

## **Investigación**

Angie Lorena Jiménez Porras

Ing. Sistemas, UNIMINUTO

William Alexander Matallana Porras

Arquitectura de software

20/03/2025

1 YAMLvs JSON: son dos formatos de serialización de datos que proporcionan un mecanismo de intercambio de datos legibles. Representan datos como pares de clave y valor, JSON admite objetos de datos como valores sin embargo YAML no

#### Diferencias

- JSON:
  - a simple vista el formato JSON se asemeja a una máquina por su escritura; usa llaves para objetos, dos puntos para separar pares clave-valor, corchetes para matrices y comillas para delimitar cadenas. Es desarrollado a partir de un subconjunto de JavaScript, por lo tanto, su escritura en documentos se asemeja a un código de programación
- Tipos de datos:
- JSON admite los siguientes tipos de datos: número, booleano, nulo, cadena, matriz y objeto
- YAML:
  - se parece más a una estructura de datos estructurado en papel siendo más fácil de leer, para la escritura usa: sangría para representar los objetos, dos puntos para separar clave-valor, guion para matrices, almohadillas para hacer un comentario
- Tipos de datos:
- YAML admite todos los tipos de datos anidados que comprenden secuencias, escalares y asignaciones

2 docker compose.yml

Archivo de configuración donde se define y documenta las dependencias de una aplicación multi contenedor. Se usa para gestionar aplicaciones compuestas, contienen una imagen del contenedor, mapea los puertos del contenedor, monta el volumen, define las variables, especifica las dependencias y servicios que se van a ejecutar

3 para crear un contenedor usando YML se usa la estructura:

```
C: > Users > yallg > Desktop > ejercicio_compose > compose.yml
1  services:
2    mysql:
3      image: mysql:latest
4      container_name: container_min
5      restart: always
6      environment:
7        MYSQL_ROOT_PASSWORD: "0000"
8        MYSQL_DATABASE: Concierto_Mexico
9        MYSQL_USER: root
10       MYSQL_PASSWORD: "0000"
11     ports:
12       - "3308:3306"
13     volumes:
14       - ./BDArquitectura:/var/lib/mysql
15       - ./init:/docker-entrypoint-initdb.d
16
17   volumes:
18     mysql_data:
19
20
```

En la cual se especifica el nombre que va a tener el container, la contraseña del puerto, la creación del volumen y el archivo .sql que se va a usar.