

ACTIVIDAD DOCKER COMPOSE

DOCENTE: WILLIAM ALEXANDER MATALLANA PORRAS

ESTUDIANTE: LADY JOHANA CASTAÑEDA ORJUELA

UNIVERSIDAD DE CUNDINAMARCA EXTENSIÓN CHÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

FACULTAD DE INGENIERÍA

2025

Objetivos

- Crear y ejecutar un contenedor de MySQL con Docker Compose.
- Conectar el contenedor de MySQL y cargar datos desde un archivo SQL.
- Ejecutar consultas SQL sobre la base de datos.

ACTIVIDAD

1. ¿Qué es un archivo YAML y para qué se utiliza en desarrollo de software?

Un archivo YAML (YAML Ain't Markup Language) es un formato de serialización de datos diseñado para ser fácil de leer y escribir, con una sintaxis sencilla basada en la indentación y sin el uso de llaves ni etiquetas como en JSON o XML. Se usa ampliamente en el desarrollo de software para la configuración de aplicaciones, definición de infraestructura como código (IaC) y almacenamiento de datos estructurados.

En desarrollo de software, YAML es especialmente útil en herramientas como Docker Compose, donde define la configuración de contenedores; Kubernetes, para la gestión de despliegues y servicios; y Ansible, en la automatización de infraestructura. Su formato intuitivo facilita la edición manual y su integración en entornos DevOps, mejorando la gestión y automatización de aplicaciones y servicios.

2. Diferencia entre YAML y JSON. ¿Cuándo se recomienda usar uno sobre el otro?

Diferencias

- Sintaxis
YAML: Más legible para humanos, usa indentación y espacios en blanco.
JSON: Más estructurado, basado en llaves, corchetes y comas.
- Comentarios
YAML: Soporta comentarios
JSON: No permite comentarios nativos
- Uso de tipos de datos
YAML: Soporta múltiples tipos (strings, listas, diccionarios, números, booleanos).
JSON: Similar a YAML, pero los booleanos deben escribirse en minúsculas (true/false).
- Uso de múltiples líneas
YAML: Soporta bloques de texto en múltiples líneas con `
JSON: >` para mayor claridad.
- Compatibilidad
YAML: Requiere librerías adicionales en algunos lenguajes.
JSON: Nativo en JavaScript y ampliamente soportado en otros lenguajes.

- Facilidad de Escritura

YAML: Más fácil de leer y escribir manualmente.

JSON: Más estructurado y rígido, pero adecuado para ser procesado por máquinas.

Se recomienda usar YAML cuando se necesiten archivos de configuración fáciles de leer y editar manualmente, como en Docker Compose, Kubernetes o Ansible, también es ideal en entornos de DevOps e infraestructura como código, además de permitir comentarios dentro del archivo. Por otro lado, JSON es más adecuado para el intercambio de datos entre sistemas y APIs debido a su estructura ligera y fácil de procesar por máquinas, es ampliamente utilizado en aplicaciones web, especialmente en JavaScript, y es compatible con múltiples lenguajes sin necesidad de bibliotecas adicionales.

En general, YAML es preferido para configuraciones por su legibilidad, mientras que JSON es más eficiente para la transmisión de datos estructurados.

3. Como se crean los archivos yaml.

Para crear un archivo YAML (.yaml o .yml), se puede utilizar cualquier editor de texto como Notepad++, Visual Studio Code, Sublime Text o Nano, se abre el editor y se escribe la configuración con la sintaxis adecuada y se guarda el archivo con la extensión .yaml o .yml. YAML usa indentación con espacios en lugar de tabulaciones para estructurar los datos correctamente.

También se puede crear un archivo YAML desde la terminal en Linux o Mac usando comandos como `nano config.yaml` para abrir un editor de texto en la terminal o `touch config.yaml` para generar un archivo vacío. En Windows se puede usar el Bloc de notas y asegurarse de guardarlo con la extensión correcta escribiendo el nombre entre comillas.

Para definir el contenido del archivo (estándar) se deben responder estas preguntas:

¿Qué servicios necesita? (Ejemplo: MySQL, PHPMyAdmin, una app, etc.)

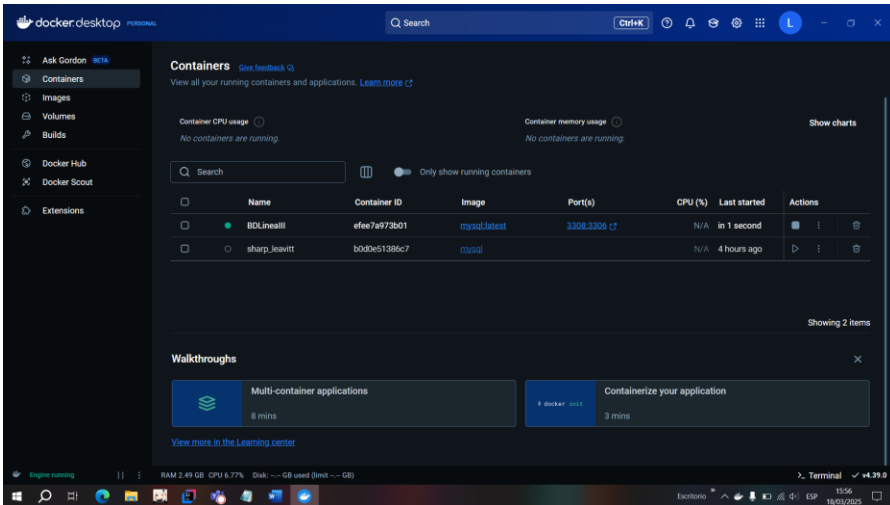
¿Qué imagen usar? (Ejemplo: `mysql:8`, `postgres:latest`, etc.)

¿Cómo se configura cada servicio? (Variables de entorno, puertos, volúmenes...)

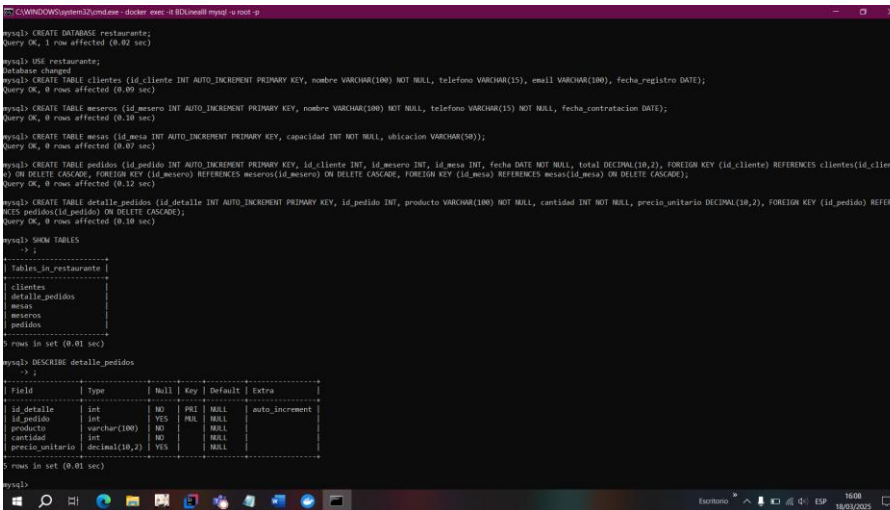
¿Cómo persistes los datos? (Para que no se pierdan al reiniciar Docker).

¿Necesita inicializar algo? (Ejecutar un `.sql` automáticamente).

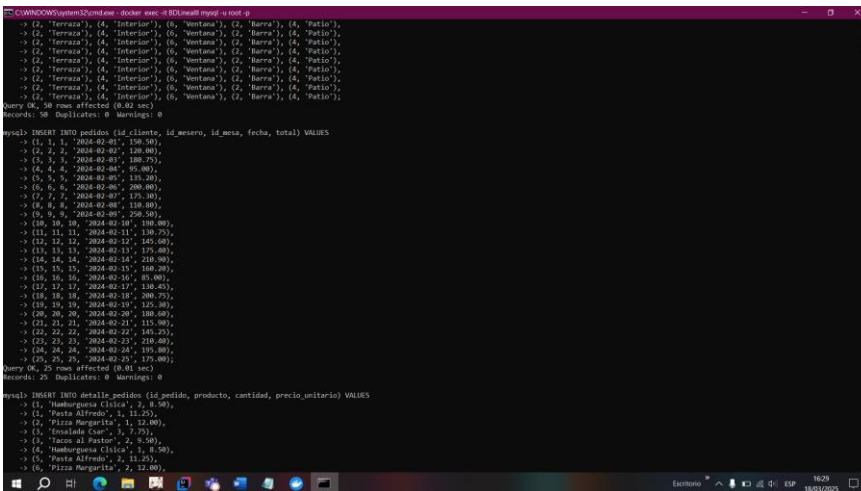
4. Crear una BD que tenga 5 tablas y desplegar el servicio usando archivos yaml.



Contenedor iniciado.



Creación de la base de datos y las tablas.



Registros insertados en todas las tablas.

CONSULTAS

1. Mostrar los pedidos con total menor a 150.

SELECT * FROM pedidos WHERE total < 150;

```
mysql> SELECT * FROM pedidos WHERE total < 50;
Empty set (0.00 sec)

mysql> SELECT * FROM pedidos WHERE total < 150;
+-----+-----+-----+-----+-----+-----+
| id_pedido | id_cliente | id_mesero | id_mesa | fecha       | total |
+-----+-----+-----+-----+-----+-----+
| 2         | 2          | 2         | 2        | 2024-02-02 | 120.00 |
| 4         | 4          | 4         | 4        | 2024-02-04 | 95.00  |
| 5         | 5          | 5         | 5        | 2024-02-05 | 135.20 |
| 8         | 8          | 8         | 8        | 2024-02-08 | 110.80 |
| 11        | 11         | 11        | 11       | 2024-02-11 | 130.75 |
| 12        | 12         | 12        | 12       | 2024-02-12 | 145.60 |
| 16        | 16         | 16        | 16       | 2024-02-16 | 85.00  |
| 17        | 17         | 17        | 17       | 2024-02-17 | 130.45 |
| 19        | 19         | 19        | 19       | 2024-02-19 | 125.30 |
| 21        | 21         | 21        | 21       | 2024-02-21 | 115.90 |
| 22        | 22         | 22        | 22       | 2024-02-22 | 145.25 |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
```

2. Aumentar en 10% el precio de los productos en los pedidos.

UPDATE detalle_pedidos SET precio_unitario = precio_unitario * 1.10;

SELECT * FROM detalle_pedidos;

```
mysql> UPDATE detalle_pedidos SET precio_unitario = precio_unitario * 1.10;
Query OK, 20 rows affected, 20 warnings (0.00 sec)
Rows matched: 20  Changed: 20  Warnings: 0

mysql> SELECT * FROM detalle_pedidos;
+-----+-----+-----+-----+-----+
| id_pedido | id_cliente | id_producto | precio_unitario |
+-----+-----+-----+-----+
| 1         | 1          | Pizza Margherita | 10.00 |
| 1         | 1          | Tacos al Pastor | 10.00 |
| 2         | 2          | Ensalada Cesar | 10.00 |
| 2         | 2          | Pasta Alfredo | 10.00 |
| 2         | 2          | Hamburguesa Clasic | 10.00 |
| 3         | 3          | Pizza Margherita | 10.00 |
| 3         | 3          | Tacos al Pastor | 10.00 |
| 3         | 3          | Ensalada Cesar | 10.00 |
| 3         | 3          | Pasta Alfredo | 10.00 |
| 3         | 3          | Hamburguesa Clasic | 10.00 |
| 4         | 4          | Pizza Margherita | 10.00 |
| 4         | 4          | Tacos al Pastor | 10.00 |
| 4         | 4          | Ensalada Cesar | 10.00 |
| 4         | 4          | Pasta Alfredo | 10.00 |
| 4         | 4          | Hamburguesa Clasic | 10.00 |
| 5         | 5          | Pizza Margherita | 10.00 |
| 5         | 5          | Tacos al Pastor | 10.00 |
| 5         | 5          | Ensalada Cesar | 10.00 |
| 5         | 5          | Pasta Alfredo | 10.00 |
| 5         | 5          | Hamburguesa Clasic | 10.00 |
| 6         | 6          | Pizza Margherita | 10.00 |
| 6         | 6          | Tacos al Pastor | 10.00 |
| 6         | 6          | Ensalada Cesar | 10.00 |
| 6         | 6          | Pasta Alfredo | 10.00 |
| 6         | 6          | Hamburguesa Clasic | 10.00 |
| 7         | 7          | Pizza Margherita | 10.00 |
| 7         | 7          | Tacos al Pastor | 10.00 |
| 7         | 7          | Ensalada Cesar | 10.00 |
| 7         | 7          | Pasta Alfredo | 10.00 |
| 7         | 7          | Hamburguesa Clasic | 10.00 |
| 8         | 8          | Pizza Margherita | 10.00 |
| 8         | 8          | Tacos al Pastor | 10.00 |
| 8         | 8          | Ensalada Cesar | 10.00 |
| 8         | 8          | Pasta Alfredo | 10.00 |
| 8         | 8          | Hamburguesa Clasic | 10.00 |
| 9         | 9          | Pizza Margherita | 10.00 |
| 9         | 9          | Tacos al Pastor | 10.00 |
| 9         | 9          | Ensalada Cesar | 10.00 |
| 9         | 9          | Pasta Alfredo | 10.00 |
| 9         | 9          | Hamburguesa Clasic | 10.00 |
| 10        | 10         | Pizza Margherita | 10.00 |
| 10        | 10         | Tacos al Pastor | 10.00 |
| 10        | 10         | Ensalada Cesar | 10.00 |
| 10        | 10         | Pasta Alfredo | 10.00 |
| 10        | 10         | Hamburguesa Clasic | 10.00 |
| 11        | 11         | Pizza Margherita | 10.00 |
| 11        | 11         | Tacos al Pastor | 10.00 |
| 11        | 11         | Ensalada Cesar | 10.00 |
| 11        | 11         | Pasta Alfredo | 10.00 |
| 11        | 11         | Hamburguesa Clasic | 10.00 |
| 12        | 12         | Pizza Margherita | 10.00 |
| 12        | 12         | Tacos al Pastor | 10.00 |
| 12        | 12         | Ensalada Cesar | 10.00 |
| 12        | 12         | Pasta Alfredo | 10.00 |
| 12        | 12         | Hamburguesa Clasic | 10.00 |
| 13        | 13         | Pizza Margherita | 10.00 |
| 13        | 13         | Tacos al Pastor | 10.00 |
| 13        | 13         | Ensalada Cesar | 10.00 |
| 13        | 13         | Pasta Alfredo | 10.00 |
| 13        | 13         | Hamburguesa Clasic | 10.00 |
| 14        | 14         | Pizza Margherita | 10.00 |
| 14        | 14         | Tacos al Pastor | 10.00 |
| 14        | 14         | Ensalada Cesar | 10.00 |
| 14        | 14         | Pasta Alfredo | 10.00 |
| 14        | 14         | Hamburguesa Clasic | 10.00 |
| 15        | 15         | Pizza Margherita | 10.00 |
| 15        | 15         | Tacos al Pastor | 10.00 |
| 15        | 15         | Ensalada Cesar | 10.00 |
| 15        | 15         | Pasta Alfredo | 10.00 |
| 15        | 15         | Hamburguesa Clasic | 10.00 |
| 16        | 16         | Pizza Margherita | 10.00 |
| 16        | 16         | Tacos al Pastor | 10.00 |
| 16        | 16         | Ensalada Cesar | 10.00 |
| 16        | 16         | Pasta Alfredo | 10.00 |
| 16        | 16         | Hamburguesa Clasic | 10.00 |
| 17        | 17         | Pizza Margherita | 10.00 |
| 17        | 17         | Tacos al Pastor | 10.00 |
| 17        | 17         | Ensalada Cesar | 10.00 |
| 17        | 17         | Pasta Alfredo | 10.00 |
| 17        | 17         | Hamburguesa Clasic | 10.00 |
| 18        | 18         | Pizza Margherita | 10.00 |
| 18        | 18         | Tacos al Pastor | 10.00 |
| 18        | 18         | Ensalada Cesar | 10.00 |
| 18        | 18         | Pasta Alfredo | 10.00 |
| 18        | 18         | Hamburguesa Clasic | 10.00 |
| 19        | 19         | Pizza Margherita | 10.00 |
| 19        | 19         | Tacos al Pastor | 10.00 |
| 19        | 19         | Ensalada Cesar | 10.00 |
| 19        | 19         | Pasta Alfredo | 10.00 |
| 19        | 19         | Hamburguesa Clasic | 10.00 |
| 20        | 20         | Pizza Margherita | 10.00 |
| 20        | 20         | Tacos al Pastor | 10.00 |
| 20        | 20         | Ensalada Cesar | 10.00 |
| 20        | 20         | Pasta Alfredo | 10.00 |
| 20        | 20         | Hamburguesa Clasic | 10.00 |
```

3. Contar cuántos pedidos hay por cada mes.

```
SELECT MONTH(fecha) AS mes, COUNT(id_pedido) AS total_pedidos FROM
pedidos GROUP BY mes ORDER BY mes;
```

```
mysql> SELECT MONTH(fecha) AS mes, COUNT(id_pedido) AS total_pedidos FROM pedidos GROUP BY mes ORDER BY mes;
+-----+-----+
| mes | total_pedidos |
+-----+-----+
| 2 | 25 |
+-----+-----+
1 row in set (0.01 sec)

mysql>
```

4. Buscar los clientes cuyo nombre empiece con 'A'.

```
SELECT * FROM clientes WHERE nombre LIKE 'A%';
```

```
mysql> SELECT * FROM clientes WHERE nombre LIKE 'A%';
+-----+-----+-----+-----+-----+
| id_cliente | nombre | telefono | email | fecha_registro |
+-----+-----+-----+-----+-----+
| 4 | Ana Gonzalez | 345678912 | ana@example.com | 2024-01-04 |
| 19 | Andrs Ortega | 789654123 | andres@example.com | 2024-01-19 |
| 31 | Alejandro Silva | 951357852 | alejandro@example.com | 2024-01-31 |
| 46 | Andrea Hidalgo | 357159486 | andrea@example.com | 2024-02-15 |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql>
```

5. Mostrar el pedido con el total más alto.

```
SELECT * FROM pedidos ORDER BY total DESC LIMIT 1;
```

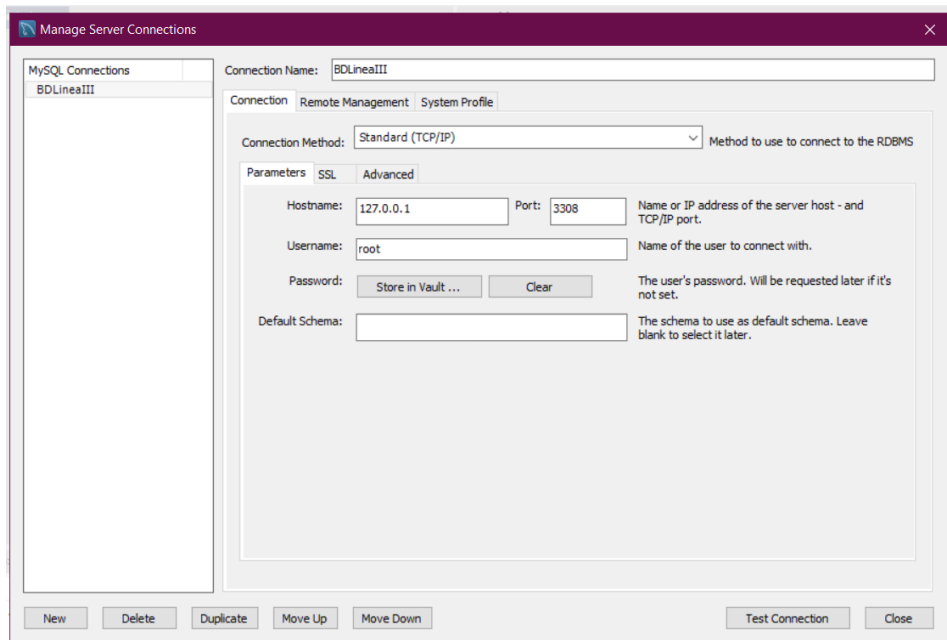
```
mysql> SELECT * FROM pedidos ORDER BY total DESC LIMIT 1;
+-----+-----+-----+-----+-----+-----+
| id_pedido | id_cliente | id_mesero | id_mesa | fecha | total |
+-----+-----+-----+-----+-----+-----+
| 9 | 9 | 9 | 9 | 2024-02-09 | 250.50 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

6. Obtener el producto más vendido en los pedidos.

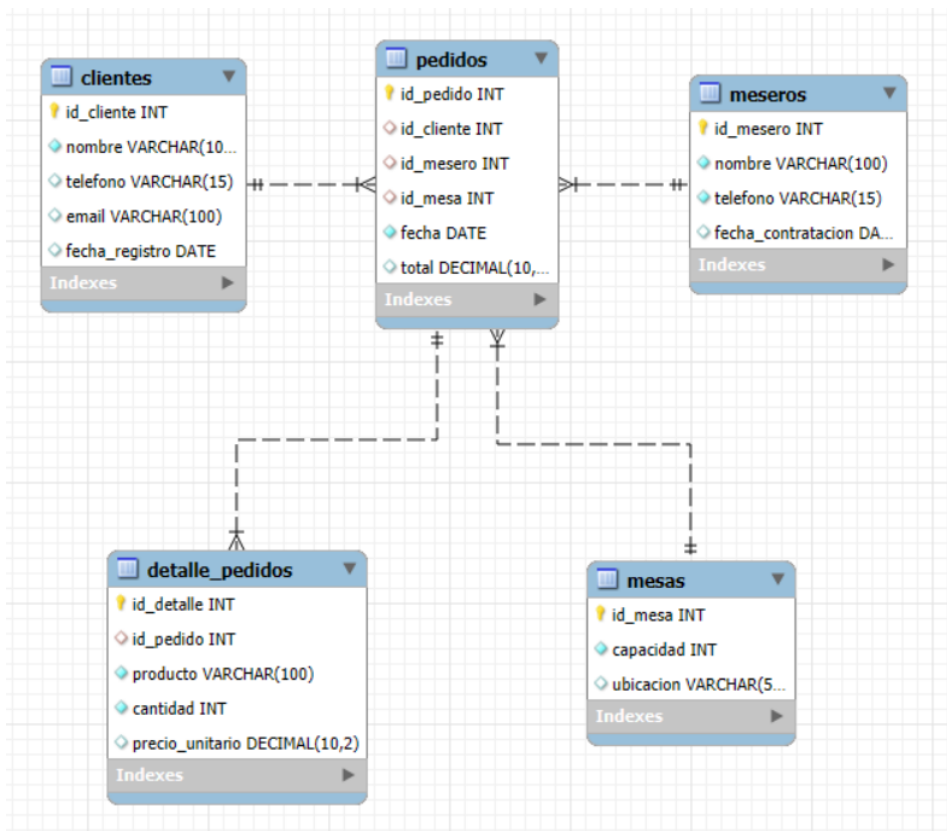
```
SELECT producto, SUM(cantidad) AS total_vendido FROM detalle_pedidos
GROUP BY producto ORDER BY total_vendido DESC LIMIT 1;
```

```
mysql> SELECT producto, SUM(cantidad) AS total_vendido FROM detalle_pedidos GROUP BY producto ORDER BY total_vendido DESC LIMIT 1;
+-----+-----+
| producto | total_vendido |
+-----+-----+
| Tacos al Pastor | 20 |
+-----+-----+
1 row in set (0.00 sec)
```

INSTANCIA



DIAGRAMA



DESPLEGAR EL SERVICIO USANDO ARCHIVOS YAML

Crear el .sql en este caso restaurante.sql

```
restaurante.sql: Bloc de notas
Archivo Edición Formato Ver Ayuda
CREATE DATABASE IF NOT EXISTS restaurante;
USE restaurante;

CREATE TABLE clientes (
  id_cliente INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  telefono VARCHAR(15),
  email VARCHAR(100),
  fecha_registro DATE
);

CREATE TABLE meseros (
  id_mesero INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  telefono VARCHAR(15) NOT NULL,
  fecha_contratacion DATE
);

CREATE TABLE mesas (
  id_mesa INT AUTO_INCREMENT PRIMARY KEY,
  capacidad INT NOT NULL,
  ubicacion VARCHAR(50)
);

CREATE TABLE pedidos (
  id_pedido INT AUTO_INCREMENT PRIMARY KEY,
  id_cliente INT NOT NULL,
  id_mesero INT NOT NULL,
  id_mesa INT NOT NULL,
  fecha DATE NOT NULL,
  total DECIMAL(10,2),
  FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente) ON DELETE CASCADE,
  FOREIGN KEY (id_mesero) REFERENCES meseros(id_mesero) ON DELETE CASCADE,
  FOREIGN KEY (id_mesa) REFERENCES mesas(id_mesa) ON DELETE CASCADE
);

CREATE TABLE detalle_pedidos (
  id_detalle INT AUTO_INCREMENT PRIMARY KEY,
```

Crear docker-compose.yml con la configuración.

```
docker-compose.yml: Bloc de notas
Archivo Edición Formato Ver Ayuda
|version: '3.8'

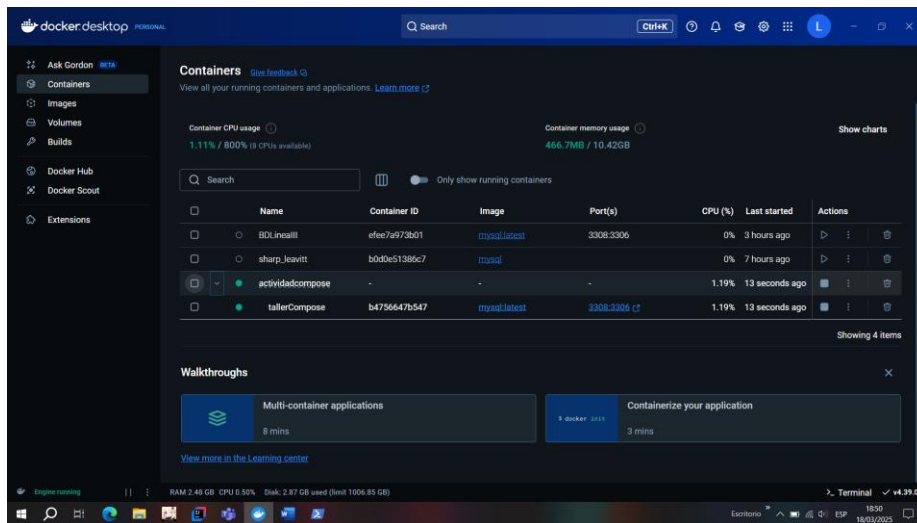
services:
  mysql_restaurante:
    image: mysql:latest
    container_name: tallerCompose
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: 1234
      MYSQL_DATABASE: restaurante
    ports:
      - "3308:3306"
    volumes:
      - mysql_data:/var/lib/mysql
      - ./restaurante.sql:/docker-entrypoint-initdb.d/restaurante.sql

volumes:
  mysql_data:
```

Iniciar los servicios en docker-compose.yml

```
PS C:\Users\lady3\Documentos\Universidad\linea_3\actividadCompose> docker-compose up -d
time="2025-03-18T18:50:03-05:00" level=warning msg="C:\Users\lady3\Documentos\Universidad\linea_3\actividadCompose\docker-compose.yml: the attribute 'v
ersion' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Building 0.0s
[+] Network actividadcompose_default Created 0.1s
[+] Container tallerCompose Started 1.4s
PS C:\Users\lady3\Documentos\Universidad\linea_3\actividadCompose>
```


Se creo correctamente el contenedor.



Ingresar a la consola mysql del contenedor y si se creó la base de datos restauranteCo.

```
PS C:\Users\lady3\Documentos\Universidad\linea_3\actividadCompose> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED      STATUS      PORTS                               NAMES
b4756647b547   mysql:latest "docker-entrypoint.s..." About a minute ago Up About a minute 33060/tcp, 0.0.0.0:3308->3306/tcp tallerCompose

PS C:\Users\lady3\Documentos\Universidad\linea_3\actividadCompose> docker exec -it tallercompose mysql -u root -p
Enter password:
welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
server version: 9.2.0 MySQL Community Server - GPL

copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> SHOW DATABASES
+----+
| Database |
+----+
| information_schema |
| mysql |
| performance_schema |
| restauranteCo |
| sys |
+----+
5 rows in set (0.02 sec)
```

Revisar si las tablas se crearon y si se cargaron los registros.

```
mysql> USE restauranteCo
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_restauranteCo |
+-----+
| clientes |
| detalle_pedidos |
| mesas |
| meseros |
| pedidos |
+-----+
5 rows in set (0.01 sec)

mysql> DESCRIBE clientes
+----+
| Field | Type | Null | Key | Default | Extra |
+----+
| id_cliente | int | NO | PRI | NULL | auto_increment |
| nombre | varchar(100) | NO | | NULL | |
| telefono | varchar(15) | YES | | NULL | |
| email | varchar(100) | YES | | NULL | |
| fecha_registro | date | YES | | NULL | |
+----+
5 rows in set (0.01 sec)

mysql> SELECT * FROM clientes WHERE nombre LIKE 'A%';
+----+
| id_cliente | nombre | telefono | email | fecha_registro |
+----+
| 4 | Ana González | 345678912 | ana@example.com | 2024-01-04 |
| 19 | Andrés Ortega | 789654123 | andres@example.com | 2024-01-19 |
| 31 | Alejandro Silva | 951357852 | alejandro@example.com | 2024-01-31 |
| 46 | Andrea Hidalgo | 357159486 | andrea@example.com | 2024-02-15 |
+----+
4 rows in set (0.01 sec)

mysql>
```

Comandos para lanzar el docker compose

docker-compose up -d

El comando docker-compose up se utiliza para iniciar los servicios definidos en un archivo docker-compose.yml. El modificador -d significa que se ejecutará los contenedores en segundo plano y no mostrará los logs de la ejecución directamente en la terminal.

Lee el archivo docker-compose.yml para ver qué contenedores y configuraciones debe usar, crea y ejecuta los contenedores en segundo plano y si el contenedor ya existe y ha sido configurado previamente, solo se reinicia.

Conclusiones

- Usar Docker Compose facilitó la creación y ejecución del contenedor sin complicaciones, ahorrando tiempo en la configuración de MySQL.
- Se pudo cargar los datos correctamente en la base de datos, lo que permitió tener todo listo para hacer las consultas necesarias.
- Las consultas funcionaron correctamente, confirmando que la base de datos estaba bien configurada y operativa dentro del contenedor.

Referencias

OpenAI. (2025). *Explicación sobre archivos YAML, JSON y sus diferencias*. ChatGPT. Recuperado el 18 de marzo de 2025, de <https://www.openai.com/chatgpt>