

TALLER MONGO

Manuel Santiago Malpica Daza

ID: 833776

Ingeniería de Sistemas

Semestre 7

Bases De Datos

NRC-60747

UNIMINUTO

Corporación Universitaria Minuto de Dios

Docente:

William Alexander Matallana Porras

Tabla de Contenido

1. Introducción
2. Objetivos
3. Desarrollo del Ejercicio 3.1. Tipo de Base de Datos y Diferencias con SQL 3.2. Colecciones vs Tablas en MongoDB 3.3. Formato de Almacenamiento en MongoDB 3.4. Diferencia entre JSON y BSON 3.5. Estructura de los Archivos JSON 3.6. Ventajas de MongoDB en Escalabilidad y Flexibilidad 3.7. Comandos CRUD en MongoDB 3.8. Relacionar Datos en MongoDB sin JOINS 3.9. Descargar Imagen de Mongo en Docker 3.10. Herramientas para Visualizar MongoDB
4. Conclusiones
5. Bibliografía

1. Introducción

MongoDB es una base de datos NoSQL orientada a documentos que permite manejar grandes volúmenes de datos de manera flexible y escalable. A diferencia de las bases de datos relacionales como MySQL, MongoDB almacena la información en documentos BSON en lugar de tablas, lo que facilita la manipulación de datos semiestructurados y dinámicos.

2. Objetivos

- Comprender las diferencias entre MongoDB y bases de datos relacionales.
- Aprender a manejar colecciones y documentos en MongoDB.
- Explorar los formatos de almacenamiento y sus ventajas.
- Aplicar comandos CRUD en MongoDB.
- Analizar métodos para relacionar datos sin JOINS.
- Descargar y configurar MongoDB en Docker.
- Evaluar herramientas para visualizar bases de datos en MongoDB.

3. Desarrollo del Ejercicio

1. ¿Qué tipo de base de datos es MongoDB y en qué se diferencia de una base de datos relacional como MySQL?

MongoDB es una base de datos NoSQL basada en documentos, en cambio MySQL es una base de datos relacional (SQL). MongoDB guarda información en documentos BSON (parecidos a JSON), permitiendo esquemas flexibles, mientras que MySQL emplea tablas con estructuras estrictas.

MongoDB se expande mejor horizontalmente (sharding), en tanto que MySQL se centra en la escalabilidad vertical y la replicación.

En consultas, MongoDB utiliza JSON y agregaciones, mientras que MySQL usa SQL con JOINS. A pesar de que MongoDB ahora ofrece soporte para transacciones, MySQL sigue teniendo ventaja en consistencia ACID, por lo que MongoDB es más adecuado para datos semi estructurados y MySQL para relaciones complejas.

Algunas Diferencias

Aquí tienes algunas diferencias clave entre MongoDB y MySQL:

1. Modelo de Datos

- **MongoDB:** Almacena datos en documentos BSON (similar a JSON).
- **MySQL:** Usa tablas con filas y columnas.

2. Esquema

- **MongoDB:** Es flexible y no requiere un esquema fijo.
- **MySQL:** Necesita definir un esquema estructurado antes de insertar datos.

3. Consultas

- **MongoDB:** Usa consultas basadas en JSON y el modelo de agregación.
- **MySQL:** Usa SQL con JOINS y relaciones entre tablas.

4. Escalabilidad

- **MongoDB:** Escala horizontalmente mediante sharding.

- **MySQL:** Escala principalmente verticalmente y mediante replicación.

5. Transacciones

- **MongoDB:** Soporta transacciones, pero su enfoque es el rendimiento en grandes volúmenes.
- **MySQL:** Soporta transacciones ACID de forma nativa para mayor consistencia.

1. ¿Qué es una colección en MongoDB y en qué se diferencia de una tabla en SQL?

Una colección es un conjunto de documentos, equivalente a una tabla en una base de datos SQL.

En MongoDB una colección consiste en un conjunto de documentos BSON (parecido a JSON) que no necesita un esquema estático, a diferencia de una **tabla** en SQL, que organiza los datos en filas y columnas con una estructura definida. Aunque las tablas utilizan claves primarias y foráneas para establecer relaciones, las colecciones permiten documentos anidados o referencias sin requerir un esquema estricto. Esta adaptabilidad convierte a MongoDB en la opción perfecta para datos cambiantes, mientras que SQL resulta más adecuado para estructuras claramente definidas.

2. ¿Cómo se almacena la información en MongoDB y qué formato utiliza?

MongoDB almacena la información en **documentos** dentro de **colecciones**, y utiliza el formato **BSON (Binary JSON)**. BSON es una versión binaria de JSON que permite almacenar datos de manera eficiente y admite tipos adicionales, como fechas y binarios. Cada documento en MongoDB es un conjunto de pares **clave-valor**, similar a un objeto JSON, lo que permite estructuras flexibles y anidadas. Esta estructura facilita el manejo de datos semi estructurados y dinámicos sin necesidad de un esquema fijo.

3. Explica la diferencia entre JSON y BSON en MongoDB.

La principal diferencia entre JSON y BSON en MongoDB es que Bson es una versión binaria optimizada de Json, creada para aumentar el rendimiento y la eficiencia en el almacenamiento y la consulta de datos.

- **Aspectos fundamentales:**

1.Estructura y Eficiencia:

JSON: Formato textual comprensible para personas, sencillo de manejar.

BSON: Formato binario mejorado para rapidez y almacenamiento.

2.Compatibilidad con Tipos de Datos:

JSON: Admite tipos primarios como cadenas, números, valores booleanos y arreglos.

BSON: Aparte de los tipos de JSON, soporta enteros de 32 y 64 bits, fechas, datos binarios y objetos ID nativos.

3. Dimensión:

JSON: Su formato de texto puede hacerlo más pesado.

BSON: Ocupa menos espacio al comprimir ciertos tipos de información, aunque los documentos podrían ser más extensos por los metadatos extra.

En MongoDB, los datos se guardan y manejan en **BSON**, aunque las interacciones del usuario generalmente se realizan en **JSON**, lo que permite una lectura y escritura más sencilla.

4. Estructura de los archivos json

Un archivo JSON (JavaScript Object Notation) presenta una organización fundamentada en pares clave-valor, en la que las claves son cadenas de texto y los valores pueden ser diversos tipos de datos. Se emplean llaves {} para objetos y corchetes [] para listas o arreglos.

- **Ejemplo**

```
{  
  "nombre": "Finca Los Álamos",  
  "ubicacion": "Medellín",  
  "animales": ["Vacas", "Gallinas", "Cerdos"],  
  "extension": 7.5,  
  "activo": true  
}
```

Elementos clave en JSON:

1. **Objetos:** Definidos con {}, contienen pares clave-valor.
2. **Arreglos:** Definidos con [], contienen una lista de valores.
3. **Valores:** Pueden ser cadenas ("texto"), números (123), booleanos (true/false), null, objetos {} o arreglos [].

6. ¿Qué ventajas tiene MongoDB sobre una base de datos relacional en términos de escalabilidad y flexibilidad

Aspecto	Ventajas de MongoDB	Desventajas de MongoDB
Escalabilidad	Escala horizontalmente con sharding , permitiendo distribuir datos en múltiples servidores.	Las bases de datos relacionales pueden manejar mejor la consistencia en entornos distribuidos.
Flexibilidad	No requiere un esquema fijo, permitiendo estructuras dinámicas.	Puede ser más difícil de gestionar si no se establecen convenciones de estructura.
Consultas	Usa JSON y el modelo de agregación , facilitando consultas flexibles.	No tiene JOINS nativos, lo que puede dificultar algunas consultas complejas.
Rendimiento	Optimizado para lecturas y escrituras rápidas con índices eficientes.	Puede consumir más almacenamiento debido a la sobrecarga de BSON.
Soporte de Transacciones	Desde versiones recientes soporta transacciones ACID.	No es tan robusto como las bases de datos SQL en transacciones complejas.
Facilidad de Uso	Fácil de usar para desarrolladores que trabajan con JSON.	Puede ser menos intuitivo para quienes vienen de SQL.

7. Comandos para realizar CRUD en Mongo

Comandos básicos para realizar operaciones **CRUD** en **MongoDB**:

Mostrar base de datos

Este comando nos muestra todas las bases de datos que se han creado en nuestro cluster

```
show dbs
```

Crear base de datos o usar una ya existente

```
use database_name
```

Cabe aclarar que MongoDB no crea bases de datos vacías, si se quiere que la base de datos se materialice se le debe de crear un documento.

Consultar ayuda sobre algún comando

```
//Muestra todas las funciones que podemos hacer a la base de datos en uso
db.help()

//Muestra todas las funciones que le podemos hacer a una colección
db.collectionName()
```

Crear Documento dentro de una colección Esto se puede realizar a través del comando `insertOne()` en la shell de mongo

```
db.collectionName.insertOne({
  // Document in JSON format
})
```

Insertar diferentes documentos en una colección

```
db.collectionName.insertMany([
  {
    ...Document1
  },{
    ...Document2
  }])
```

Operación	Comando MongoDB	Ejemplo
Crear (Insertar)	db.coleccion.insertOne({...})	db.mascotas.insertOne({nombre: "Max", especie: "Perro", edad: 3})
	db.coleccion.insertMany([...])	db.mascotas.insertMany([{nombre: "Luna", especie: "Gato"}, {nombre: "Rocky", especie: "Perro"}])
Leer (Consultar)	db.coleccion.find({})	db.mascotas.find({}) → Muestra todas las mascotas
	db.coleccion.findOne({filtro})	db.mascotas.findOne({nombre: "Max"})
Actualizar	db.coleccion.updateOne({filtro}, {\$set: {campo: valor}})	db.mascotas.updateOne({nombre: "Max"}, {\$set: {edad: 4}})
	db.coleccion.updateMany({filtro}, {\$set: {campo: valor}})	db.mascotas.updateMany({especie: "Perro"}, {\$set: {vacunado: true}})
Eliminar	db.coleccion.deleteOne({filtro})	db.mascotas.deleteOne({nombre: "Rocky"})
	db.coleccion.deleteMany({filtro})	db.mascotas.deleteMany({especie: "Gato"})

8. Cómo se pueden relacionar datos en Mongo sin usar joins como en sql

En MongoDB, los datos pueden asociarse a través de documentos anidados o mediante referencias entre ellos. Los documentos anidados guardan la información conectada dentro del mismo archivo, perfecto para datos que se revisan de manera conjunta:

Ejemplo Json

```
{
  "nombre": "Carlos Pérez",
  "mascotas": [{"nombre": "Max", "especie": "Perro"}]
}
```

Las **referencias entre documentos** utilizan un **ObjectId** para vincular documentos en diferentes colecciones, útil para relaciones más complejas:

```
// Dueño
{ "_id": ObjectId("123abc"), "nombre": "Carlos Pérez" }

// Mascota
{ "nombre": "Max", "especie": "Perro", "dueño_id": ObjectId("123abc") }
```

Esto permite evitar duplicación de datos y facilita la escalabilidad

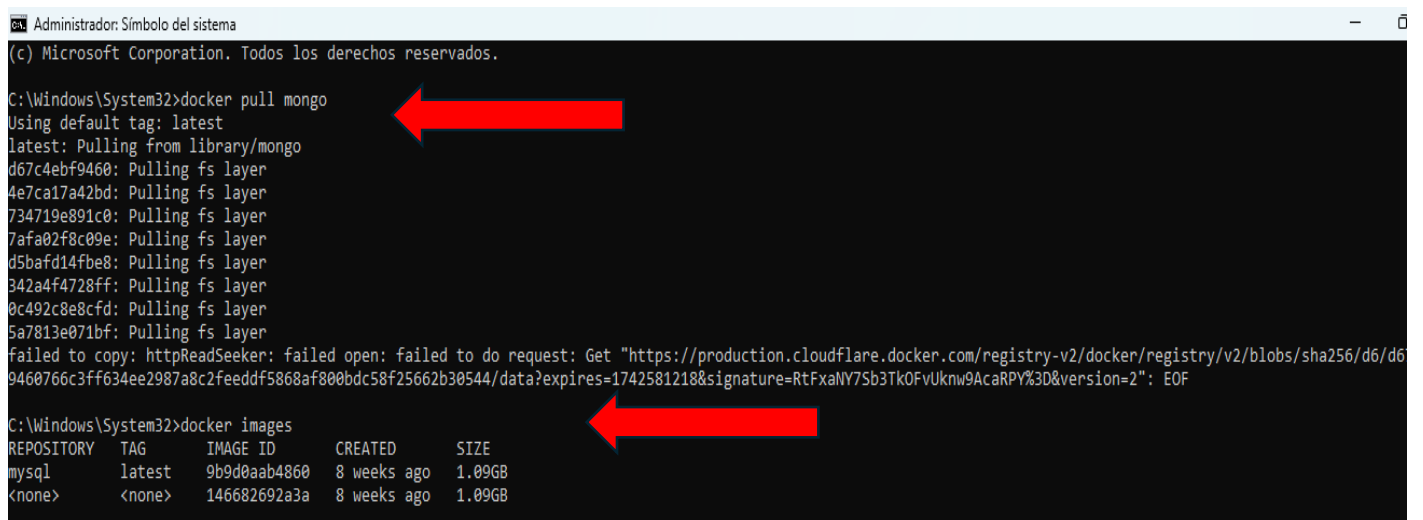
9. Descargar imagen de mongo en Docker

1. Descargar la imagen de MongoDB

Se ejecuta el siguiente comando en la terminal: `docker pull mongo`

2. Verificar la imagen descargada

Después de la descarga, verifica que la imagen está disponible con: `docker images`



```
Administrador: Símbolo del sistema
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32>docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
d67c4ebf9460: Pulling fs layer
4e7ca17a42bd: Pulling fs layer
734719e891c0: Pulling fs layer
7afa02f8c09e: Pulling fs layer
d5bafd14fbe8: Pulling fs layer
342a4f4728ff: Pulling fs layer
0c492c8e8cfd: Pulling fs layer
5a7813e071bf: Pulling fs layer
failed to copy: httpReadSeeker: failed open: failed to do request: Get "https://production.cloudflare.docker.com/registry-v2/docker/registry/v2/blobs/sha256/d6/d69460766c3ff634ee2987a8c2feeddf5868af800bdc58f25662b30544/data?expires=1742581218&signature=RtFxaNY7Sb3TkOFvUknw9AcaRPY%3D&version=2": EOF

C:\Windows\System32>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mysql          latest    9b9d0aab4860   8 weeks ago   1.09GB
<none>         <none>    14682692a3a   8 weeks ago   1.09GB
```

10. Herramientas similares a Workbench para visualizar los datos de mongo.

Herramienta	Características	Nivel de usuario
MongoDB Compass	Oficial, interfaz gráfica simple	Básico - Intermedio
Studio 3T	SQL-to-Mongo, consultas visuales	Avanzado
Robo 3T	Ligero, ejecuta shell de MongoDB	Básico
NoSQLBooster	Autocompletado, consultas SQL-like	Intermedio - Avanzado
DBeaver	Soporta múltiples bases de datos	Intermedio
NoSQL Manager	UI avanzada, monitoreo	Intermedio - Avanzado
TablePlus	Minimalista y rápido	Básico - Intermedio
DataGrip (JetBrains)	IDE profesional con soporte Mongo	Avanzado
Navicat for MongoDB	Administración, respaldo de datos	Avanzado
Humongous.io	Plataforma en la nube	Básico

MongoDB proporciona varias herramientas para la visualización y administración de sus bases de datos, que van desde opciones oficiales como MongoDB Compass hasta soluciones más sofisticadas como Studio 3T y DataGrip. La elección se basa en el grado de experiencia y requerimientos del usuario: Compass y Robo 3T son perfectos para quienes recién comienzan, mientras que Studio 3T, NoSQLBooster y DBeaver brindan características más sofisticadas para programadores y administradores de bases de datos.

4. Conclusiones

MongoDB es una base de datos NoSQL ideal para datos dinámicos y escalabilidad horizontal. Su flexibilidad, velocidad y compatibilidad con JSON/BSON lo hacen una excelente opción para proyectos modernos, aunque su falta de estructura estricta puede requerir buenas prácticas en su diseño y gestión.

Referencias

- Amazon Web Services. (s.f.). *Diferencias entre MongoDB y MySQL*. AWS. Recuperado el 21 de marzo de 2025, de <https://aws.amazon.com/es/compare/the-difference-between-mongodb-vs-mysql/>
- Astera. (s.f.). *MongoDB vs. SQL Server*. Astera. Recuperado el 21 de marzo de 2025, de <https://www.astera.com/es/type/blog/mongodb-vs-sql-server/>
- Pure Storage. (s.f.). *¿Qué es MongoDB?*. Pure Storage. Recuperado el 21 de marzo de 2025, de <https://www.purestorage.com/es/knowledge/what-is-mongodb.html>
- Couchbase. (s.f.). *Conceptos: JSON vs. BSON*. Couchbase. Recuperado el 21 de marzo de 2025, de <https://www.couchbase.com/es/resources/concepts/json-vs-bson/>
- Arsys. (s.f.). *Formato JSON: ¿qué es y para qué sirve?*. Arsys. Recuperado el 21 de marzo de 2025, de <https://www.arsys.es/blog/formato-json-que-es-y-para-que-sirve>
- Amazon Web Services. (s.f.). *MongoDB y MySQL: Comparación de rendimiento y escalabilidad*. AWS. Recuperado el 21 de marzo de 2025, de <https://aws.amazon.com/es/compare/the-difference-between-mongodb-vs-mysql/>
- Platzi. (s.f.). *Cosas básicas de un CRUD en MongoDB*. Platzi. Recuperado el 21 de marzo de 2025, de <https://platzi.com/tutoriales/1533-mongodb-basico/4102-cosas-basicas-de-un-crud-en-mongodb/>
- Stack Overflow en español. (s.f.). *Relaciones entre colecciones en MongoDB*. Stack Overflow. Recuperado el 21 de marzo de 2025, de <https://es.stackoverflow.com/questions/249344/relaciones-entre-collections-mongo>
- ClickUp. (s.f.). *Alternativas a MongoDB*. ClickUp. Recuperado el 21 de marzo de 2025, de <https://clickup.com/es-ES/blog/116518/alternativas-a-mongodb>