitácora para manipulación de datos en sistemas multibase”

**Asignatura:**

*TOPICOS DE BASES DE DATOS*

**UNIDAD**

“3”

**Docente:**

**AMBROSIO CARDOSO JIMENEZ**

**Alumno:**  
**HUGOLINO VALENTÍN MARTÍNEZ**

**INGENIERÍA INFORMÁTICA**

**SEMESTRE: 5º**

**GRUPO: A**

**PERIODO: Agosto-Diciembre**

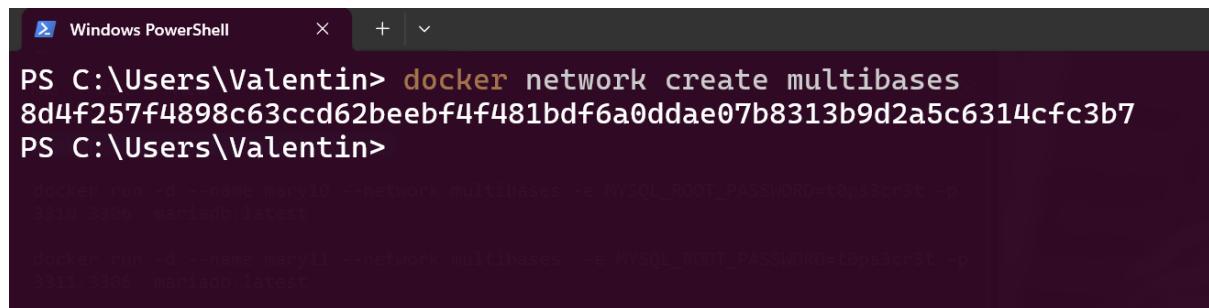
**Nº 21920117**

**Oaxaca, Oaxaca Juárez a 10 de Noviembre de  
2023**

# INDICE

<b>INDICE.....</b>	<b>2</b>
<b>Conexión entre dos bases de datos.....</b>	<b>3</b>
<b>Database Mary10.....</b>	<b>4</b>
<b>Database Mary11.....</b>	<b>6</b>
<b>MariaDB con Excel a través de ODBC.....</b>	<b>9</b>
<b>MariaDB desde Postgres dentro de Docker.....</b>	<b>12</b>
Descarga de imagen PostgreSQL.....	13
<b>MariaDB con CSV y XML.....</b>	<b>18</b>
STUDENTMOVILE.....	19
BOOKS.....	20
bookjson.....	21
<b>Postgres desde MariaDB en Docker.....</b>	<b>22</b>
<b>MariaDB desde Oracle.....</b>	<b>28</b>

# Conexión entre dos bases de datos



```
PS C:\Users\Valentin> docker network create multibases
8d4f257f4898c63cccd62beebf4f481bdf6a0ddae07b8313b9d2a5c6314cf3b7
PS C:\Users\Valentin>
docker run -d --name mary10 --network multibases -e MYSQL_ROOT_PASSWORD=t0ps3cr3t -p 3310:3306 mariadb:latest
docker run -d --name mary11 --network multibases -e MYSQL_ROOT_PASSWORD=t0ps3cr3t -p 3311:3306 mariadb:latest
```

Figura 1.0→ Crear red Docker

Creamos un red, donde se crearon los contenedores de mariadb. y ambos en la misma red.



```
PS C:\Users\Valentin>
PS C:\Users\Valentin> docker run -d --name mary10 --network multibases -e MYSQL_ROOT_PASSWORD=t0ps3cr3t -p 3310:3306 mariadb:latest
a0d87e188519d535c8aea90f06f1d7a5c91fb3d8d197d5abe9a793100435551d
PS C:\Users\Valentin>
PS C:\Users\Valentin> docker run -d --name mary11 --network multibases -e MYSQL_ROOT_PASSWORD=t0ps3cr3t -p 3311:3306 mariadb:latest
4d4717f9d9e5222a266cdff34e7c50f6f65eeb83b199f23deeab4aca8ccaff35
PS C:\Users\Valentin> _
```

Figura 1.1→ Crear contenedor mary10 y11

Creamos los contenedores, en la red llamada multibases, cual nos permite estar en misma red, y así poder hacer comunicar ambas contenedores

## Database Mary10

```
PS C:\Users\Valentin> docker exec -it mary10 bash
root@a0d87e188519:/# mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.1.2-MariaDB-1:11.1.2+maria~ubu2204 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE handcraft;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> INSERT INTO handcraft (article, description, category, image, price)
VALUES
('Gaban','Gaban hecho de 100% lana de borrego','lana','gaban_1.png', 3000),
('Unicornio','Unicornio en vuelo al infinito','madera','unicornio_1.png','unicornio_2.png', 1000),
('Búho','Búho para colorear','madera','bubo_1.png', 500);
Query OK, 3 rows affected (0.00 sec)

MariaDB [(none)]>
```

Figura 1.2→ Ingresar al contenedor mary10.

Nos conectamos a nuestra base de datos, con nuestra contraseña y usuario.

```
MariaDB [(none)]> CREATE DATABASE maria10 CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0.051 sec)

MariaDB [(none)]> USE maria10;
Query OK, 0 rows affected (0.00 sec)

Database changed
MariaDB [maria10]> INSERT INTO handcraft (article, description, category, image, price)
VALUES
('Gaban','Gaban hecho de 100% lana de borrego','lana','gaban_1.png', 3000),
('Unicornio','Unicornio en vuelo al infinito','madera','unicornio_1.png','unicornio_2.png', 1000),
('Búho','Búho para colorear','madera','bubo_1.png', 500);
Query OK, 3 rows affected (0.00 sec)

MariaDB [maria10]>
```

Figura 1.3→Crear base de datos maria10.

Creamos una base de datos llamado maria10, para luego usarlo en los siguientes puntos.

```
MariaDB [maria10]> CREATE TABLE handicraft
->     ( id BIGINT AUTO_INCREMENT
->       , article VARCHAR (100)
->       , description VARCHAR (100)
->       , category VARCHAR (50)
->       , image VARCHAR (200)
->       , price float
->     CONSTRAINT pkHandicraft PRIMARY KEY (id)
->   );
Query OK, 0 rows affected, 1 warning (0.065 sec)

MariaDB [maria10]>
```

Figura 1.4→ Crear tabla handicraft

Creamos una table, juntas con sus atributos, donde posteriormente se insertarán los datos.

```
MariaDB [maria10]>
MariaDB [maria10]> INSERT INTO handicraft (article, description, category, image, price)
-> VALUES
->     ('Gabán', 'Gabán hecho de 100% lana de borrego', 'lana', 'gaban_1.png', 3000),
->     ('Unicornio', 'Unicornio en vuelo al infinito', 'madera', 'unicornio_1.png,unicornio_2.png', 1000),
->     ('Buho', 'Buho para colorear', 'madera', 'buho_1.png', 500),
->     ('Robot programador', 'Robot de hojalatería sentado frente a una computadora imaginaria', 'hojalater
ia', 'robot_1.png,robot_2.png', 1230),
->     ('Chapulin', '', 'Chapulin al comal', 'chapulin_1.png,chapulin_2.png,chapulin_3.png', 130);
Query OK, 5 rows affected (0.054 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [maria10]>
```

Figura 1.5→ Insertar datos en la tabla handicraft

Agregamos algunos datos en la tabla handicraft, para posteriormente realizar consultas

```
MariaDB [maria10]> GRANT ALL PRIVILEGES ON maria10.* TO brandon@'%' IDENTIFIED BY '1234567890';
Query OK, 0 rows affected (0.006 sec)

MariaDB [maria10]>
```

Figura 1.6→ Conceder privilegios a brandon

Elevamos los privilegios a brandon, y ponemos su identificador, con esto hemos concedido ciertos privilegios al usuario brandon en la base de datos de maria10

## Database Mary11

```
PS C:\Users\Valentin> docker exec -it mary11 bash
root@4d4717f9d9e5:/# mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.1.2-MariaDB-1:11.1.2+maria~ubu2204 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use handcraft
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> INSERT INTO handicraft (article, description, category, image, price)
    VALUES
    ('Unicornio', 'Unicornio hecho de 100% lana de corregidora', 'lana', 'guate_1.png', 2000),
    ('Unicornio', 'Unicornio en vuelo al infinito', 'madera', 'unicornio_1.png',1000),
    ('Busto', 'Busto para colorear', 'madera', 'busto_1.png', 100),
    ('Robot programable', 'Robot de hojalata sentado frente a una computadora',
    'juguetes', 'robot_1_programable_1.png', 120),
    ('Chapulin', 'Chapulin al costal', 'chapulin_1.png', 'chapulin_2.png', chapulin_1),
    ('Chapulin', 'Chapulin al costal', 'chapulin_1.png', 'chapulin_2.png', chapulin_1)
```

Figura 1.7→ Crear base de datos test en mary11 y configurar CONNECT.

Por otra lado, en Mary nos logueamos con nuestros usuarios y contraseña.

```
MariaDB [(none)]> CREATE DATABASE test CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0.053 sec)

MariaDB [(none)]> use test;
Database changed
MariaDB [test]> INSTALL SONAME 'ha_connect';
Query OK, 0 rows affected (0.00 sec)

MariaDB [test]> SHOW VARIABLES LIKE 'plugin';
+----+-----+
| Variable_name | Value |
+----+-----+
| plugin       | mysql |
+----+-----+
```

Figura 1.8→ Desde el contenedor mary11.

luego creamos un base de datos llamado test, y luego nos conectamos.

```
MariaDB [test]> INSTALL SONAME 'ha_connect';
Query OK, 0 rows affected (0.012 sec)

MariaDB [test]>
```

Figura 1.9→ Instalar el plugin CONNECT si es necesario.

Instalamos el ha connect, para que esto nos pueda conectar a otros contenedores.

```
MariaDB [test]> CREATE TABLE handicraft ENGINE=CONNECT TABLE_TYPE=MYSQL CONNECTION='mysql://brandon:1234567890@mary10:3306/maria10/handicraft';
Query OK, 0 rows affected (0.029 sec)

MariaDB [test]>
```

Figura 2.0→ # Crear tabla conectada desde mary11 a mary10

aquí creamos una tabla desde mary11 a mary10, si la conexión es exitosa habremos creado una tabla handcraft.

```
root@a0d87e188519: /      X | root@4d4717f9d9e5: /      X | Windows PowerShell      X | + | 
ngrok

Introducing Always-On Global Server Load Balancer: https://ngrok.com/r/gslb

Session Status          online
Account                 Hugolino Valentin (Plan: Free)
Version                3.3.5
Region                 United States (us)
Latency                104ms
Web Interface          http://127.0.0.1:4040
Forwarding             tcp://4.tcp.ngrok.io:16709 -> localhost:3310

Connections            ttl     opn         rt1        rt5       p50       p90
                      10      0          0.03      0.02      0.23      2.64
```

Figura 2.1→ Activamos Ngrok, para poder conectarse desde fuera.

Esto nos permite conectarnos de diferentes partes del mundo.

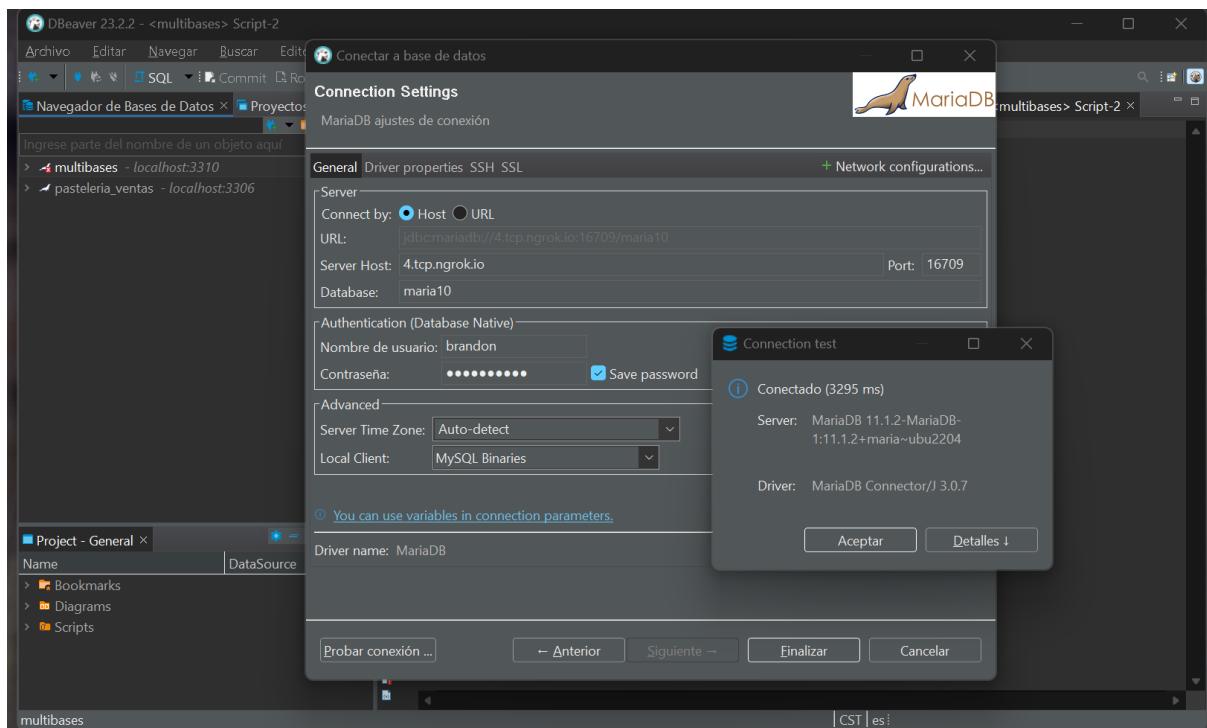
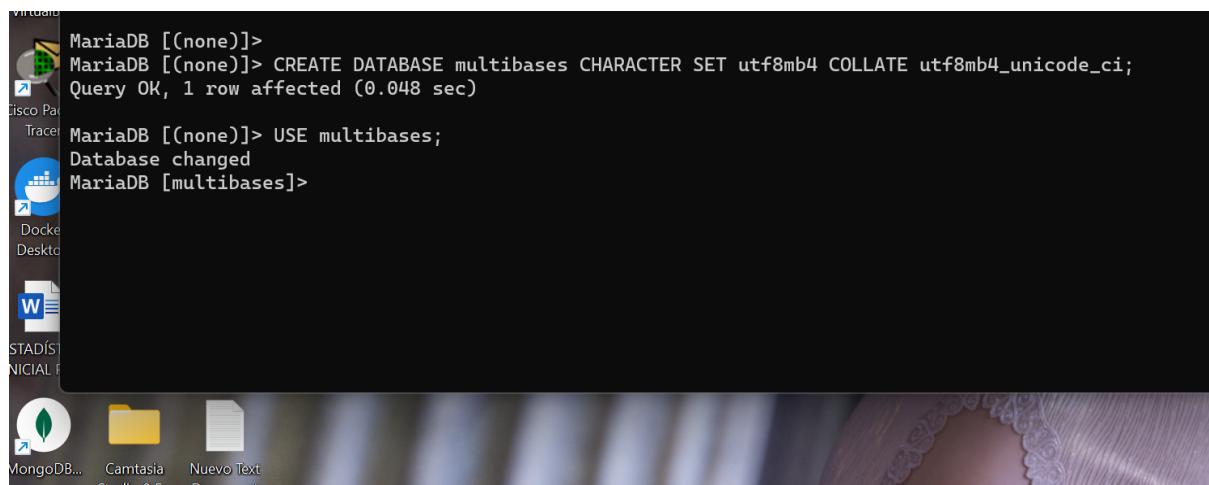


Figura 2.2→ Identificamos con nuestro datos y probamos conexión.

Nos conectamos a la base de datos, ponemos los datos, y esto nos conecta a la base de datos de mary 10, desde el otro ordenador.

# MariaDB con Excel a través de ODBC

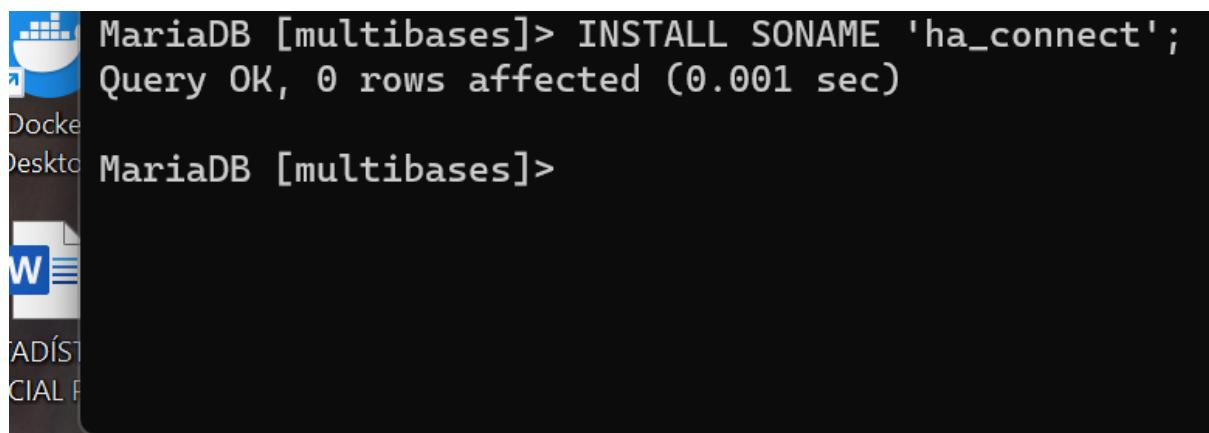


```
MariaDB [(none)]>
MariaDB [(none)]> CREATE DATABASE multibases CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0.048 sec)

MariaDB [(none)]> USE multibases;
Database changed
MariaDB [multibases]>
```

Figura 1→ Creación de la base de datos

Creamos una base de datos, donde se almacenarán los datos de excel, al momento de consultar datos, luego entramos a la base de datos.



```
MariaDB [multibases]> INSTALL SONAME 'ha_connect';
Query OK, 0 rows affected (0.001 sec)

MariaDB [multibases]>
```

Figura 2→ instalar el ha\_connect

Esto para que la base de datos se conecte a excel

```
ace1 MariaDB [multibases]> CREATE TABLE info7b (
->     nocontrol VARCHAR(20),
->     nombres VARCHAR(100),
->     correo VARCHAR(100)
-> ) ENGINE=CONNECT
->     TABLE_TYPE=ODBC
->     TABNAME='I7B'
->     CONNECTION='DRIVER={Microsoft Excel Driver (*.xls)};DSN=NombreDSNExcel;DBQ=c:/docs/tbd_i7b.xls;ReadOnly=0';
Query OK, 0 rows affected (0.066 sec)

MariaDB [multibases]>
```

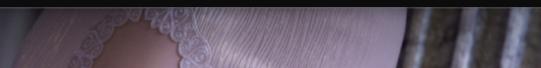


Figura 3→ creamos una tabla

Esta tabla se conecta al archivo excel, que previamente se ha creado y guardado en windows.

```
MariaDB [multibases]> SELECT * FROM info7b;
+-----+-----+
| nocontrol | nombres           | correo          |
+-----+-----+
| 18920019 | DIEGO VASQUEZ JENNIFER | L18920019@voaxaca.tecnm.mx |
| 18920020 | FERNANDEZ LOPEZ FABIAN | L18920020@voaxaca.tecnm.mx |
| 17920331 | HERNANDEZ CRUZ DIANA ALEJANDRA | L17920331@voaxaca.tecnm.mx |
| 18920032 | LOPEZ VELASCO JOSE ANTONIO | L18920032@voaxaca.tecnm.mx |
| 19920307 | MARTINEZ BARRAGAN JESUS ALBERTO | L19920307@voaxaca.tecnm.mx |
| 18920039 | MORENO FLORES ANA CELINA | L18920039@voaxaca.tecnm.mx |
| 18920040 | OROZCO VASQUEZ MARCO ANTONIO | L18920040@voaxaca.tecnm.mx |
| 18920042 | PABLO MARTINEZ MIGUEL ANGEL | L18920042@voaxaca.tecnm.mx |
| 18920044 | PEDRO BOHORQUEZ LUIS DANIEL | L18920044@voaxaca.tecnm.mx |
| 18920055 | SOLANO CRUZ MARIO | L18920055@voaxaca.tecnm.mx |
| C15920382 | VASQUEZ MARTINEZ DANIEL | LC15920382@voaxaca.tecnm.mx |
+-----+
11 rows in set (0.001 sec)

MariaDB [multibases]>
```



Figura 4→ Hacemos una consulta.

Esta consulta se obtiene desde excel, es decir que los datos lo estamos jalando desde el archivo excel que contiene ese datos que se muestra en la figura 4.

```
MariaDB [multibases]> INSERT INTO info7b (nocontrol, nombres, correo) VALUES ('999999', 'Ambrosio Cardoso Jimenez', 'ambrosio.cj@voaxaca.tecnm.mx');
Query OK, 1 row affected (0.007 sec)

MariaDB [multibases]>
```

Figura 5→ insertar datos

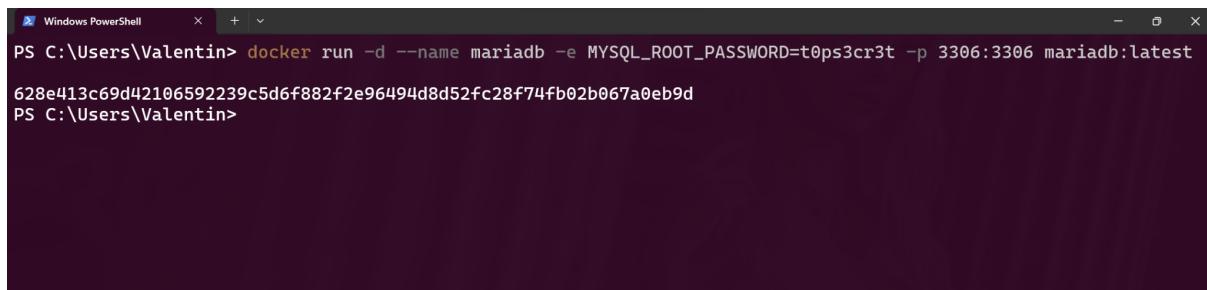
Se insertan los datos en el archivo excel, a través de mariadb.

	A	B	C	D	E	F	G
1	NOCONTROL	NOMBRES	CORREO				
2	18920019	DIEGO VASQUEZ JENNIFER	L18920019@voaxaca.tecnm.mx				
3	18920020	FERNANDEZ LOPEZ FABIAN	L18920020@voaxaca.tecnm.mx				
4	17920331	HERNANDEZ CRUZ DIANA ALEJANDRA	L17920331@voaxaca.tecnm.mx				
5	18920032	LOPEZ VELASCO JOSE ANTONIO	L18920032@voaxaca.tecnm.mx				
6	19920307	MARTINEZ BARRAGAN JESUS ALBERTO	L19920307@voaxaca.tecnm.mx				
7	18920039	MORENO FLORES ANA CELINA	L18920039@voaxaca.tecnm.mx				
8	18920040	OROZCO VASQUEZ MARCO ANTONIO	L18920040@voaxaca.tecnm.mx				
9	18920042	PABLO MARTINEZ MIGUEL ANGEL	L18920042@voaxaca.tecnm.mx				
10	18920044	PEDRO BOHORQUEZ LUIS DANIEL	L18920044@voaxaca.tecnm.mx				
11	18920055	SOLANO CRUZ MARIO	L18920055@voaxaca.tecnm.mx				
12	C15920382	VASQUEZ MARTINEZ DANIEL	LC15920382@voaxaca.tecnm.mx				
13	999999	Ambrosio Cardoso Jimenez	ambrosio.cj@voaxaca.tecnm.mx				
14							
15							
16							
17							
18							
19							
20							
21							

Figura 6 → Finalmente abrimos el archivo excel

Y así queda el archivo, con los datos insertados, y después de ejecutar estos pasos, la tabla info7b en la base de datos MariaDB, la cual está vinculada al archivo Excel mediante ODBC. Se puede realizar consultas y operaciones en la tabla como lo se haría con cualquier otra tabla de MariaDB.

# MariaDB desde Postgres dentro de Docker



```
PS C:\Users\Valentin> docker run -d --name mariadb -e MYSQL_ROOT_PASSWORD=t0ps3cr3t -p 3306:3306 mariadb:latest
628e413c69d42106592239c5d6f882f2e96494d8d52fc28f74fb02b067a0eb9d
PS C:\Users\Valentin>
```

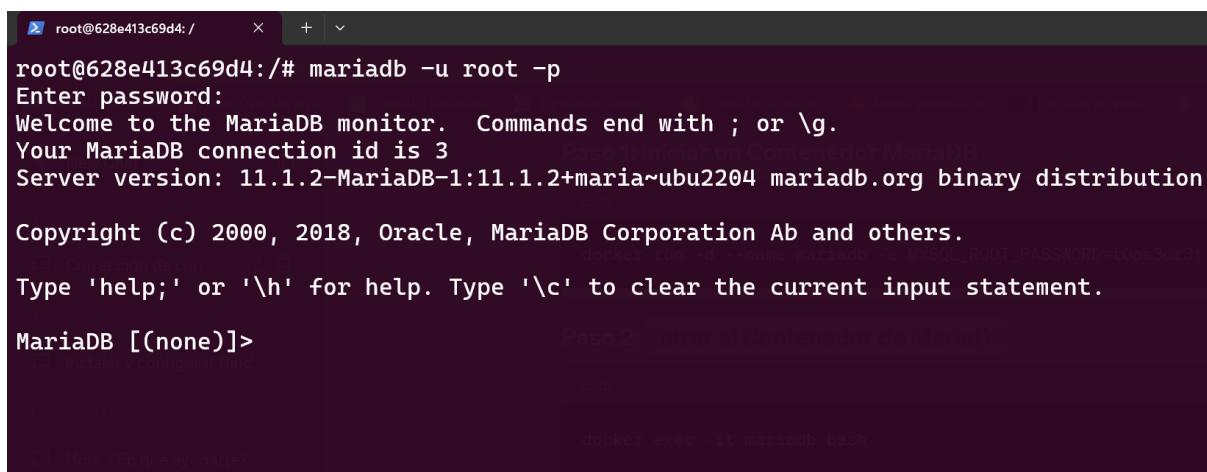
Figura 1→ **Iniciar un Contenedor MariaDB**

Creamos un contenedor de mariadb, y obtenemos la versión más reciente.

```
PS C:\Users\Valentin> docker exec -it mariadb bash
root@628e413c69d4:/#
```

Figura 2→ **Entrar al Contenedor de MariaDB**

Entramos al contenedor con el comando que se muestra en la figura 2.



```
root@628e413c69d4:/# mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.1.2-MariaDB-1:11.1.2+maria~ubu2204 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Figura 3→ **Iniciar la Sesión de MariaDB como Root**

Inicia sesión en la base de datos MariaDB como usuario root.

```
MariaDB [(none)]> CREATE DATABASE itvo CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> USE itvo;CREATE TABLE test (id INT AUTO_INCREMENT PRIMARY KEY, data VARCHAR(100));
Database changed
Query OK, 0 rows affected (0.049 sec)

MariaDB [itvo]> GRANT ALL PRIVILEGES ON itvo.* TO cardoso@'%' IDENTIFIED BY 't0ps3cr3t';
Query OK, 0 rows affected (0.004 sec)

MariaDB [itvo]> INSERT INTO test (data) VALUES ('First data'), ('Second data'), ('Third data');
Query OK, 3 rows affected (0.009 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [itvo]> SELECT * FROM test;
+----+-----+
| id | data   |
+----+-----+
| 1  | First data |
| 2  | Second data |
| 3  | Third data  |
+----+-----+
3 rows in set (0.000 sec)

MariaDB [itvo]>
```

## Figura 4→ Crear Base de Datos, Usuario e Insertar Datos

Crea una base de datos llamada 'itvo', una tabla 'test', y un usuario 'cardoso' con privilegios en la base de datos 'itvo'. Luego, inserta algunos datos en la tabla 'test'.

[Descarga de imagen PostgreSQL](#)

```
PS C:\Users\Valentin> docker pull toleg/postgres_mysql_fdw
Using default tag: latest
latest: Pulling from toleg/postgres_mysql_fdw
bb263680fed1: Pull complete
75a54e59e691: Pull complete
3ce7f8df2b36: Pull complete
f30287ef02b9: Pull complete
dc1f0e9024d8: Pull complete
7f0a68628bce: Pull complete
32b11818cae3: Pull complete
48111fe612c1: Pull complete
fcedb9c04393: Pull complete
8943748d4e1f: Pull complete
204b98eddef7: Pull complete
9e0624990483: Pull complete
01ebe7b28449: Pull complete
a90c050ffe7: Pull complete
06fb8c675ba: Pull complete
Digest: sha256:a4424ee680b3511286905669de8be62d901d34f9771b8b3735edc6e9550598c2
Status: Downloaded newer image for toleg/postgres_mysql_fdw:latest
docker.io/tolog/postgres_mysql_fdw:latest

What's Next?
 1. Sign in to your Docker account → docker login
 2. View a summary of image vulnerabilities and recommendations → docker scout q
fdw
PS C:\Users\Valentin>
```

## Figura 5→ Descargar la Imagen de PostgreSQL con mysql\_fdw

Descarga la imagen de PostgreSQL con la extensión mysql\_fdw

```
PS C:\Users\Valentin> docker run --name postgres14 --network tbd -e POSTGRES_PASSWORD=t0ps3cr3t -p 5432:5432 -v
pgdata:/var/lib/postgresql/data -d toleg/postgres_mysql_fdw
4c9594f8645c76532e6c25dc10aa1efe90a72381aa22604b6d77c32f1d67a291
PS C:\Users\Valentin>
```

## Figura 6→ Crear el Contenedor PostgreSQL con mysql\_fdw

Crea un contenedor de PostgreSQL con la extensión mysql\_fdw, configurando la contraseña y puertos.

```
[REDACTED]  
PS C:\Users\Valentin> docker exec -it postgres14 bash  
root@4c9594f8645c:/# su postgres  
postgres@4c9594f8645c:/$ psql  
psql (15.2 (Debian 15.2-1.pgdg110+1))  
Type "help" for help.  
  
postgres=#
```

## Figura 7→ Entrar al Contenedor y Ejecutar psql

Accede al shell del contenedor de PostgreSQL y luego al cliente psql.

```
postgres=# CREATE USER cardoso_postgres WITH PASSWORD 't0ps3cr3t' SUPERUSER;  
CREATE ROLE  
postgres=# CREATE DATABASE prueba_pos OWNER cardoso_postgres;  
CREATE DATABASE  
postgres=#  
[REDACTED] Hola, ¿En qué ayudarte?  
[REDACTED] postgres=#
```

docker exec -it postgres14 bash  
su postgres  
psql

## Figura 8→ Crear Usuario y Base de Datos en PostgreSQL

Crea un usuario y una base de datos en PostgreSQL.

```
postgres@43c4b02e3f2d:/# psql -h 127.0.0.1 -p 5432 -d prueba_pos -U cardoso_postgres  
psql (15.2 (Debian 15.2-1.pgdg110+1))  
Type "help" for help.  
  
prueba_pos=#
```

## Figura 9→ Conectar con la Cuenta de cardoso\_postgres en PostgreSQL

Conecta a la base de datos PostgreSQL con el usuario creado.

```
prueba_pos=# CREATE EXTENSION IF NOT EXISTS mysql_fdw;  
CREATE EXTENSION  
prueba_pos=#
```

### Figura 10→ Crear extensión mysql\_fdw

- Habilita la extensión mysql\_fdw en PostgreSQL y configura la conexión con el servidor MariaDB.

```
prueba_pos=#  
prueba_pos=# CREATE SERVER mariadb_server  
FOREIGN DATA WRAPPER mysql_fdw  
OPTIONS (host '172.19.0.2', port '3306');  
CREATE SERVER  
prueba_pos=#
```

### Figura 11→ Configurar el servidor MySQL FDW:

```
prueba_pos=#  
prueba_pos=# CREATE USER MAPPING FOR cardoso_postgres  
SERVER mariadb_server  
OPTIONS (username 'cardoso', password 't0ps3cr3t');  
CREATE USER MAPPING  
prueba_pos=#
```

### Figura 12→ CREACIÓN DE MAPPING

Se crea un usuario mapping para cardoso, que llama a mariadbserver y se loguea con la contraseña y el usuario.

```
postgres=# CREATE FOREIGN TABLE test
  ( id INT8
    , data VARCHAR (200)      Walkthroughs
  )
  SERVER mariadb_server OPTIONS (dbname 'itvo', table_name 'test');
CREATE FOREIGN TABLE
postgres=#

```

**Figura 13→ creación de tabla TEST**

se crea una tabla donde van los datos de la base de datos, de itvo y la tabla de test que está dentro de test.

```
prueba_pos=#
prueba_pos=# SELECT * FROM test;
  id |     data
-----+
  1 | First data
  2 | Second data
  3 | Third data
(3 rows)

O ACTIVIDADES
prueba_pos=#
  O Tarea_3_1_Bilacora para ...
Paso 11: Consu
```

**Figura 14→Consulta a la tabla test**

Se hace un consulta a la tabla test desde postgres a MARIADB., y vemos que es exitoso la consulta.

# MariaDB con CSV y XML

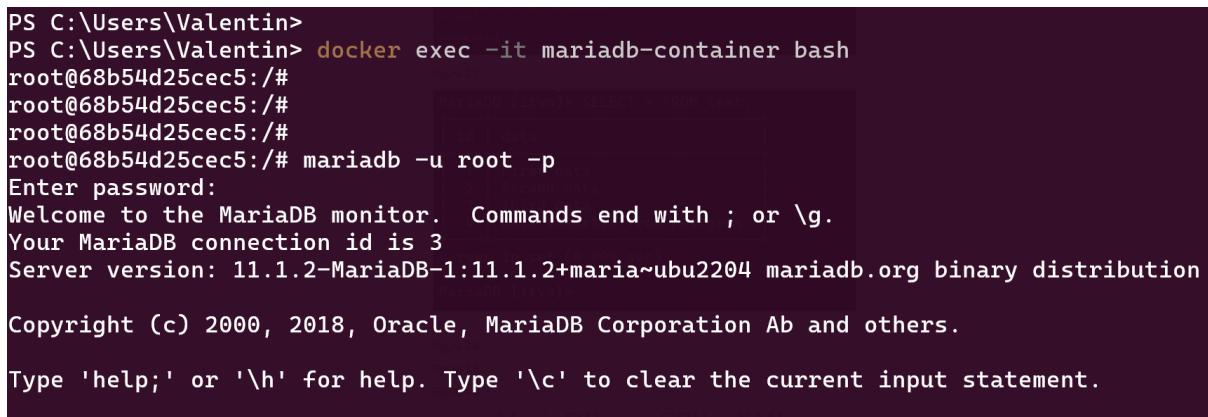


```
PS C:\Users\Valentin> docker run --name mariadb-container -e MYSQL_ROOT_PASSWORD=t0ps3cr3t -d -p 3306:3306 -v C:\tdb:/mnt/shared mariadb
68b54d25cec58aa7f84f3d7bc9a2d8e61e4099694d00df508d198ed308c8a707
PS C:\Users\Valentin>
PS C:\Users\Valentin>
```

## Figura→1 Configurar un contenedor Docker para MariaDB

Utilizamos Docker para crear un contenedor de MariaDB, un sistema de gestión de bases de datos.

Establecemos una contraseña para el usuario root de la base de datos.



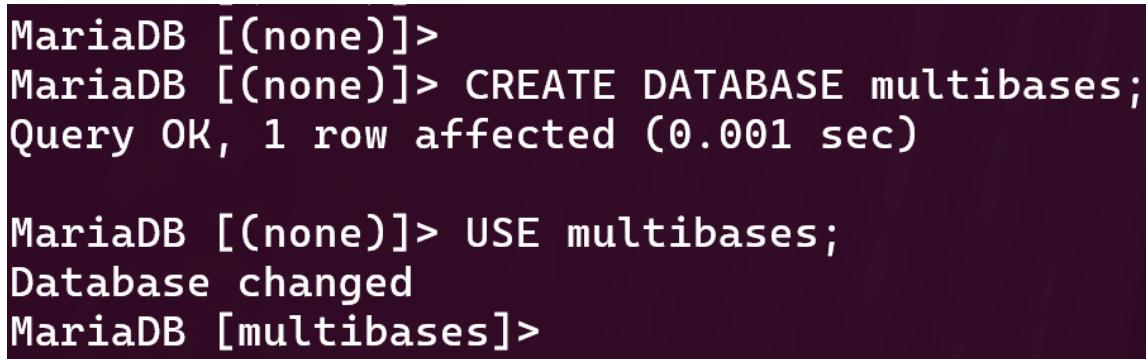
```
PS C:\Users\Valentin>
PS C:\Users\Valentin> docker exec -it mariadb-container bash
root@68b54d25cec5:#
root@68b54d25cec5:#
root@68b54d25cec5:#
root@68b54d25cec5:#
root@68b54d25cec5:#
root@68b54d25cec5:#
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.1.2-MariaDB-1:11.1.2+maria~ubu2204 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

## Figura→2 Entramos al contenedor

Entramos al mariadb, para crear las bases de datos necesarios



```
MariaDB [(none)]>
MariaDB [(none)]> CREATE DATABASE multibases;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> USE multibases;
Database changed
MariaDB [multibases]>
```

## Figura→3 Creación de base de datos

Creamos una base de datos, desde donde estaremos trabajando, y alistando los archivos.

## STUDENTMOVILE

```
MariaDB [multibases]>
MariaDB [multibases]> CREATE TABLE studentmobile (
    ->     controlnumber BIGINT,
    ->     fullname VARCHAR(100),
    ->     email VARCHAR(100),
    ->     career VARCHAR(50)
    -> ) engine=CONNECT table_type=CSV file_name='/mnt/shared/moviles.csv' header=1 sep_char='
| ' quoted=1;
Query OK, 0 rows affected (0.098 sec)
```

### Figura→ 4 Crear una tabla en MariaDB para los datos CSV

Utilizamos el comando SQL para conectarnos a MariaDB y crear una tabla (studentmobile) que utiliza el formato CSV.

Configuramos detalles como el nombre del archivo, el encabezado y el carácter de separación.

```
MariaDB [multibases]> SELECT * FROM studentmobile;
+-----+-----+-----+-----+
| controlnumber | fullname | email | career |
+-----+-----+-----+-----+
| 17920315 | CARDENAS VASQUEZ BRALIO | L17920315@voaxaca.tecnm.mx | INGENIERÍA INFORMÁTICA |
| 17920317 | CARRIZOSA BUSTAMANTE JOSE MANUEL | L17920317@voaxaca.tecnm.mx | INGENIERÍA INFORMÁTICA |
| 17920318 | CASAS PEREZ LIZBETH VERONICA | L17920318@voaxaca.tecnm.mx | INGENIERÍA INFORMÁTICA |
| 17920321 | CHAVEZ OLIVERA MARIO YOBANY | L17920321@voaxaca.tecnm.mx | INGENIERÍA INFORMÁTICA |
| 17920324 | CURIEL SANTIAGO LUCILA | L17920324@voaxaca.tecnm.mx | INGENIERÍA INFORMÁTICA |
| 17920333 | HERNANDEZ MORALES KEVIN UZIEL | L17920333@voaxaca.tecnm.mx | INGENIERÍA INFORMÁTICA |
| 17920339 | LOPEZ ZERMEÑO EMMANUEL JOSUE | L17920339@voaxaca.tecnm.mx | INGENIERÍA INFORMÁTICA |
| 17920343 | MARTINEZ JIMENEZ DAMIAN | L17920343@voaxaca.tecnm.mx | INGENIERÍA INFORMÁTICA |
| 17920352 | PROCOPIO GONZALEZ HERMAS | L17920352@voaxaca.tecnm.mx | INGENIERÍA INFORMÁTICA |
| 17920359 | ROMERO MONTERROZA GUSTAVO | L17920359@voaxaca.tecnm.mx | INGENIERÍA INFORMÁTICA |
+-----+-----+-----+-----+
10 rows in set (0.080 sec)

MariaDB [multibases]>
```

### Figura→5 Consulta a studentmobile

Consultamos a la tabla, y vemos que si nos devuelve los datos, desde la carpeta compartida, gracias a que docker permite hacer conexión con windows desde donde se encuentra el archivo.

## BOOKS

```
MariaDB [multibases]>
MariaDB [multibases]> CREATE TABLE books (
->     id VARCHAR(20) xpath='@id',
->     author VARCHAR(100),
->     title VARCHAR(100),
->     genre VARCHAR(50),
->     price FLOAT,
->     publish_date DATE,
->     description VARCHAR(1000)
-> ) engine=CONNECT table_type=XML file_name='/mnt/shared/books.xml' tablename='catalog' option_list='rownode=book';
Query OK, 0 rows affected (0.029 sec)

MariaDB [multibases]>
MariaDB [multibases]>
```

BOOKS

## Figura→6 Crear una tabla en MariaDB para los datos XML

Utilizamos el comando SQL para crear otra tabla (book) que utiliza el formato XML.

Especificamos detalles como el nombre del archivo, el nodo de fila y los campos que deben extraerse del XML.

```
MariaDB [multibases]>
MariaDB [multibases]>
MariaDB [multibases]> SELECT * FROM books;
+-----+-----+-----+-----+
| id   | author          | title        | genre | pr |
+-----+-----+-----+-----+
| NULL | Gambardella, Matthew | Computer      | NULL  | 44 |
| NULL | Ralls, Kim       | Fantasy       | NULL  | 5   |
| NULL | Corets, Eva      | Fantasy       | NULL  | 5   |
| NULL | Corets, Eva      | Fantasy       | NULL  | 5   |
| NULL | Corets, Eva      | Fantasy       | NULL  | 5   |
| NULL | Corets, Eva      | Fantasy       | NULL  | 5   |
```

## Figura→7 Consulta a books

Consultamos a la tabla, y vemos que si nos devuelve los datos, desde la carpeta compartida, gracias a que docker permite hacer conexión con windows desde donde se encuentra el archivo.

```
MariaDB [multibases]>
MariaDB [multibases]> SELECT * FROM books;
+----+-----+-----+-----+-----+-----+
| id | author | title | genre | price | publish_date | description |
+----+-----+-----+-----+-----+-----+
| NULL | GambardeLLA, Matthew | Computer | NULL | 44.95 | 1970-01-01 | An in-depth look at creating applications with XML.
| NULL | Ralls, Kim | Fantasy | NULL | 5.95 | 1970-01-01 | A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.
| NULL | Corets, Eva | Fantasy | NULL | 5.95 | 1970-01-01 | After the collapse of a nanotechnology society in England, the young survivors lay the foundation for a new society.
| NULL | Corets, Eva | Fantasy | NULL | 5.95 | 1970-01-01 | In post-apocalyptic England, the mysterious agent known only as Oberon helps to create a new life for the inhabitants of London. Sequel to Maeve's Ascendant.
| NULL | Corets, Eva | Fantasy | NULL | 5.95 | 1970-01-01 | The two daughters of Maeve, half-sisters, battle one another for control of England. Sequel to Oberon's Legacy.
| NULL | Randall, Cynthia | Romance | NULL | 4.95 | 1970-01-01 | When Carla meets Paul at an ornithology conference, tempers fly as feathers get ruffled.
| NULL | Thurman, Paula | Romance | NULL | 4.95 | 1970-01-01 | A deep sea diver finds true love twenty thousand leagues beneath the sea.
| NULL | Knorr, Stefan | Horror | NULL | 4.95 | 1970-01-01 | An anthology of horror stories about roaches, centipedes, scorpions and other insects.
| NULL | Kress, Peter | Science Fiction | NULL | 6.95 | 1970-01-01 | After an inadvertent trip through a Heisenberg Uncertainty Device, James Salway discovers the problems of being quantum.
| NULL | O'Brien, Tim | Computer | NULL | 36.95 | 1970-01-01 | Microsoft's .NET initiative is explored in detail in this deep programmer's reference.
| NULL | O'Brien, Tim | Computer | NULL | 36.95 | 1970-01-01 | The Microsoft MSXML3 parser is covered in detail, with attention to XML DOM interfaces, XSLT processing, SAX and more.
| NULL | Galos, Mike | Computer | NULL | 49.95 | 1970-01-01 | Microsoft Visual Studio 7 is explored in depth, looking at how Visual Basic, Visual C++, C#, and ASP+ are integrated into a comprehensive development environment.
+----+-----+-----+-----+-----+-----+
12 rows in set (0.018 sec)

MariaDB [multibases]>
```

## Figura→8 Visto desde lejos

Aquí podemos ver la tabla completa.

## bookjson

```
MariaDB [multibases]> CREATE TABLE bookjson (
->     isbn CHAR(15),
->     lang CHAR(2),
->     subject CHAR(32),
->     author CHAR(128),
->     title CHAR(32),
->     translated CHAR(80),
->     publisher CHAR(20),
->     yearofpublication INT(4)
-> ) engine=CONNECT table_type=JSON file_name='/mnt/shared/books.json';
Query OK, 0 rows affected (0.075 sec)

MariaDB [multibases]>
```

## Figura→9 Crear una tabla en MariaDB para los datos JSON

Utilizamos el comando SQL para crear una tabla (bookjson) que utiliza el formato JSON.

Especificamos detalles como el nombre del archivo y los campos que deben extraerse del JSON.

```
MariaDB [multibases]>
MariaDB [multibases]> SELECT * FROM bookjson;
+-----+-----+-----+-----+-----+-----+-----+-----+
| isbn | lang | subject | author | title | translated | publisher | yearofpublication |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 9782212899819 | fr | applications | Jean-Christophe Bernadac, François Knab | Construire une application XML | NULL | Eyrolles Paris | NULL |
| 9782840825685 | fr | applications | William J. Pardi | XML en Action | adapté de l'anglais par James Guerin | Microsoft Press Pari | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 2 warnings (0.018 sec)

MariaDB [multibases]>
```

## Figura→10 Consultas

Consultamos a la tabla, y vemos que si nos devuelve los datos, desde la carpeta compartida, gracias a que docker permite hacer conexión con windows desde donde se encuentra el archivo.

## Postgres desde MariaDB en Docker

```
PS C:\Users\Valentin> docker network create PosForMary
f01b3287533355235337ac9d5203864ee71d65661db656b1e86bf24662a5cebc
PS C:\Users\Valentin>
```

## Figura 1→ Crear una red de Docker

Creamos un red, donde se crearon los contenedores de mariadb. y ambos en la misma red.

```
PS C:\Users\Valentin>
PS C:\Users\Valentin> docker run --name postgres-container --network PosForMary -e POSTGRES_PASSWORD=t0ps3cr3t -p 5432:5432 -d postgres
74a2c0824eccf6a9b1ce2ab0693e9ce56fc46159e5ace4f55269c7d2e5995e3
PS C:\Users\Valentin>
```

## Figura 2→ Crear contenedor de PostgreSQL

Creamos lo contenedores, en la red llamada PosForRMay, cual nos permite estar en misma red, y así poder hacer comunicar ambas contenedores

```
PS C:\Users\Valentin>
PS C:\Users\Valentin> docker run --name mariadb-container --network PosForMary -e MYSQL_ROOT_PASSWORD=t0ps3cr3t -p 3306:3306 -d mariadb
e3b1297803e90acfe38e5fa25655f2ff09c52926e8bb70e6b89df9ffe50dad4e
PS C:\Users\Valentin>
```

## Figura 3→3: Crear contenedor de MariaDB

Creamos lo contenedores, en la red llamada PosForRMay, cual nos permite estar en misma red, y así poder hacer comunicar ambas contenedores

```
PS C:\Users\Valentin> docker exec -it mariadb-container bash
root@e3b1297803e9:/#
```

## Figura 4: Acceder al contenedor de MariaDB

Accedemos al bash, donde actualizaremos los datos del contenedor e instalar los datos requeridos.

```
root@e3b1297803e9:/# apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1194 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 https://archive.mariadb.org/mariadb-11.1.2/repo/ubuntu jammy InRelease [7764 B]
Get:6 https://archive.mariadb.org/mariadb-11.1.2/repo/ubuntu jammy/main/debug amd64 Packages
Get:7 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:8 https://archive.mariadb.org/mariadb-11.1.2/repo/ubuntu jammy/main amd64 Packages [15.7 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1008 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1392 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [44.0 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
```

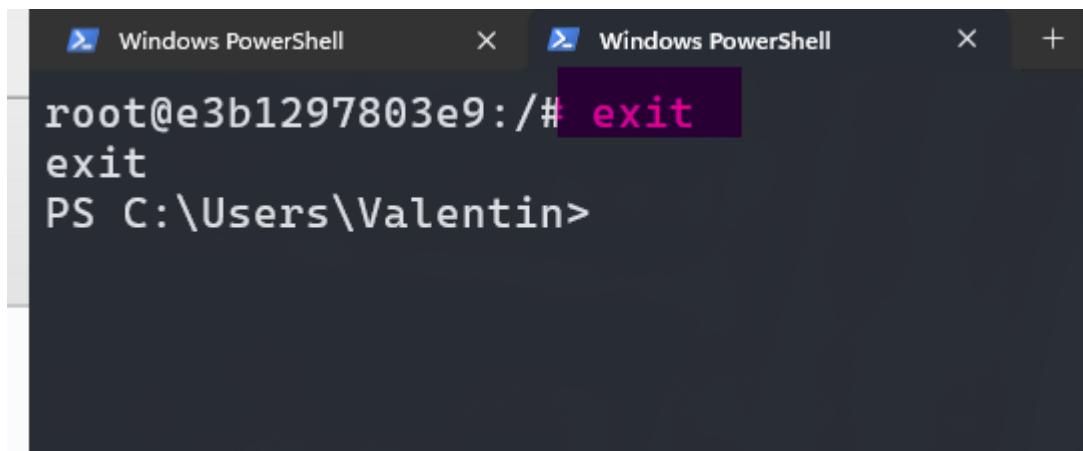
```
root@e3b1297803e9:/# apt -y install unixodbc unixodbc-dev odbcinst openssl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libltdl7 libodbc2 libodbccr2 libodbcinst2 unixodbc-common
Suggested packages:
  odbc-postgresql tdsodbc
```

```
root@e3b1297803e9:/# apt -y install mariadb-plugin-connect
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  curl libbrotli1 libcurl4 libicu70 libldap-2.5-0 libldap-common libnghhttp
  libssh-4 libxml2 publicsuffix
Suggested packages:
```

```
root@e3b1297803e9:/# apt -y install odbc-postgresql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libpq5
The following NEW packages will be installed:
```

## Figura 5: Instalar herramientas en el contenedor de MariaDB

Aquí actualizamos los repositorios e instalamos los driver para conectarnos a postgres o viceversa. con esto podremos hacer la conexión de manera segura.



```
Windows PowerShell          Windows PowerShell
root@e3b1297803e9:/# exit
exit
PS C:\Users\Valentin>
```

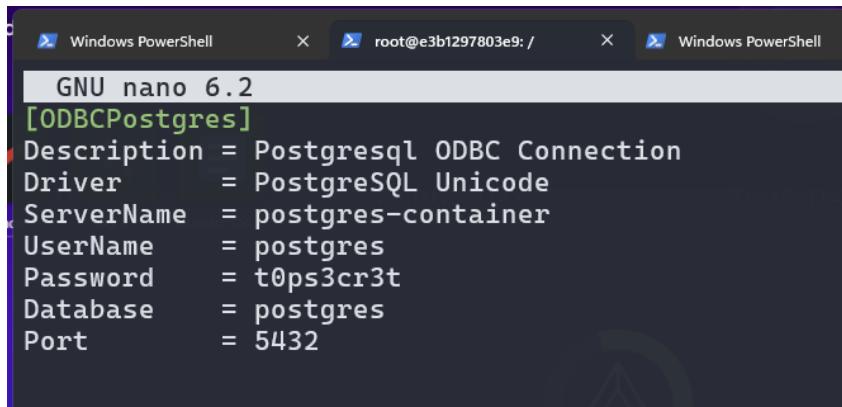
**Figura 6: Salir del contenedor de MariaDB**

Una vez terminado, nos salimos del contenedor.

```
PS C:\Users\Valentin> docker exec -it postgres-container bash
root@74a2c0824ecc:/#
```

**Figura 7: Acceder al contenedor de PostgreSQL**

Entramos al contenedor de Postgres



```
GNU nano 6.2
[ODBCPostgres]
Description = Postgresql ODBC Connection
Driver      = PostgreSQL Unicode
ServerName  = postgres-container
UserName    = postgres
Password    = t0ps3cr3t
Database   = postgres
Port        = 5432
```

**Figura 8: Edición de odbc.ini**

Editar el archivo /etc/odbc.ini en el contenedor de MariaDB, y lo guardamos para que sea segura la conexión, y se estable los datos de acceso a postgres

```
root@e3b1297803e9:/# exit
exit
PS C:\Users\Valentin> docker restart mariadb-container
mariadb-container
PS C:\Users\Valentin>
```

## Figura 9: Reiniciar el contenedor de MariaDB

Reiniciamos el contenedor para que surtan efectos y haga los cambios

```
PS C:\Users\Valentin> docker exec -it mariadb-container bash
root@e3b1297803e9:/# isql ODBCPostgres -v
+-----+
| Connected!
|
| sql-statement
| help [tablename]
| quit
|
+-----+
SQL>
```

## Figura 9:Probar conexión

Probar la conexión ODBC desde el contenedor de MariaDB, y nos devuelve que si está corriendo la conexión y listo para usarse.

```
root@e3b1297803e9:/# mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 11.1.2-MariaDB-1:11.1.2+maria~ubu2204 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use bdX;
Database changed
MariaDB [bdX]> CREATE TABLE student
->     engine=CONNECT
->     table_type=ODBC
->     tablename='student'
->     connection='DSN=ODBCPostgres;UserName=postgres;Password=t0ps3cr3t;'
->     srcdef='SELECT id, fullname, career, genre, grade FROM student';
Query OK, 0 rows affected (0.039 sec)

MariaDB [bdX]> SELECT * FROM student;
+----+-----+-----+-----+-----+
| id | fullname        | career          | genre | grade |
+----+-----+-----+-----+-----+
| 1  | Nombrel Apellido1 | Informática    | M     | 8      |
| 2  | Nombre2 Apellido2 | Ingeniería     | F     | 7      |
| 3  | Nombre3 Apellido3 | Matemáticas   | M     | 9      |
| 4  | HugoLino Valentin | Informática    | M     | 8      |
| 5  | Esmeralda Suarez | Ingeniería     | F     | 7      |
| 6  | Andres Cuevas    | Matemáticas   | M     | 9      |
+----+-----+-----+-----+-----+
6 rows in set (0.012 sec)

MariaDB [bdX]>
```

## Figura 10 consulta a la base de datos

Al realizar la consulta, esto nos devuelve los datos, y con esto hemos creado una conexión de manera efectiva con la otra base de datos.

# MariaDB desde Oracle

```
MariaDB [(none)]> CREATE DATABASE bdmysql CHARACTER SET UTF8 COLLATE utf8_spanish_ci;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]> USE bdmysql;
Database changed
MariaDB [bdmysql]>
```

## Figura 1→ creación de la base de datos

Creamos una base de datos en mariadb, con el nombre de bdmysql, posteriormente entramos a la base de datos.

```
MariaDB [bdmysql]> GRANT ALL PRIVILEGES ON bdmysql.* to usrMysql@'%' IDENTIFIED BY 'topsecret';
Query OK, 0 rows affected (0.008 sec)

MariaDB [bdmysql]> GRANT ALL PRIVILEGES ON bdmysql.* to usrMysql@'localhost' IDENTIFIED BY 'topsecret';
Query OK, 0 rows affected (0.006 sec)

MariaDB [bdmysql]>
```

## Figura 2→ Damos privilegios

otorgan todos los privilegios en la base de datos bdmysql al usuario usrMysql. La primera declaración otorga estos privilegios al usuario desde cualquier host (%), mientras que la segunda declaración solo los otorga al usuario desde el host local (localhost).

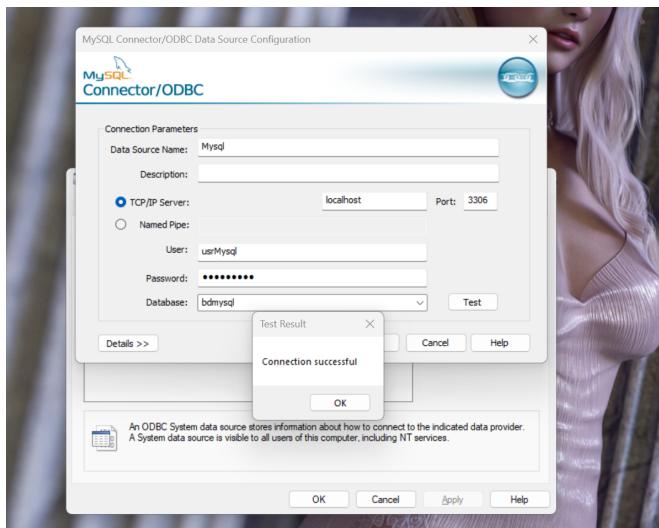
```
MariaDB [bdmysql]> CREATE TABLE prueba
-> ( prueba_id INT AUTO_INCREMENT
-> , dato VARCHAR (100)
-> , CONSTRAINT pkprueba PRIMARY KEY
-> (prueba_id)
-> );
Query OK, 0 rows affected, 1 warning (0.041 sec)

MariaDB [bdmysql]> INSERT INTO prueba (dato)
-> VALUES
-> ('Primera fila de la tabla prueba'),
-> ('Esta tabla es creada desde mysql'),
-> ('Los datos serán accesibles desde Oracle'),
-> ('Practica de topicos de BD del ITV0'),
-> ('docente: Ambrosio Cardoso Jimenez');
Query OK, 5 rows affected (0.008 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [bdmysql]>
```

### Figura 3→ creación de tabla e inserción de datos

Se ha creado una tabla llamada "prueba" en una base de datos MariaDB, con dos columnas principales: "prueba\_id" y "dato". La columna "prueba\_id" está configurada como una clave primaria con autoincremento, lo que significa que se generará automáticamente un valor único cada vez que se inserta una nueva fila. La columna "dato" es una cadena de caracteres con una longitud máxima de 100 caracteres. Se han insertado cinco filas de datos en esta tabla, proporcionando valores únicamente para la columna "dato", ya que la columna "prueba\_id" se genera automáticamente.



## Figura 4→ Conectar a la base de datos

Nos conectamos a la base de datos, desde el conector, y al probar la conexión, nos da successful, por lo tanto ya está conectado.

```
PS C:\Users\Valentin> tnsping LNKORCLMYSQL
TNS Ping Utility for 64-bit Windows: Version 19.0.0.0.0 - Production on 14-NOV-2023 23:33:06
Copyright (c) 1997, 2019, Oracle. All rights reserved.

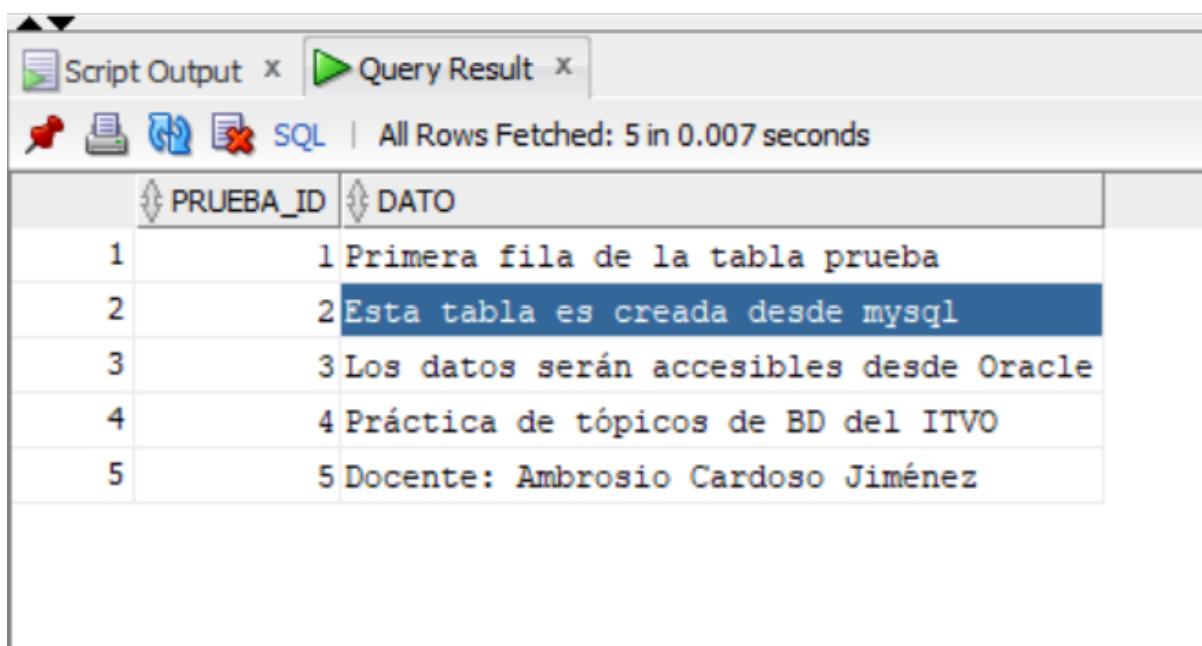
Used parameter files:
C:\Users\Valentin\Downloads\WINDOWS.X64_193000_db_home\network\admin\sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA = (SID =
oraclemysq)) (HS=OK))
OK (0 msec)
PS C:\Users\Valentin>
```

## Figura 5→Prueba de conexión

Probar desde la consola de Windows tnsping LNKORCLMYSQL

Y luego nos debe de resolver lo siguiente, y nos debe de dar todo bien



The screenshot shows a SQL developer interface with two tabs: "Script Output" and "Query Result". The "Query Result" tab is active, displaying the following table:

PRUEBA_ID	DATO
1	1 Primera fila de la tabla prueba
2	2 Esta tabla es creada desde mysql
3	3 Los datos serán accesibles desde Oracle
4	4 Práctica de tópicos de BD del ITVO
5	5 Docente: Ambrosio Cardoso Jiménez

**Figura 6→ Consulta la tabla de mariadb.**

Como podemos ver, hemos hecho una consulta desde la base de datos de oracle, y no hemos conectado a mariadb, esta tabla se obtiene desde la tabla 3.