

EJERCICO YML

Valeria Medina Ocampo

ID:834515

Ingeniería de sistemas, Facultad de ingeniería, Corporación Universitaria Minuto de Dios

NRC 60-66167: Arquitectura de software

Prof. William Alexander Matallana Porras

Marzo 23, 2025

1. Archivos yaml vs json

YAML y JSON son formatos populares para representar datos estructurados, Facilitan un método de intercambio de datos comprensible para las personas. Estructuran la información de forma estandarizada para permitir su transferencia entre diferentes aplicaciones de software, manteniendo un formato legible por humanos. Además, ambos organizan los datos en pares de clave y valor. Mas sin embargo tienen varias diferencias.

Archivo YML: abreviatura de YAML Ain't Markup Language,
Archivo JSON: abreviatura de JavaScript Object Notation

YAML es más legible gracias a su uso de indentación, mientras que JSON utiliza llaves y corchetes, lo que puede ser menos claro.

YAML admite comentarios en cambio JSON no

YAML soporta estructuras de datos más complejas, pero es más complicado de procesar.

JSON es ampliamente utilizado en APIs web por su compatibilidad con JavaScript, mientras que YAML es preferido para archivos de configuración.

2. Docker compose.yml -uso

Docker Compose es una herramienta diseñada para definir y ejecutar aplicaciones que utilizan múltiples contenedores. Desempeña un papel fundamental en la optimización y eficiencia del desarrollo e implementación.

Gracias a Compose, la gestión de toda la pila de aplicaciones se vuelve más sencilla, permitiendo administrar servicios, redes y volúmenes desde un único archivo de configuración en formato YAML. Con un solo comando, es posible crear e iniciar todos los servicios definidos en dicho archivo.

Para trabajar con Docker Compose, se utilizan los siguientes comandos:

`docker-compose build`: Construye las imágenes definidas en el archivo `docker-compose.yml`.

`docker-compose up`: Inicia los contenedores definidos en el archivo. Puedes usar `-d` para ejecutar en segundo plano.

`docker-compose stop`: Detiene los contenedores en ejecución.

`docker-compose down`: Detiene y elimina los contenedores y redes creadas.

`docker-compose logs`: Muestra los logs de los contenedores.

3. Como se crea un contenido usando yml

La estructura básica es:

Indentación: Utiliza espacios para indentar y definir la jerarquía de los datos. No se deben usar tabuladores.

Pares clave-valor: Se representan con un formato clave: valor, siempre con un espacio después de los dos puntos.

Listas: Se definen con guiones (-) al principio de cada elemento.

Comentarios: Se agregan con el símbolo #.

Para crea contenido es:

Definir la estructura: Identifica las secciones y los datos que necesitas incluir.

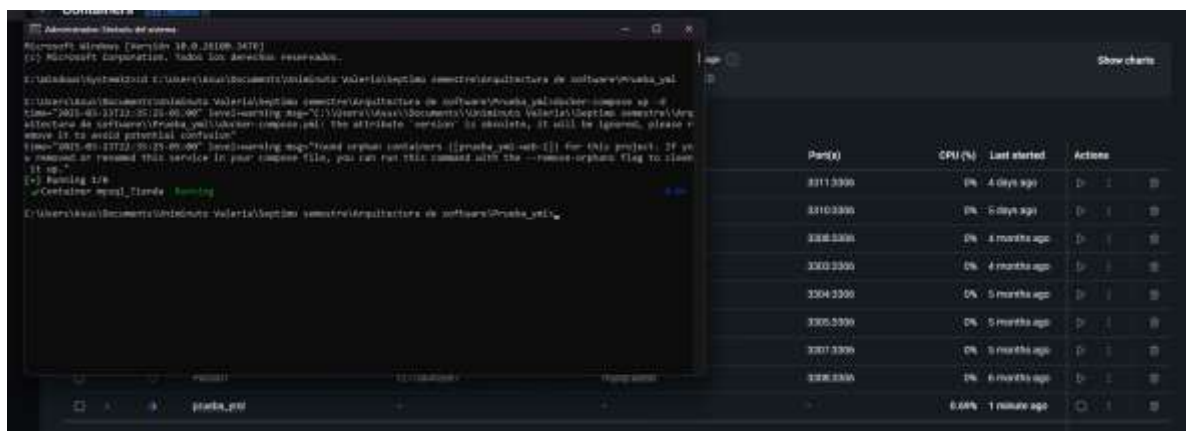
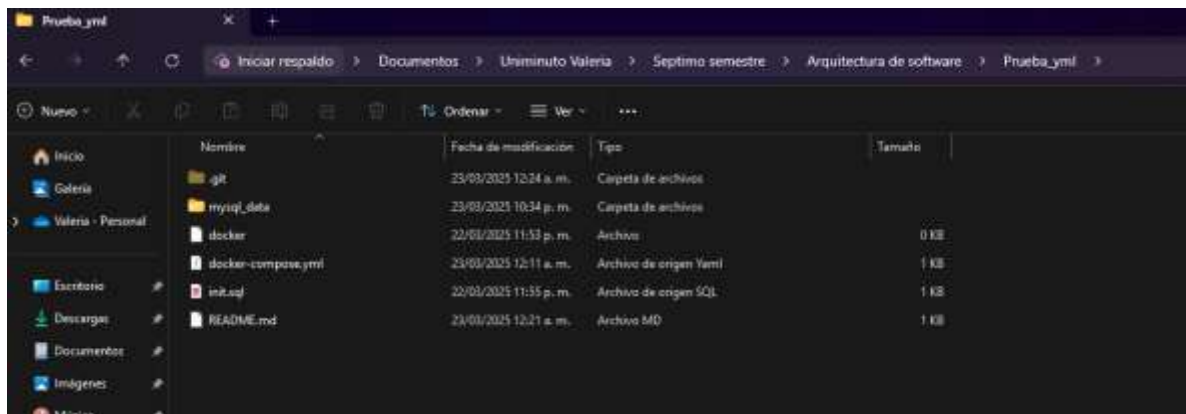
Usar un editor adecuado: Herramientas como Notepad++ o Visual Studio Code son ideales para editar archivos YAML debido a su soporte para espacios y resaltado de sintaxis.

Crear pares clave-valor: Utiliza el formato clave: valor para cada dato.

Agregar listas y mapas: Usa guiones (-) para listas y indentación para mapas.

Incluir comentarios: Agrega comentarios con # para documentar el archivo.

4. Crea BD que tenga tablas y le insertan registros

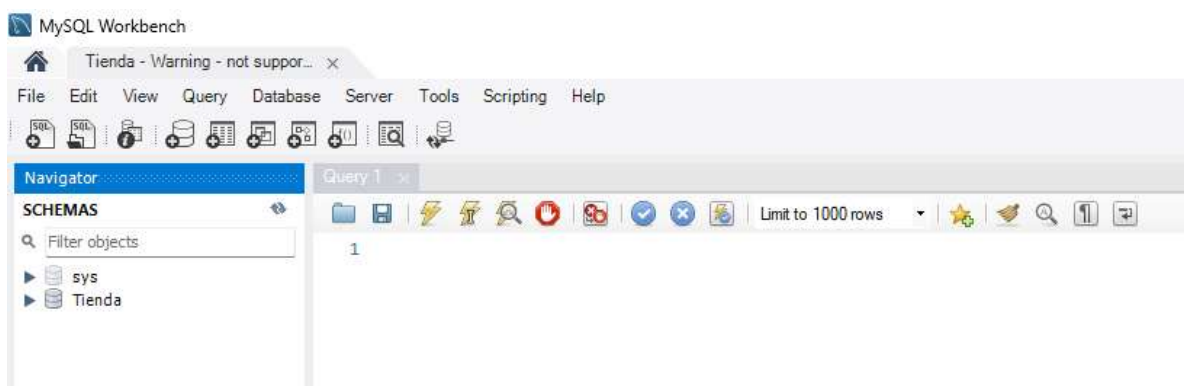
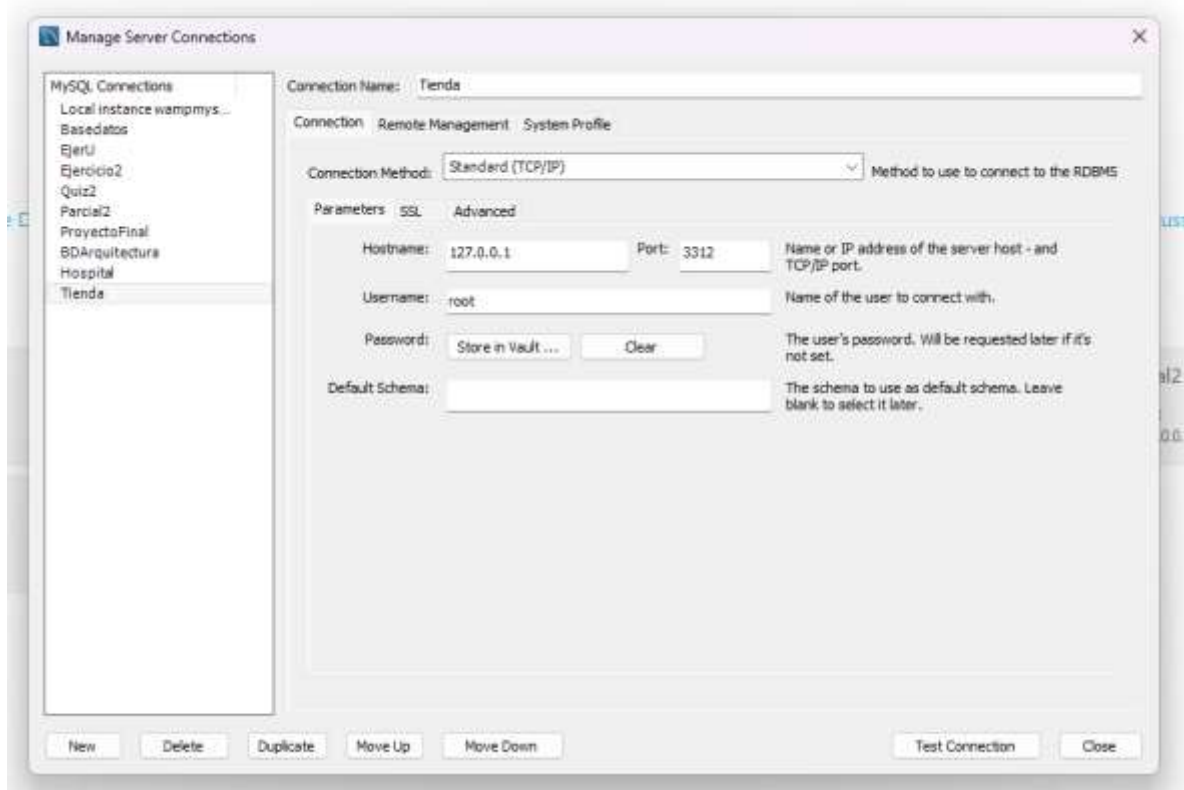


```
C:\Users\Asus\Documents\Uniminuto Valeria\Septimo semestre\Arquitectura de software\Prueba_yml>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
3415c1c29040   mysql:latest   "docker-entrypoint.s..." 23 hours ago   Up 2 minutes   33060/tcp, 0.0.0.0:3312->3306/tcp
mysql_Tienda

C:\Users\Asus\Documents\Uniminuto Valeria\Septimo semestre\Arquitectura de software\Prueba_yml>
```

```
docker-compose.yml X
C:\Users\Asus\Documents\Uniminuto Valeria\Septimo semestre\Arquitectura de software\Prueba_yml > docker-compose.yml
1  version: '3.8'
2
3  services:
4    mysql:
5      image: mysql:latest
6      container_name: mysql_Tienda
7      restart: always
8      environment:
9        MYSQL_ROOT_PASSWORD: "12345"
10       MYSQL_DATABASE: Tienda
11       MYSQL_USER: "user"
12       MYSQL_PASSWORD: "password"
13      ports:
14        - "3312:3306"
15      volumes:
16        - "C:/Users/Asus/Documents/Uniminuto Valeria/Septimo semestre/Arquitectura de software/Prueba_yml/mysql_data:/var/lib/mysql"
17        - ./init.sql:/docker-entrypoint-initdb.d/init.sql
18
19
20  volumes:
21    mysql_data:
22      driver: local
23
24
```

```
docker-compose.yml  init.sql X
C:\Users\Asus\Documents\Uniminuto Valeria\Septimo semestre\Arquitectura de software\Prueba_yml > init.sql
1  CREATE DATABASE Tienda;
2  USE Tienda;
3
4  CREATE TABLE Clientes(
5    id_cliente int primary key auto_increment,
6    nombre varchar(100) not null,
7    telefono varchar(15) unique not null
8  );
9
10 CREATE TABLE Pedidos(
11   id_pedido int primary key auto_increment,
12   id_cliente int not null,
13   fecha_pedido date not null,
14   total_compra decimal(10,2) not null,
15   FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente) ON DELETE CASCADE
16 );
17 CREATE TABLE Productos(
18   id_producto int primary key auto_increment,
19   nombre varchar(100) not null,
20   precio decimal(10,2) not null
21 );
22
```



Referencias

- AWS. (s.f). ¿Cuál es la diferencia entre YAML y JSON?. Amazon Web Services. <https://aws.amazon.com/es/compare/the-difference-between-yaml-and-json/>
- Whitfield, B . (28 de septiembre de 2023). YAML vs. JSON: A Dive Into 2 Popular Data Serialization Languages. builtin. <https://builtin.com/software-engineering-perspectives/yaml-vs-json>
- Docker docs. (s.f). Docker Compose. Docker docs. <https://docs.docker.com/compose/>
- Docker. (s.f). El fichero docker-compose.yml. iesgn. https://iesgn.github.io/curso_docker_2021/sesion5/docker-compose.html
- Red Hat. (3 de marzo de 2023). YAML: qué es y usos. Red Hat. <https://www.redhat.com/es/topics/automation/what-is-yaml>

- Estrategia del Contenido. (31 de octubre de 2019). Diseño de modelos de contenido estructurado con YAML y fin de octubre. Estrategia del Contenido.
<https://www.estrategiadelcontenido.com/disenio-de-modelos-de-contenido-estructurado-con-yaml/>