

## **Taller**

David Santiago Noguera Perez - 827823

Corporación Universitaria Minuto de Dios

Ing. William Alexander Matallana Porras

Arquitectura de software

2025

1. Archivos yml vs json
2. Docker-compose.yml su uso
3. Como se crea un contenedor usando yml
4. Crear una bd que tenga tablas e insertar registros

## DESARROLLO

### 1. Diferencias entre YAML y JSON

Tanto **YAML** (.yml o .yaml) como **JSON** (.json) son formatos de serialización de datos ampliamente utilizados en el desarrollo de software. Ambos permiten estructurar y almacenar información, pero tienen diferencias clave en sintaxis, uso, compatibilidad y flexibilidad.

#### Sintaxis y legibilidad

**YAML:** Usa indentación con espacios para estructurar los datos, evitando el uso de llaves {} y corchetes []. Esto lo hace más legible y fácil de escribir manualmente.

**JSON:** Utiliza una estructura más rígida basada en llaves {}, corchetes [] y comas ,, lo que puede ser más difícil de leer a simple vista.

#### Comentarios

**YAML permite comentarios** usando #, lo que facilita la documentación en archivos de configuración.

**JSON no admite comentarios**, lo que puede ser una limitación en algunos casos.

#### Uso y compatibilidad

**JSON** es el formato más utilizado para intercambio de datos en APIs y almacenamiento estructurado en aplicaciones web y móviles. Es compatible de forma nativa con muchos lenguajes como JavaScript, Python, Java, etc.

**YAML** se usa más en **archivos de configuración** de sistemas, infraestructura y herramientas como Docker, Kubernetes, Ansible y GitHub Actions.

#### Soporte en lenguajes de programación

**JSON es más compatible** con la mayoría de los lenguajes de programación, ya que su estructura es similar a la de los objetos en JavaScript.

**YAML generalmente requiere bibliotecas adicionales** para ser procesado en algunos lenguajes.

#### Complejidad y facilidad de escritura

**JSON es más estricto** en su sintaxis. Debes asegurarte de usar comas correctamente y que las claves siempre estén entre comillas dobles ("").

**YAML es más flexible**, ya que permite omitir comillas en cadenas de texto simples y no requiere comas para separar elementos.





## 2. ¿Para qué se usa docker-compose.yml?

El archivo docker-compose.yml es un **archivo de configuración en YAML** utilizado por **Docker Compose** para definir y gestionar **contenedores Docker** de manera sencilla. Permite especificar múltiples servicios, redes y volúmenes en un solo archivo, facilitando la ejecución de aplicaciones con varios contenedores.

3. Para crear un archivo de Docker compose crearemos un fichero .yml que se llame **docker-compose.yml** y le daremos la información necesaria para levantar el contenedor. Este lo levantaremos con un **docker-compose up -d**

```
services:
  mysql:
    image: mysql:latest
    container_name: contenedor_compose
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: 1234
      MYSQL_DATABASE: videojuegos
    ports:
      - "3310:3306"
    volumes:
      - "C:/Users/Esteban/Desktop/Contenedores/compose:/var/lib/mysql"
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql
volumes:
  mysql_data:
```

```
C:\Users\Esteban\Desktop\Santiago\Trabajos\Ejercicios\compose1>docker-compose up -d
[+] Running 2/2
  Network compose1_default      Created
  Container contenedor_compose  Started
```

<input type="checkbox"/>		compose1	-	-	-	0.56%	4 minutes ago	
<input type="checkbox"/>		contenedor_c	ec388172ae23	mysql:latest	3310:3306	0.56%	4 minutes ago	

4. Ahora crearemos la conexión al workbench para verificar que todo este corriendo en orden

