



INVESTIGACION LINEA 3

ESTUDIANTE: JUAN SEBASTIAN ROJAS SANCHEZ

PROFESOR:

Ing. William Alexander Matallana Porras

UNIVERSIDAD DE CUNDINAMARCA EXTENSION CHIA

PROGRAMA DE INGENIERIA DE SISTEMAS

FACULTAD DE INGENIERIA

2025

ARCHIVO YAML EN EL DESARROLLO DE SOFTWARE: GUÍA COMPLETA

OBJETIVO GENERAL

Analizar el uso de los archivos YAML en el desarrollo de software, destacando sus características, diferencias con JSON y su aplicación en la configuración y despliegue de servicios, incluyendo la creación de una base de datos con cinco tablas.

OBJETIVOS ESPECÍFICOS

- A. Describir qué es un archivo YAML y su propósito dentro del desarrollo de software.
- B. Comparar YAML con JSON, resaltando sus diferencias y estableciendo criterios sobre cuándo utilizar uno u otro.
- C. Explicar el proceso de creación de archivos YAML, detallando su estructura y sintaxis.
- D. Implementar una base de datos con cinco tablas y desplegar su servicio mediante archivos YAML.

INTRODUCCIÓN

YAML (YAML Ain't Markup Language) se ha convertido en un formato de serialización de datos fundamental en el ecosistema de desarrollo moderno. Su diseño centrado en la legibilidad humana lo ha posicionado como una alternativa preferida para configuraciones de aplicaciones, especialmente en entornos donde desarrolladores y operadores necesitan interactuar frecuentemente con estos archivos. A diferencia de formatos como XML o JSON, YAML utiliza indentación y una sintaxis minimalista que facilita su escritura y mantenimiento.

INVESTIGACIÓN

Características fundamentales de YAML

- Legibilidad: Diseñado priorizando la facilidad de lectura para humanos
- Estructura jerárquica: Utiliza indentación para representar relaciones anidadas
- Tipos de datos: Soporta cadenas, números, booleanos, listas, mapas, nulos, fechas y más
- Referencias: Permite crear anclas (&) y alias (*) para evitar repetición
- Comentarios: Admite comentarios con el símbolo #
- Multilinealidad: Ofrece múltiples estilos para representar texto multilinea

COMPARATIVA CON JSON

La investigación muestra que YAML supera a JSON en legibilidad y expresividad, mientras que JSON destaca en velocidad de procesamiento. Esta diferencia se debe principalmente a:

- A. Complejidad de parseo: YAML requiere más procesamiento por su sintaxis flexible
- B. Ambigüedad: YAML tiene más formas de representar el mismo dato
- C. Especificación: La especificación de YAML es más compleja (23K palabras vs 1.8K de JSON)

CASOS DE USO ÓPTIMOS

- YAML se utiliza predominantemente en:
- Configuración de infraestructura (Kubernetes, Docker Compose)
- Pipelines de CI/CD (GitHub Actions, GitLab CI, Jenkins)
- Configuración de aplicaciones (Spring Boot, Django)
- Definición de infraestructura como código (Ansible, Terraform)

Características Técnicas de YAML

Estructura Básica

YAML se basa en tres bloques de construcción fundamentales:

- Escalares: Valores simples como cadenas, números, booleanos
- Secuencias: Listas ordenadas de elementos
- Mapeos: Conjuntos de pares clave-valor no ordenados

La característica distintiva de YAML es su uso de indentación significativa para denotar estructura jerárquica, similar a Python. Esta característica elimina la necesidad de delimitadores explícitos, lo que mejora significativamente la legibilidad.

Tipos de Datos

YAML soporta una amplia gama de tipos de datos, incluyendo:

- Cadenas: Con múltiples estilos de representación (literal, plegado, comillas simples, comillas dobles)
- Números: Enteros, flotantes, octal, hexadecimal
- Booleanos: true/false, yes/no, on/off
- Nulos: null, ~, "
- Fechas y horas: ISO8601
- Conjuntos: Colecciones no ordenadas de elementos únicos

- Pares ordenados: Tuplas de dos elementos
- Binarios: Datos codificados en base64
- Tipos definidos por la aplicación: Extensible a través de etiquetas

Esta riqueza de tipos nativos supera significativamente a JSON, que solo soporta cadenas, números, booleanos, null, objetos y arreglos.

Problemas y Limitaciones

La flexibilidad de YAML también introduce desafíos:

- Ambigüedad de tipos: YAML intenta inferir tipos, lo que puede llevar a comportamientos inesperados
- Complejidad de la especificación: La especificación completa es extensa (23,000+ palabras)
- Inconsistencias entre implementaciones: Diferentes parsers pueden interpretar el mismo YAML de manera diferente
- Sensibilidad a la indentación: Los errores de espaciado son difíciles de detectar visualmente
- Problemas de seguridad: La deserialización insegura puede llevar a ejecución de código remoto en algunos lenguajes

YAML vs JSON

Aspecto	YAML	JSON
Sintaxis	Basada en indentación	Basada en llaves y corchetes
Comentarios	Soportados	No soportados
Tipos de datos	Rico conjunto de tipos nativos	Conjunto limitado de tipos
Referencias	Soporta anclas y alias	No soporta referencias
Multidocumento	Soportado	No soportado
Texto multilinea	Múltiples formatos nativos	Requiere secuencias de escape
Legibilidad	Alta, optimizada para humanos	Moderada, optimizada para máquinas
Popularidad en APIs	Baja a moderada	Dominante

Complejidad de parseo	Alta	Baja
Rendimiento	Típicamente más lento	Más rápido
Seguridad	Potencialmente problemática	Relativamente segura

Estudios empíricos han demostrado que los archivos YAML suelen ser un 30% más concisos que sus equivalentes JSON y que los desarrolladores cometen significativamente menos errores al escribir YAML manualmente en comparación con JSON.

YAML vs TOML

Aspecto	YAML	TOML
Diseño	General, flexible	Específico para archivos de configuración
Complejidad	Alta	Baja
Curva de aprendizaje	Pronunciada	Suave
Estructuras anidadas	Fácil vía indentación	Mediante notación de tablas
Capacidades	Muy extenso	Intencionalmente limitado
Ambigüedad	Potencialmente alta	Baja

CREACIÓN DE ARCHIVOS YAML

1. Comprender YAML: es un formato de datos estructurado y fácil de leer. Se usa en configuración, almacenamiento de datos y definición de estructuras.

2. Elegir un Editor de Texto: Puedes usar editores como Notepad++, VS Code, Sublime Text o incluso un simple Bloc de notas.

3. Definir la Estructura: YAML usa jerarquías con espacios en lugar de llaves o paréntesis. La clave es mantener una indentación adecuada.

4. Usar Claves y Valores: Los datos se organizan en pares clave-valor. Se pueden definir valores simples como números o texto, así como listas y objetos anidados.

5. Guardar con la Extensión Correcta

Debes guardar el archivo con la extensión `.yaml` o `.yml` para que sea reconocido por las herramientas que lo utilizan.

6. Validar el Archivo

Usa herramientas en línea o comandos específicos para comprobar que el formato sea correcto antes de usarlo.

CONCLUSIONES

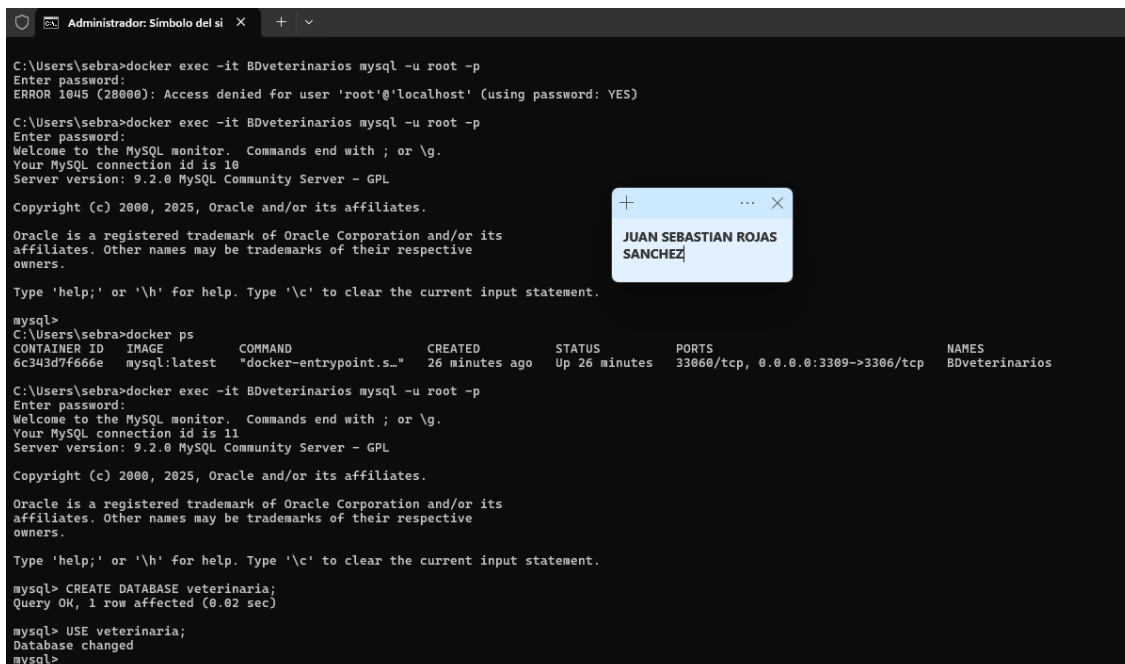
Los archivos YAML son esenciales en el desarrollo de software, ya que permiten una configuración clara y estructurada en diversos entornos, facilitando la automatización y gestión de servicios.

YAML y JSON presentan diferencias clave: YAML es más legible y adecuado para archivos de configuración, mientras que JSON es más estructurado y utilizado en el intercambio de datos entre sistemas.

La creación de archivos YAML requiere seguir una sintaxis estricta basada en la indentación, lo que permite una correcta organización y representación de datos.

El despliegue de servicios mediante YAML es una práctica común en infraestructura como código, permitiendo la definición de bases de datos y otros recursos de manera eficiente y reproducible.

ANEXO PANTALLAZOS



```
Administrador: Símbolo del si x + v
C:\Users\sebra>docker exec -it BDveterinarios mysql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

C:\Users\sebra>docker exec -it BDveterinarios mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
C:\Users\sebra>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
6c343d7f666e   mysql:latest  "docker-entrypoint.s..."  26 minutes ago Up 26 minutes  33060/tcp, 0.0.0.0:3309->3306/tcp    BDveterinarios

C:\Users\sebra>docker exec -it BDveterinarios mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE veterinaria;
Query OK, 1 row affected (0.02 sec)

mysql> USE veterinaria;
Database changed
mysql>
```

```
Administrador: Símbolo del si X + v
Microsoft Windows [Versión 10.0.26100.2476]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\sebra>docker pull mysql:Lasted
Error response from daemon: manifest for mysql:Lasted not found: manifest unknown: manifest unknown

C:\Users\sebra>docker pull mysql:Latest
Error response from daemon: manifest for mysql:Latest not found: manifest unknown: manifest unknown

C:\Users\sebra>docker pull mysql:latest
latest: Pulling from library/mysql
804bb8ae89de: Pull complete
1b515e7ceb69: Pull complete
ea11c0a9f08: Pull complete
8d18181893b8: Pull complete
e0a910cc8604: Pull complete
bc0c792ca096: Pull complete
8d73d2a73425: Pull complete
4a7e00d873b9: Pull complete
27a2553d6a80: Pull complete
69e76254f502: Pull complete
Digest: sha256:9b9d0aab4860798acff13d2a0ece3bc26639fe18b83fa5cd3e3d0e16b3ed05dd
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest

C:\Users\sebra>docker --version
Docker version 28.0.1, build 068a01e

C:\Users\sebra>docker run mysql
2025-03-18 00:52:31+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 9.2.0-1.el9 started.
2025-03-18 00:52:32+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2025-03-18 00:52:32+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 9.2.0-1.el9 started.
2025-03-18 00:52:32+00:00 [ERROR] [Entrypoint]: Database is uninitialized and password option is not specified
You need to specify one of the following as an environment variable:
- MYSQL_ROOT_PASSWORD
- MYSQL_ALLOW_EMPTY_PASSWORD
- MYSQL_RANDOM_ROOT_PASSWORD

C:\Users\sebra>
```

+

...

×

JUAN SEBASTIAN ROJAS
SANCHEZ

```
Administrador: Símbolo del si X + v

C:\Users\sebra>docker pull mysql:latest
latest: Pulling from library/mysql
Digest: sha256:9b9d0aab4860798acff13d2a0ece3bc26639fe18b83fa5cd3e3d0e16b3ed05dd
Status: Image is up to date for mysql:latest
docker.io/library/mysql:latest

C:\Users\sebra>docker run -d --name BDveterinarios -p 3309:3306 -e MYSQL_ROOT_PASSWORD=12345 mysql:latest
6c343d7f666e89becd16a8c43aba581a39cc82e2355ad4088bbe09ea0903ba70

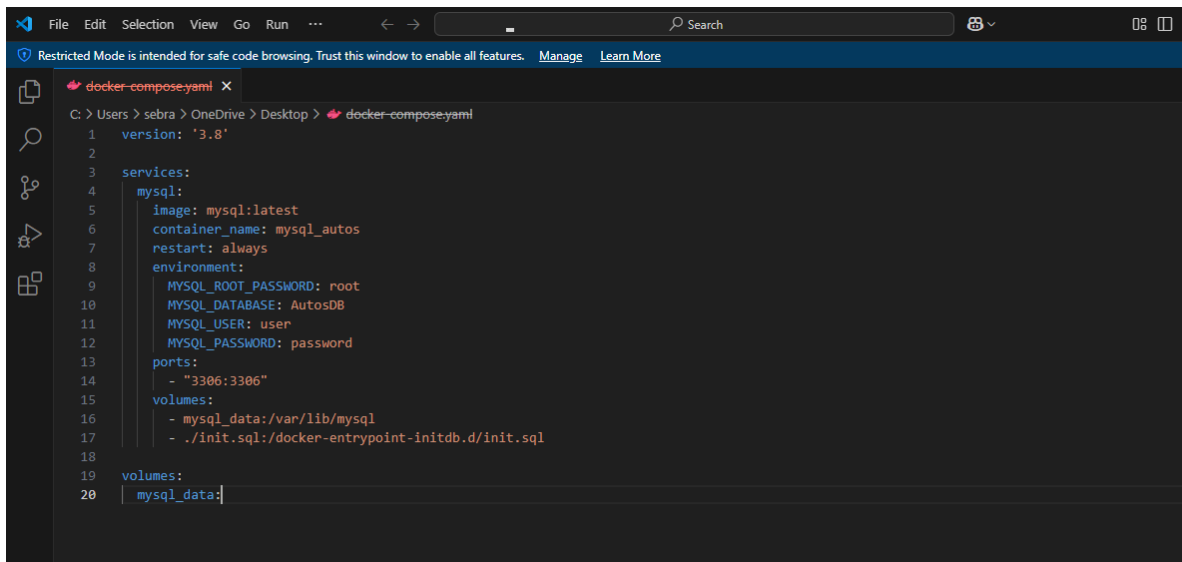
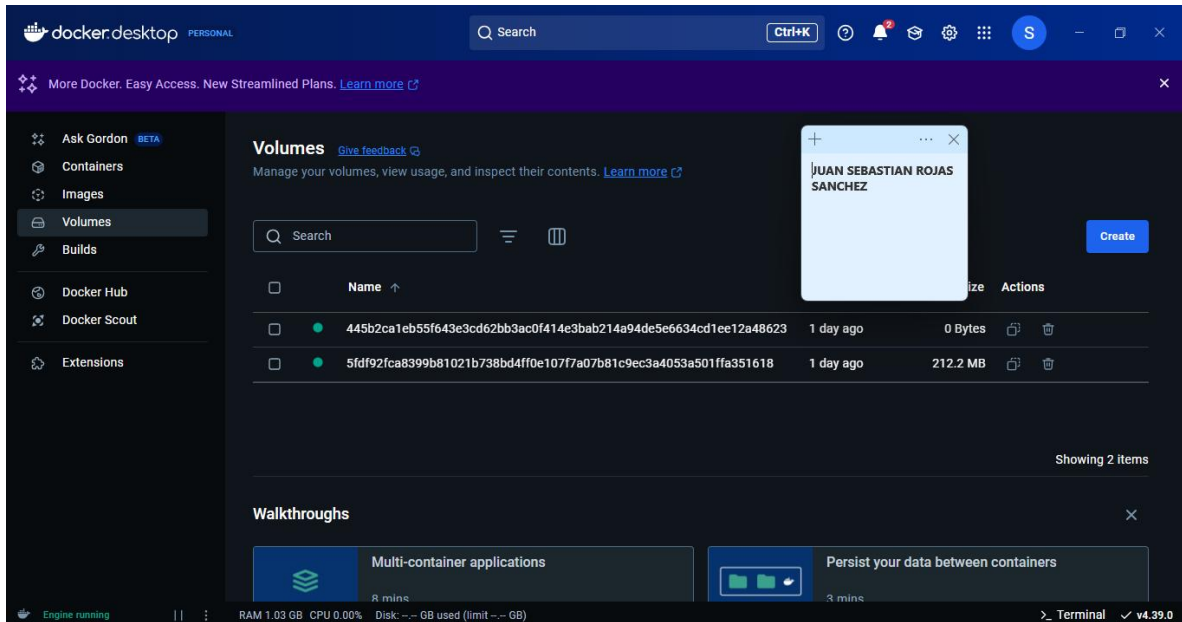
C:\Users\sebra>
```

+

...

×

JUAN SEBASTIAN ROJAS
SANCHEZ



BIBLIOGRAFIAS

YAML frente a JSON: diferencia entre los formatos de serialización de datos - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/compare/the-difference-between-yaml-and-json/>

Mallón, X. (2024, 27 septiembre). Fichero YAML: ¿Qué es y para qué sirve? [2025] | KeepCoding. KeepCoding Bootcamps. <https://keepcoding.io/blog/que-es-y-para-que-sirve-el-fichero-yaml/>

Equipo editorial de IONOS. (2022, 31 mayo). Introducción a YAML. IONOS Digital Guide.
<https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/yaml-el-ligero-framework-css-para-expertos/>