

Manejo de archivos ymal con docker y mysql workbench

Juan Fernando Trujillo Rivera

Universidad de Cundinamarca

2025

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 que se usa comúnmente en el desarrollo de software para almacenar configuraciones y datos estructurados de manera sencilla y legible para los humanos.

Características de YAML:

- **Legible:** Su sintaxis es clara y fácil de entender, similar a JSON pero sin llaves `{}` ni corchetes `[]`.
- **Basado en indentación:** Usa espacios en lugar de símbolos para definir la jerarquía de los datos.
- **Soporta múltiples tipos de datos:** Como cadenas, números, listas y diccionarios.
- **Compatible con JSON:** Puede ser convertido fácilmente a JSON y viceversa.

Usos en desarrollo de software:

- **Archivos de configuración:** Se utiliza en frameworks y herramientas como:
 - Kubernetes (`deployment.yaml`)
 - Docker Compose (`docker-compose.yaml`)
 - GitHub Actions (`.github/workflows/`)
 - Ansible (`playbooks.yaml`)
- **Definición de infraestructura como código (IaC):** En herramientas como AWS CloudFormation o Terraform.
- **Almacenamiento de datos estructurados:** Puede servir como una alternativa a JSON o XML en aplicaciones que manejan datos jerárquicos.
- **Configuración de CI/CD:** Para definir flujos de integración y despliegue continuo en plataformas como GitLab CI/CD o CircleCI.

Ejemplo de un archivo YAML:

```

version: "3.8"
services:
  web:
    image: nginx
    ports:
      - "80:80"
  database:
    image: postgres
    environment:
      POSTGRES_USER: admin
      POSTGRES_PASSWORD: secret

```

Este archivo se usa en **Docker Compose** para definir servicios de una aplicación

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

Característica	YAML	JSON
Legibilidad	Más fácil para humanos, usa indentación y es más limpio	Más compacto pero con más símbolos como {}, [], "
Comentarios	Permite comentarios con #	No admite comentarios
Sintaxis	Basada en espacios y sangría	Basada en estructuras de clave-valor con comillas
Compatibilidad	Se puede convertir fácilmente a JSON	Se puede convertir a YAML, pero puede ser más verboso
Soporte en lenguajes	Menos soporte nativo, necesita bibliotecas	Soporte nativo en la mayoría de los lenguajes de programación

Característica	YAML	JSON
Uso en configuraciones	Muy usado en archivos de configuración por su claridad	Usado en configuraciones, pero menos legible para humanos

¿Cuándo usar YAML o JSON?

Usar YAML cuando:

- Se necesiten archivos de configuración legibles, como en Kubernetes, Docker Compose o GitHub Actions.
- Se requiera escribir comentarios dentro del archivo.
- Se prefiera un formato más amigable y menos propenso a errores de sintaxis debido a comillas o comas.

Usar JSON cuando:

- Se necesite compatibilidad con APIs y sistemas que requieran datos estructurados (REST APIs, bases de datos NoSQL como MongoDB).
- Se busque mejor soporte en múltiples lenguajes de programación sin necesidad de bibliotecas adicionales.
- Se quiera un formato más estricto y fácil de procesar por máquinas.

3. ¿Como se crean los archivos yml?

Crear un archivo **YAML** (.yaml o .yml) es bastante sencillo. Puedes hacerlo en cualquier editor de texto o en un entorno de desarrollo como VS Code, Sublime Text o incluso el Bloc de notas.

- **Pasos para crear un archivo .yaml**
- **Método 1: En Windows**
- **Abrir un editor de texto** (Bloc de notas, VS Code, Notepad++, etc.).
- **Escribir el contenido en formato YAML**, por ejemplo:

```
nombre: "Mi Aplicación"
version: 1.0
configuracion:
  base_de_datos:
    usuario: "admin"
    contraseña: "secreta"
  puerto: 8080
```

-
- **Guardar el archivo** con la extensión .yaml o .yml:
 - En el Bloc de notas:
 - Haz clic en **Archivo > Guardar como...**
 - En **Tipo**, selecciona **Todos los archivos (*.*)**
 - Escribe el nombre del archivo con la extensión .yaml, por ejemplo: config.yaml
 - Haz clic en **Guardar**

Base de datos (consultas)

```
7 -- Tabla: productos
8 INSERT INTO productos (ID_Producto, Nombre, Categoria, Precio, Stock, ID_Proveedor) VALUES
9 (1, 'Cerveza', 'Bebida', 5000, 100, 1),
0 (2, 'Whisky', 'Bebida', 120000, 20, 2),
1 (3, 'Hamburguesa', 'Comida', 25000, 50, 1);
2
3 -- Tabla: inventario
4 INSERT INTO inventario (ID_Inventario, ID_Producto, Cantidad_Inicial, Cantidad_Actual, Fecha_Registro) VALUES
5 (1, 1, 100, 90, NOW()),
6 (2, 2, 20, 15, NOW()),
7 (3, 3, 50, 45, NOW());
8
```

```

-- Tabla: mesas
INSERT INTO mesas (ID_Mesa, Numero_Mesa, Capacidad, Estado) VALUES
(1, 1, 4, 'Disponible'),
(2, 2, 6, 'Ocupada');

-- Tabla: ventas
INSERT INTO ventas (ID_Venta, ID_Usuario, ID_Cliente, ID_Mesa, Fecha_Hora, Total, Metodo_Pago) VALUES
(1, 2, 1, 1, NOW(), 50000, 'Efectivo'),
(2, 3, 2, 2, NOW(), 75000, 'Tarjeta');

-- Tabla: detalle_venta
INSERT INTO detalle_venta (ID_Detalle, ID_Venta, ID_Producto, Cantidad, Precio_Unitario, Subtotal) VALUES
(1, 1, 1, 4, 5000, 20000),
(2, 1, 3, 1, 25000, 25000),
(3, 2, 2, 1, 120000, 50000);

-- Tabla: incidencias_fraude
INSERT INTO incidencias_fraude (ID_Incidencia, ID_Usuario, ID_Venta, Tipo_Incidencia, Descripcion, Fecha_Hora, Estado) VALUES
(1, 2, 1, 'Robo', 'Dinero faltante en caja', NOW(), 'Pendiente');

-- Tabla: reservaciones
INSERT INTO reservaciones (ID_Reservacion, ID_Cliente, ID_Mesa, Fecha_Hora, Estado) VALUES
(1, 1, 1, NOW(), 'Confirmada');

-- Tabla: cierres_caja
INSERT INTO cierres_caja (ID_Cierre, ID_Usuario, Fecha_Hora, Monto_Declarado, Monto_Real, Diferencia) VALUES
(1, 2, NOW(), 500000, 495000, 5000);

-- DESACTIVAR restricciones de clave foránea para evitar errores al eliminar
SET FOREIGN_KEY_CHECKS = 0;

```

```

60 -- ELIMINAR REGISTROS RELACIONADOS CON LAS VENTAS
61 DELETE FROM detalle_venta WHERE ID_Venta = 1;
62 DELETE FROM incidencias_fraude WHERE ID_Venta = 1;
63 DELETE FROM ventas WHERE ID_Venta = 1;
64
65 -- REACTIVAR restricciones de clave foránea
66 SET FOREIGN_KEY_CHECKS = 1;
67
68 -- INNER JOIN: Mostrar ventas con los datos del cliente y usuario que la realizó
69 SELECT v.ID_Venta, c.Nombre AS Cliente, u.Nombre AS Cajero, v.Total, v.Metodo_Pago
70 FROM ventas v
71 INNER JOIN clientes c ON v.ID_Cliente = c.ID_Cliente
72 INNER JOIN usuarios u ON v.ID_Usuario = u.ID_Usuario;

```

```

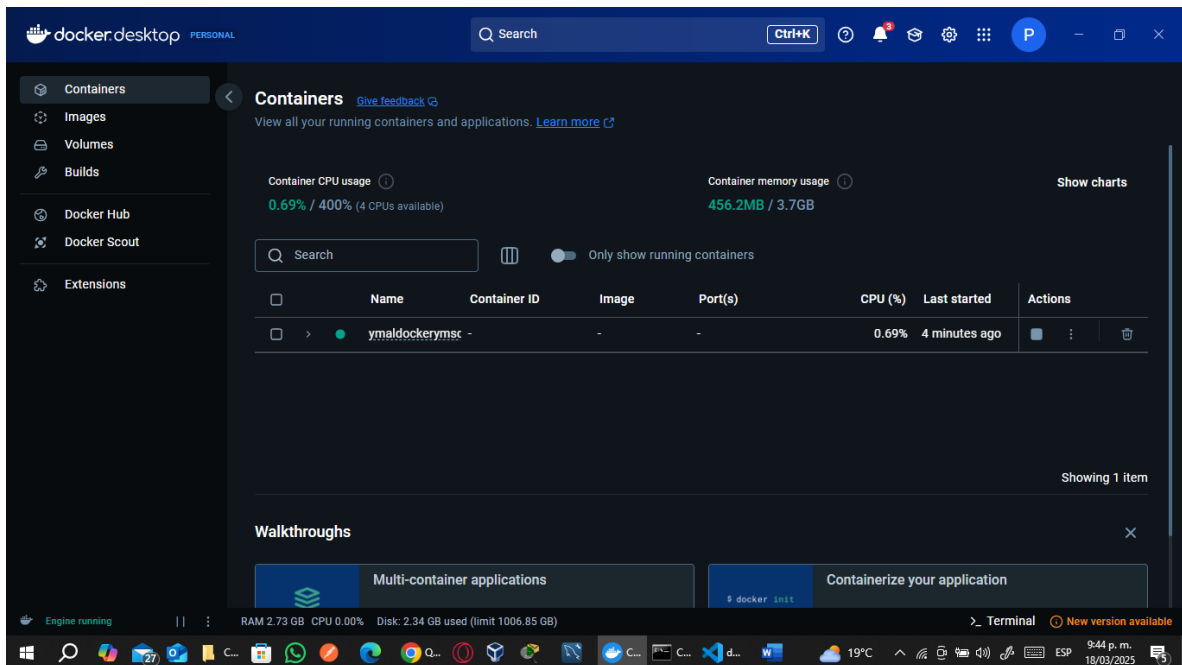
4 -- LEFT JOIN: Mostrar todos los productos y las ventas en las que han sido vendidos (incluye productos sin ventas)
5 SELECT p.Nombre AS Producto, dv.ID_Venta, dv.Cantidad
6 FROM productos p
7 LEFT JOIN detalle_venta dv ON p.ID_Producto = dv.ID_Producto;
8
9 -- RIGHT JOIN: Mostrar todas las ventas y los productos vendidos (incluye ventas sin productos vendidos)
10 SELECT v.ID_Venta, p.Nombre AS Producto, dv.Cantidad
11 FROM productos p
12 RIGHT JOIN detalle_venta dv ON p.ID_Producto = dv.ID_Producto
13 RIGHT JOIN ventas v ON dv.ID_Venta = v.ID_Venta;
14

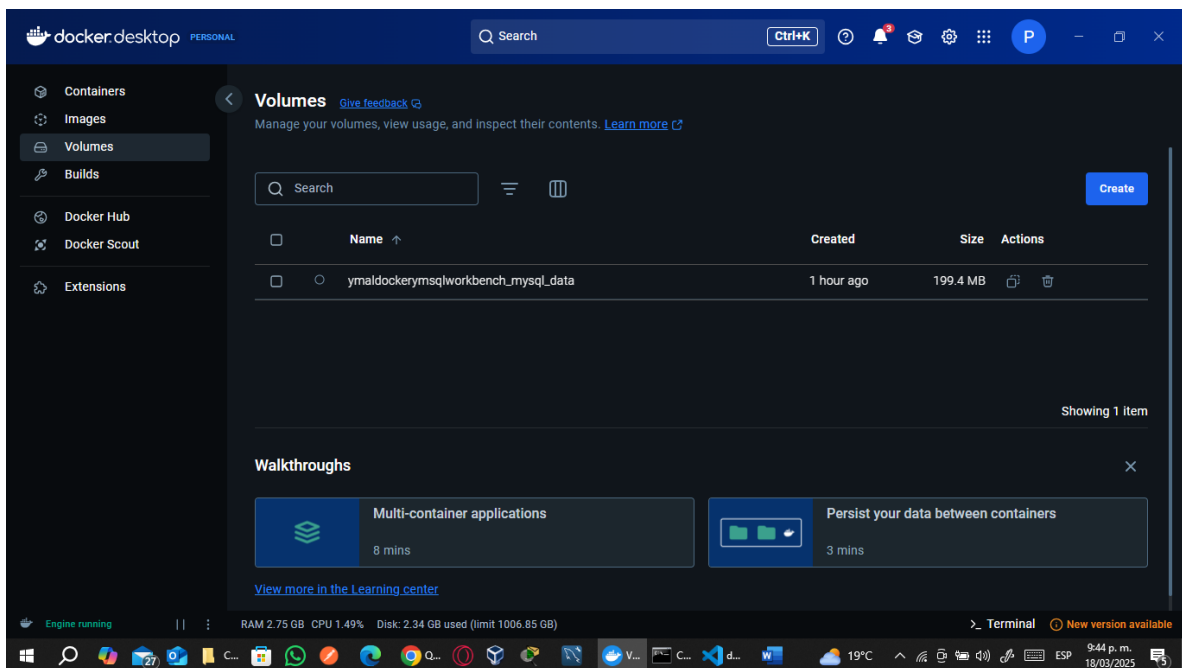
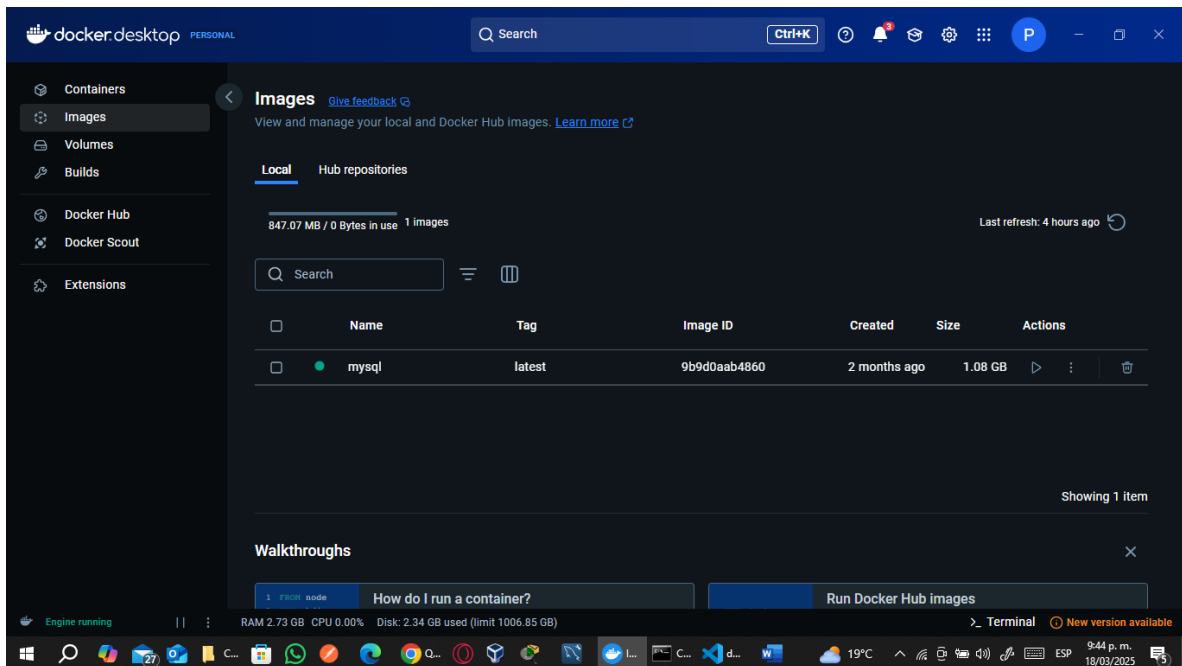
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Lenovo\Desktop\ymal,docker y mysqlworkbench>
C:\Users\Lenovo\Desktop\ymal,docker y mysqlworkbench>
C:\Users\Lenovo\Desktop\ymal,docker y mysqlworkbench>
C:\Users\Lenovo\Desktop\ymal,docker y mysqlworkbench>
C:\Users\Lenovo\Desktop\ymal,docker y mysqlworkbench>
C:\Users\Lenovo\Desktop\ymal,docker y mysqlworkbench>docker-compose up -d
time="2025-03-18T21:37:08-05:00" level=warning msg="C:\\Users\\Lenovo\\Desktop\\ymal,docker y mysqlworkbench\\docker-comp
ose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 1/3
  - mysql [66] 135.3MB / 135.7MB Pulling 129.8s
    69e76254f502 Already exists 0.0s
  - 27a2553d6a80 Downloading [=====> ] 135... 126.5s
canceled

C:\Users\Lenovo\Desktop\ymal,docker y mysqlworkbench>docker-compose up -d
time="2025-03-18T21:39:36-05:00" level=warning msg="C:\\Users\\Lenovo\\Desktop\\ymal,docker y mysqlworkbench\\docker-comp
ose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 3/3
  - mysql Pulled 32.4s
    69e76254f502 Already exists 0.1s
    27a2553d6a80 Download complete 8.5s
[+] Running 2/2
  - Network ymaldockerymysqlworkbench_default Created 0.4s
  - Container discoteca_mysql_container Started 16.4s

C:\Users\Lenovo\Desktop\ymal,docker y mysqlworkbench>
```





¡Crear, levantar y bajar Docker!

Levantar

`docker-compose up -d`

Bajar

docker-compose down

Construir

docker-compose up --build -d