

Ejercicio clase 18 de marzo

1. ¿Qué es un archivo YAML?

YAML es un lenguaje versátil y legible por humanos para serializar datos, que se utiliza a menudo para escribir archivos de configuración.

Proporciona un formato estandarizado para representar datos estructurados de forma que sea fácil de entender para los humanos y de interpretar para las máquinas, el lenguaje está pensado para datos y no para documentos.

YAML se desarrolló pensando en la simplicidad y la legibilidad. YAML utiliza una sintaxis clara y minimalista basada en:

- Sangrías
- Pares clave-valor
- Convenciones intuitivas

Este enfoque permite a desarrolladores y usuarios expresar estructuras de datos complejas en un formato que se asemeja al lenguaje natural y es fácil de entender a primera vista.

2. ¿Para qué se utiliza en desarrollo de software?

Como la legibilidad ocupa un lugar central, YAML es especialmente adecuado para diversas aplicaciones, por ejemplo, para archivos de configuración (config) e intercambio de datos entre distintos sistemas. Su estructura sencilla e intuitiva hace que YAML pueda utilizarse en muchos ámbitos y permite definir y organizar los datos de forma clara y comprensible.

3. Diferencia entre YAML y JSON.

YAML y JSON son dos formatos de serialización de datos que proporcionan un mecanismo de intercambio de datos legible por humanos. Dan formato a los datos de manera estandarizada para su intercambio entre aplicaciones de software, y su texto también es legible por humanos. Ambos representan datos como pares de clave y valor. JSON admite objetos de datos como valores, mientras que YAML no. Sin embargo, YAML admite más tipos de datos y utiliza de forma más próxima el lenguaje natural para que los desarrolladores puedan utilizarlo.

Representación de datos: El aspecto y la sintaxis de JSON y YAML son similares pero ligeramente diferentes.

A simple vista, el formato JSON se asemeja más a una máquina en su representación de datos que YAML. Un mensaje JSON incluye estos símbolos:

- Llaves para objetos
- Dos puntos para separar pares clave-valor
- Corchetes para matrices

- Comillas para delimitar cadenas

JSON se desarrolló a partir de un pequeño subconjunto de JavaScript. Por tanto, su sintaxis hace que los documentos JSON se parezcan al código de programación.

A diferencia de JSON, el formato YAML se parece más a cómo alguien escribiría datos estructurados en papel. Se diseñó específicamente para que fuera más fácil de leer para los humanos. Utiliza lo siguiente:

- Sangría para representar objetos
- Dos puntos para separar pares clave-valor
- Guiones para matrices
- Almohadillas para señalar un comentario

YAML se creó originalmente para simplificar XML.

Ejemplo de YAML y JSOM

YAML	JSON
type: Camiseta	{
price: 20,00	"product": {
sizes:	"type": "Camiseta",
• S	"price": 20,00,
• M	"sizes": ["S", "M", "L"]
• L	"reviews": {
reviews: # Nota sobre las reseñas	{ "username": "usuario1", "rating": 4, "created_at": "2023-04-19T12:30:00Z" },
• de usuario: usuario1	{ "username": "usuario2", "rating": 5, "created_at": "2023-05-02T15:00:00Z" }
rating: 4	}
created_at: 2023-04-19T12:30:00Z	}
• username: usuario2	}
rating: 5	}
created_at: 2023-05-02T15:00:00Z	}

Diferencias clave

Debido a la sintaxis de YAML, se considera más fácil leer, comprender y escribir un archivo YAML que uno JSON, sobre todo para los no desarrolladores. A continuación, analizaremos otras diferencias clave.

Tipos de datos

El formato JSON admite los siguientes tipos de datos:

- Número
- Booleano
- Nulo
- Cadena
- Matriz
- Objeto (colección de pares clave-valor anidados)

El formato YAML admite todos los tipos de datos nativos de cualquier lenguaje de programación dinámico. Por ejemplo, admite colecciones de datos anidados formadas por secuencias (listas), escalares (como números y cadenas) y

asignaciones (pares clave-valor). Este sistema le permite crear estructuras de datos más complejas.

Compatibilidad integrada: JSON se utiliza de manera generalizada en combinación con muchos lenguajes de programación diferentes. Por ejemplo, JavaScript ofrece compatibilidad integrada para analizar JSON. La biblioteca estándar de Python también incluye JSON, y PHP 5.2 incorpora una biblioteca JSON. Del mismo modo, más de otros 50 lenguajes populares son compatibles con JSON mediante distintos paquetes y bibliotecas.

En cambio, ningún lenguaje conocido tiene compatibilidad integrada con YAML. En su lugar, los desarrolladores tienen que instalar y utilizar bibliotecas. Dos analizadores populares de YAML son PyYAML para Python y LibYAML para C.

Control de versiones: Se puede versionar tanto JSON como YAML usando un software de control de versiones como Git. Sin embargo, se considera más fácil analizar y comprender los cambios en los archivos YAML entre versiones.

Comentarios: El formato de datos JSON no admite comentarios. Para sortear esta situación, los desarrolladores tienen que definir un par clave-valor para los comentarios.

En un archivo YAML, los comentarios van precedidos por una almohadilla

4. ¿Cuándo se recomienda usar uno sobre el otro?

Gracias al soporte generalizado y la integración con JavaScript, JSON es un formato de serialización de datos más popular que YAML para la mayoría de los casos de uso. JSON se usa ampliamente en comunicaciones de software distribuidas, aplicaciones web, archivos de configuración y API.

Aunque YAML pueda parecer una mejor opción en función de la tipificación de datos y su formato legible por humanos, normalmente se prefiere JSON por motivos de compatibilidad cruzada. Esto se debe a que muchas aplicaciones y servicios ya analizan el formato de datos JSON.

Por otro lado, YAML ha ganado una fuerte presencia en determinados ámbitos de la computación debido a su legibilidad y soporte de comentarios. En particular, YAML es el principal formato de serialización de datos para archivos de configuración en muchas herramientas y servicios de automatización, DevOps e infraestructura como código (IaC). Por ejemplo, YAML se usa a menudo en archivos de Docker y Kubernetes.

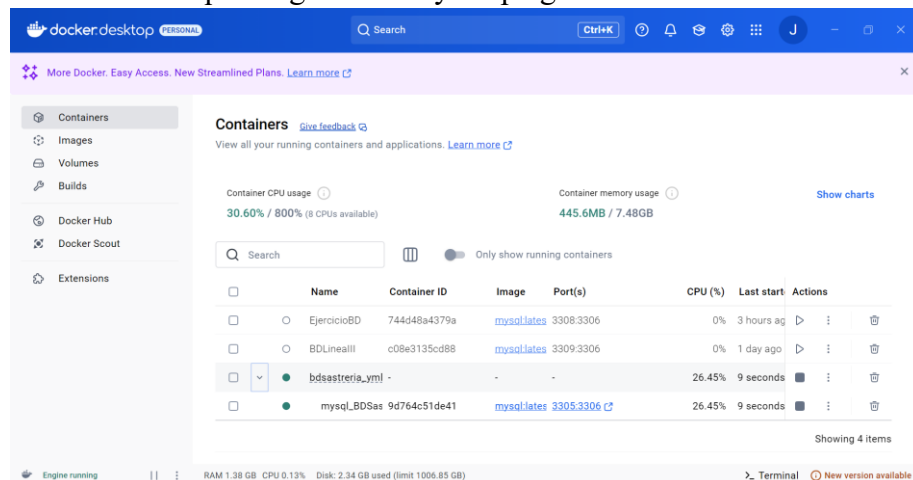
5. Como se crean los archivos yml

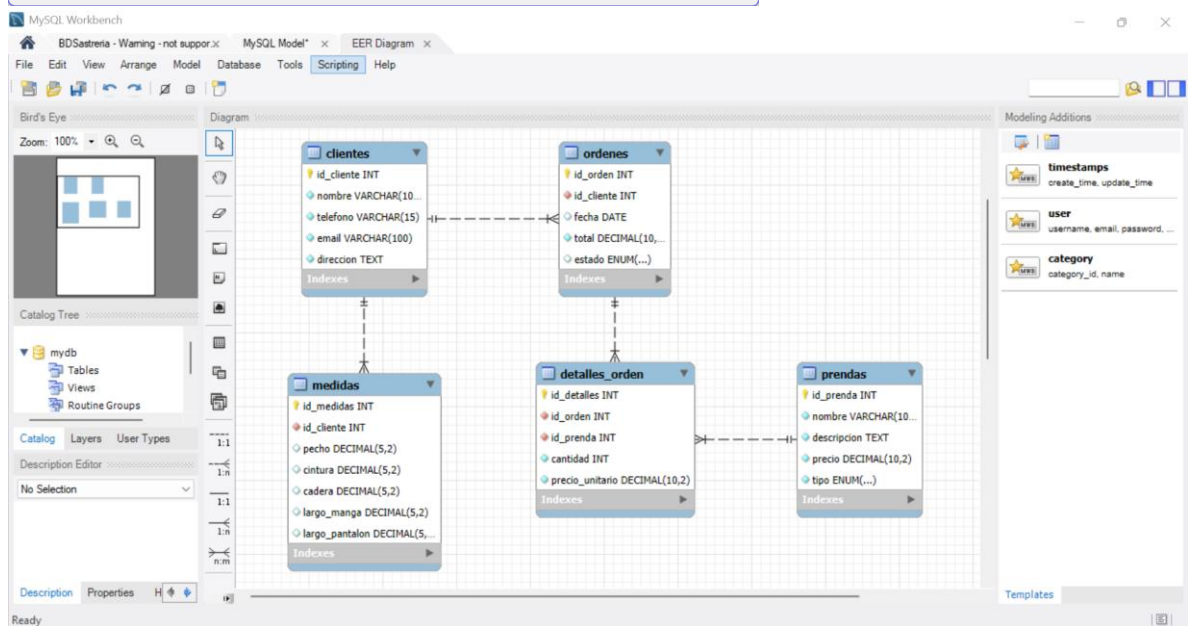
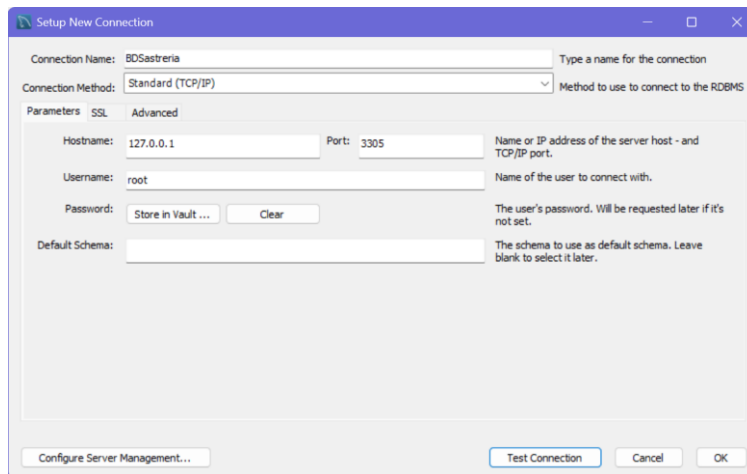
Estos son los pasos generales para escribir un archivo YAML:

- Elige un editor de texto: Puedes usar cualquier editor de texto para escribir un archivo YAML. Algunos ejemplos son Notepad++, Sublime Text, Atom, etc.

- Determina la estructura de tus datos: YAML utiliza la sangría para determinar la estructura de tus datos. Empieza por decidir cómo quieres organizarlos.
- Comience con la estructura básica: los archivos YAML empiezan con tres guiones (— -) y terminan con tres puntos (...). Todo lo que esté entre estos símbolos forma parte de su archivo YAML.
- Añadir pares clave-valor: YAML utiliza pares clave-valor para almacenar datos. La clave y el valor se separan con dos puntos, y el valor puede ser una cadena, un número u otro tipo de dato.
- Usar espacios en blanco para crear estructuras anidadas: YAML utiliza sangría para crear estructuras anidadas, como listas y diccionarios. Por ejemplo, si desea crear una lista, comience cada elemento en una nueva línea, con la misma sangría de espacios.
- Guardar el archivo: guarde el archivo con una extensión, yml o .yaml.
- Verificar la sintaxis: Antes de usar el archivo YAML, asegúrese de que la sintaxis sea correcta mediante un linter YAML. Una sintaxis incorrecta puede causar errores en los datos.

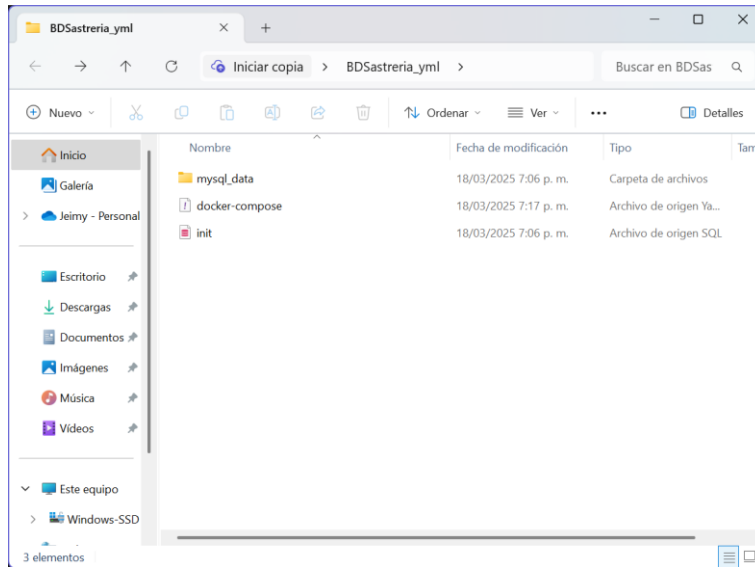
6. Crear una BD que tenga 5 tablas y desplegar el servicio usando archivos yml





Comandos Docker usados

1. Iniciar una terminal desde la carpeta donde se guardó el archivo Docker-compose.yml y el archivo init.sql



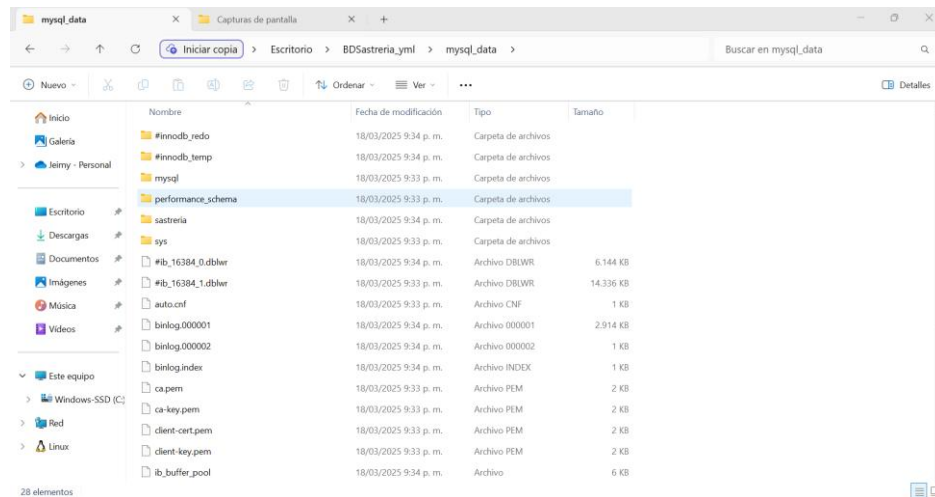
2. En la terminal escribir el comando docker compose up para iniciar la pila de aplicaciones y agrega el indicador -d para ejecutar todo en segundo plano, el comando debe quedar docker compose up -d
3. Para remover los volúmenes persistentes en caso de que hallan errores se usa el comando docker-compose down -v
4. Para reiniciar el contener se usa el comando docker-compose up --build

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

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

PS C:\Users\Jeimy\Desktop\BDSastreria.yml> docker-compose down -v
[*] Running 2/2
✓Container mysql_BDSastreria      Removed      0.0s
✓Network bdsastreria.yml_default Removed      0.7s
PS C:\Users\Jeimy\Desktop\BDSastreria.yml> docker-compose up --build
[*] Running 2/2
✓Network bdsastreria.yml_default Created        0.3s
✓Container mysql_BDSastreria      Created        0.2s
Attaching to mysql_BDSastreria
mysql_BDSastreria | 2025-03-19 02:17:40+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 9.2.0-1.el9 start
ed.
mysql_BDSastreria | 2025-03-19 02:17:40+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql_BDSastreria | 2025-03-19 02:17:40+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 9.2.0-1.el9 start
ed.
mysql_BDSastreria | 2025-03-19 02:17:41+00:00 [Note] [Entrypoint]: Initializing database files
mysql_BDSastreria | 2025-03-19T02:17:41.283405Z 0 [System] [MY-015017] [Server] MySQL Server Initialization - start.
mysql_BDSastreria | 2025-03-19T02:17:41.285065Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 9.2.0) initial
izing of server in progress as process 80
mysql_BDSastreria | 2025-03-19T02:17:41.303402Z 0 [Warning] [MY-010159] [Server] Setting lower_case_table_names=2 becau
se file system for 'var/lib/mysql/' is case insensitive
mysql_BDSastreria | 2025-03-19T02:17:41.331104Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql_BDSastreria | 2025-03-19T02:17:44.748373Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql_BDSastreria | 2025-03-19T02:17:50.279647Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empt
y password ! Please consider switching off the --initialize-insecure option.
mysql_BDSastreria | 2025-03-19T02:17:56.132461Z 0 [System] [MY-015018] [Server] MySQL Server Initialization - end.
  
```



Referencias

¿Qué es YAML? (2024, octubre 4). Ibm.com. <https://www.ibm.com/es-es/topics/yaml>

YAML: qué es, usos, sintaxis y ejemplos. (s/f). Redhat.com. Recuperado el 18 de marzo de 2025, de <https://www.redhat.com/es/topics/automation/what-is-yaml>

(S/f). Amazon.com. Recuperado el 18 de marzo de 2025, de <https://aws.amazon.com/es/compare/the-difference-between-yaml-and-json/>

BuildPiper. (2023, marzo 7). All you need to know about YAML Files! BuildPiper. <https://medium.com/buildpiper/all-you-need-to-know-about-yaml-files-8fa319b1f26f>

https://docs.docker.com/get-started/workshop/08_using_compose/