

TALLER YML

Docente WILLIAM MATAYANES

Diego Alejandro Sarmiento Rodríguez

Universidad De Cundinamarca

2025

DESARROLLO

QUE ES UN ARCHIVO YAML

Es un archivo de configuración que se utiliza en el desarrollo de software para intercambiar datos entre aplicaciones , lenguaje de programación que se basa en JSON , se utiliza para definir flujos de trabajo, dependencias y rutas de ejecución

- Para crear recursos de KUBERNETS como los pods, y objetivos de implementación
- Para definir los procesos de compilación , prueba e implementación de objetivos

Entre sus características mas usadas se puede destacar:

- Es un lenguaje de serialización de datos legible a humanos
- Proporciona un formato estandarizado para representar datos estructurados
- Utiliza sangria al estilo de Python para determinar la estructura
- Archivos YAML utilizan una extensión de archivo .yaml

DIFERENCIAS ENTRE YML Y JSON

Ambos dan formato a los datos de manera estandarizada para un intercambio entre aplicaciones de software , su uso texto también es legible a humanos, Ambos datos como pares de clave y valor, JSON admite objetos de datos como valores, mientras que YAML no, pero YML admite mas tipos de datos y utiliza de formas mas próxima el lenguaje natural

Entre condiciones de sintaxis de YML y JSON son:

Tipos de Datos

- Numero
- Booleano
- Nulo
- Cadena
- Matriz
- Objeto

YAML

Archivos de configuración en muchas herramientas y servicios de automatización como código.

Dentro de este rango YAML es la mejor función en cuanto a su formato pero normalmente se utiliza JSON que por motivos de compatibilidad cruzada muchas aplicaciones y servicios ya analizan el formato de datos JSON

JSON y YAML tienen un formato de serialización pero JSON da prioridad de uso de aplicaciones antes que al humano y YAML prioriza al humano antes que las aplicaciones

JSON No admite comentarios mientras que YAML admite los comentarios

COMO SE CREAN LOS ARCHIVOS YMAL

Crear un archivo llamado Docker—compose.yml

Definir los servicios de como y que se quieren ejecutar

Colocar los contenedores y volúmenes asociados

Luego en la carpeta Docker-- compose

Dirigirnos a la carpeta que contenga el archivo Docker—compose

Crear achivo de init.sql

Ejecutar el comando Docker—compose up -d

Luego colocar el comando Docker exec -it computadoras mysql -u root -p1234

COMANDOS

- `oc create myapp --image=me/myapp:v1`=Indica que la imagen que se usará para esta aplicación es `me/myapp:v1`, que se espera que esté en un registro de contenedores accesible.
- `-o yaml --dry-run >`=Simula la ejecución del comando sin crear realmente el recurso en el clúster.
- `myapp.yaml`= Especifica que la salida debe generarse en formato YAML.
- Esto es útil para ver la configuración antes de aplicarla en OpenShift.
- Crea y ejecuta la imagen `myapp` y dirige el objeto al archivo de salida `myapp.yaml`
- **DOCKER COMPOUSE** = se utiliza para definir y ejecutar aplicaciones Docker de varios contenedores

```
mysql> show databases;
+-----+
| Database |
+-----+
| computadoras |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.03 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_computadoras |
+-----+
| componentes |
| computadora_componentes |
| computadoras |
| historial_usos |
| usuarios |
+-----+
```

```
mysql> select * from usuarios;
+-----+-----+-----+
| id_usuario | nombre | email |
+-----+-----+-----+
| 1 | Juan | juan.perez@email.com |
| 2 | Maria | maria.gomez@email.com |
| 3 | Carlos | carlos.rodriguez@email.com |
| 4 | Laura | laura.fernandez@email.com |
| 5 | Pedro | pedro.ramirez@email.com |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select * from computadoras;
```

id_computadora	marca	modelo	anio
1	Dell	Inspiron 15	2021
2	HP	Pavilion x360	2022
3	Apple	MacBook Pro 14"	2023
4	Lenovo	ThinkPad X1 Carbon	2020
5	ASUS	ROG Zephyrus G14	2024

5 rows in set (0.00 sec)

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. <https://aka.ms/PSWindows>

```
PS C:\Users\diego\OneDrive\Documentos\computadorarC> docker-compose up -d
time="2025-03-18T21:01:49-05:00" level=warning msg="C:\\Users\\diego\\OneDrive\\Documentos\\computadorarC\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 1/2
  ✓ Network computadorar_db_network    Created                                0.3s
  - Container computadoras            Creating                                0.0s
Error response from daemon: Conflict. The container name "/computadoras" is already in use by container "42543984c2bc9e369453b726521b0e2dd7fafac4a51d63deaf185ec300eccbe". You have to remove (or rename) that container to be able to reuse that name.
PS C:\Users\diego\OneDrive\Documentos\computadorarC> docker exec -it computadoras mysql -u usuario -p computadoras
Enter password:
ERROR 1045 (28000): Access denied for user 'usuario'@'localhost' (using password: YES)
PS C:\Users\diego\OneDrive\Documentos\computadorarC> docker exec -it computadoras mysql -u usuario -p computadoras
Enter password:
ERROR 1045 (28000): Access denied for user 'usuario'@'localhost' (using password: YES)
PS C:\Users\diego\OneDrive\Documentos\computadorarC> docker exec -it computadoras mysql -u root -p1234
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 9.2.0 MySQL Community Server - GPL
```

```
mysql> show tables;
```

Tables_in_computadoras
componentes
computadora_componentes
computadoras
historial_usos
usuarios

5 rows in set (0.00 sec)

```
mysql> select * from componentes;
```

id_componente	tipo	descripcion
1	CPU	Intel Core i7
2	RAM	16GB DDR4
3	Almacenamiento	512GB SSD
4	CPU	AMD Ryzen 5
5	RAM	8GB DDR4
6	Almacenamiento	256GB SSD
7	CPU	Apple M1 Pro
8	RAM	16GB Unified Memory
9	Almacenamiento	1TB SSD
10	CPU	Intel Core i5
11	RAM	8GB DDR4
12	Almacenamiento	512GB SSD
13	CPU	AMD Ryzen 9
14	RAM	32GB DDR5
15	Almacenamiento	1TB SSD

```

1  version: '3.8'
2
3  services:
4    db:
5      image: mysql:latest
6      container_name: computadoras
7      environment:
8        MYSQL_ROOT_PASSWORD: 1234
9        MYSQL_DATABASE: computadoras
10       MYSQL_USER: usuario
11       MYSQL_PASSWORD: 1234
12      ports:
13        - "3309:3306"
14      volumes:
15        - ./init.sql:/docker-entrypoint-initdb.d/init.sql
16      networks:
17        - db_network
18
19  networks:
20    db_network:
21      driver: bridge

```

```

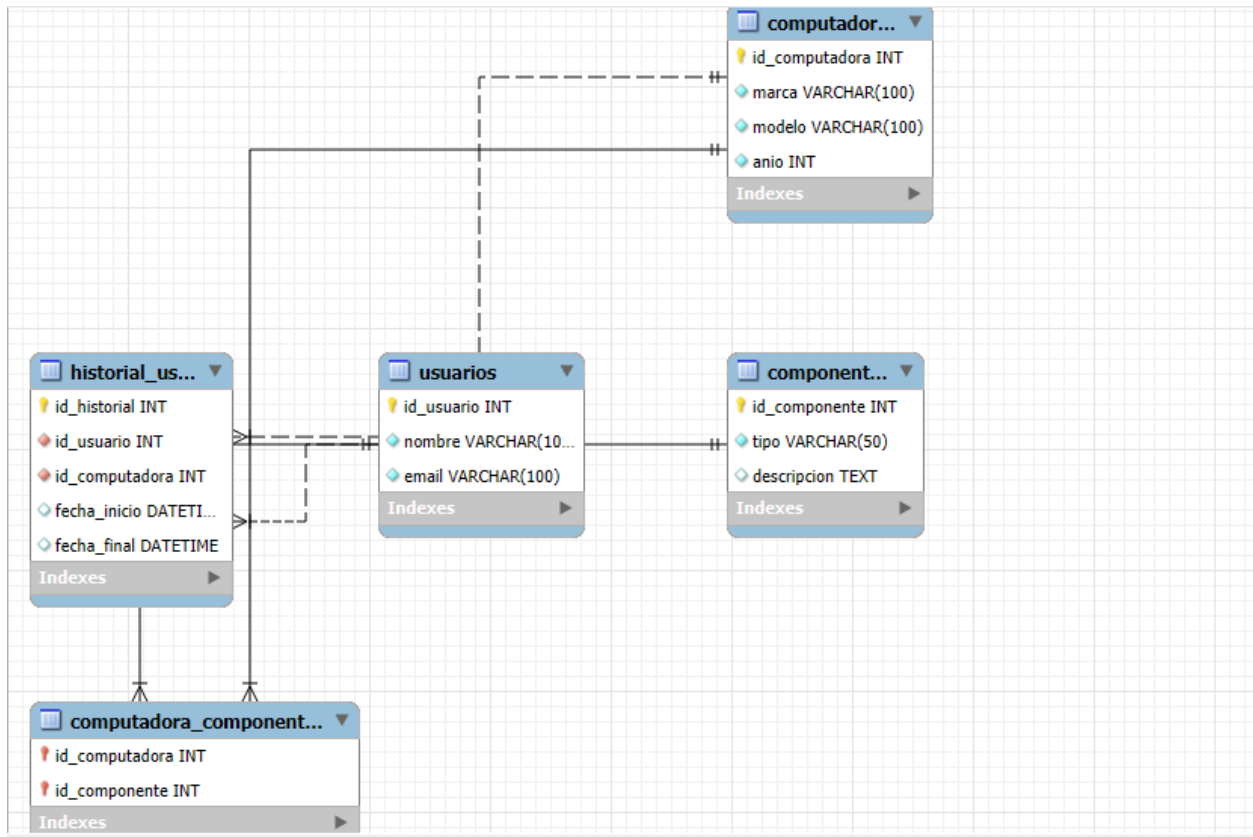
CREATE TABLE IF NOT EXISTS usuarios (
  id_usuario INT PRIMARY KEY AUTO_INCREMENT,
  nombre VARCHAR(100) NOT NULL,
  email VARCHAR(100) NOT NULL
);

CREATE TABLE IF NOT EXISTS computadoras (
  id_computadora INT PRIMARY KEY AUTO_INCREMENT,
  marca VARCHAR(100) NOT NULL,
  modelo VARCHAR(100) NOT NULL,
  anio INT NOT NULL
);

CREATE TABLE IF NOT EXISTS componentes (
  id_componente INT PRIMARY KEY AUTO_INCREMENT,
  tipo VARCHAR(100) NOT NULL,
  descripcion TEXT
);

CREATE TABLE IF NOT EXISTS computadora_componentes (
  id_computadora INT,
  id_componente INT,
  PRIMARY KEY (id_computadora, id_componente),
  FOREIGN KEY (id_computadora) REFERENCES computadoras(id_computadora),
  FOREIGN KEY (id_componente) REFERENCES componentes(id_componente)
);

```



UPDATE

```
mysql> update usuarios set nombre = "juanp" where id_usuaio=1;
ERROR 1054 (42S22): Unknown column 'id_usuaio' in 'where clause'
mysql> UPDATE usuarios
  -> SET nombre = 'Juan Prez'
  -> WHERE id_usuario = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from usuarios;
```

id_usuario	nombre	email
1	Juan Prez	juan.perez@email.com
2	Maria	maria.gomez@email.com
3	Carlos	carlos.rodriguez@email.com
4	Laura	laura.fernandez@email.com
5	Pedro	pedro.ramirez@email.com

5 rows in set (0.00 sec)

CONSULTA INNER JOIN

```
mysql> SELECT u.nombre, c.marca, c.modelo, h.fecha_inicio, h.fecha_final FROM Usuarios u INNER JOIN Historial_Usos h ON
u.id_usuario = h.id_usuario INNER JOIN Computadoras c ON h.id_computadora = c.id_computadora;
```

nombre	marca	modelo	fecha_inicio	fecha_final
Juan Prez	Dell	Inspiron 15	2024-03-15 08:00:00	2024-03-15 10:30:00
Juan Prez	HP	Pavilion x360	2024-03-15 09:15:00	2024-03-15 11:45:00
Carlos	Apple	MacBook Pro 14"	2024-03-15 10:00:00	2024-03-15 12:30:00
Laura	Lenovo	ThinkPad X1 Carbon	2024-03-15 11:00:00	2024-03-15 13:15:00
Pedro	ASUS	ROG Zephyrus G14	2024-03-15 12:00:00	2024-03-15 14:45:00

5 rows in set (0.00 sec)

LEFT JOIN

```
mysql> SELECT u.nombre, c.marca, c.modelo
-> FROM Usuarios u
-> LEFT JOIN Historial_Usos h ON u.id_usuario = h.id_usuario
-> LEFT JOIN Computadoras c ON h.id_computadora = c.id_computadora;
```

nombre	marca	modelo
Juan Prez	Dell	Inspiron 15
Juan Prez	HP	Pavilion x360
Maria	NULL	NULL
Carlos	Apple	MacBook Pro 14"
Laura	Lenovo	ThinkPad X1 Carbon
Pedro	ASUS	ROG Zephyrus G14

6 rows in set (0.00 sec)

CONSULTA

```
mysql> SELECT c.marca, c.modelo, u.nombre
-> FROM Computadoras c
-> RIGHT JOIN Historial_Usos h ON c.id_computadora = h.id_computadora
-> RIGHT JOIN Usuarios u ON h.id_usuario = u.id_usuario;
```

marca	modelo	nombre
Dell	Inspiron 15	Juan Prez
HP	Pavilion x360	Juan Prez
NULL	NULL	Maria
Apple	MacBook Pro 14"	Carlos
Lenovo	ThinkPad X1 Carbon	Laura
ASUS	ROG Zephyrus G14	Pedro

6 rows in set (0.00 sec)

REFERENCIAS

<https://aws.amazon.com/es/compare/the-difference-between-yaml-and->

<https://imaginaformacion.com/tutoriales/que-es-docker->

composejson/#:~:text=En%20particular%2C%20YAML%20es%20el,archivos%20de%20Docker%20y%20Kubernetes.

<https://aws.amazon.com/es/compare/the-difference-between-yaml-and-json/>

https://iesgn.github.io/curso_docker_2021/sesion5/docker-compose.html

<https://programacionymas.com/blog/como-funciona-inner-left-right-full-join>

CHATGPT:USO DE CONSULTAS A REALIZAR