

## TALLER 1

Javier Adrian Pedraza Garcia

Ingeniería de sistemas, Corporación Universitaria Minuto de Dios

60747: Bases d Datos Masivas

Prof. Alexander Matallana

21/032025

INTRODUCCION .....	3
OBJETIVO.....	3
TALLER 1 .....	3
1. Qué tipo de base de datos es MongoDB y en qué se diferencia de una base de datos relacional como MySQL.....	3
2. ¿Qué es una colección en MongoDB y en qué se diferencia de una tabla en SQL? .....	4
3. ¿Cómo se almacena la información en MongoDB y qué formato utiliza? .....	5
4. Explica la diferencia entre JSON y BSON en MongoDB. ....	6
5. Estructura de los archivos json .....	6
6. ¿Qué ventajas tiene MongoDB sobre una base de datos relacional en términos de escalabilidad y flexibilidad .....	6
7. Comandos para realizar CRUD en Mongo .....	7
8. Cómo se pueden relacionar datos en Mongo sin usar joins como en sql.....	8
9. Descargar imagen de mongo en Docker .....	9
10. Herramientas similares a Workbench para visualizar los datos de mongo.....	10
CONCLUSION .....	11
REFERENCIAS.....	11

## **INTRODUCCION**

MongoDB es una herramienta para la creación de base de datos NoSQL de las mas populares, en este trabajo se encontrarán conceptos, que es mongo diferencias entre mongo y MySQL, que son las colecciones y documentos, como funciona el almacenamiento de las bases de datos en Mongo, en que formato se basa Mongo, los comando que se utilizan para generar un CRUD en Mongo, si existen las JOIN o si no cual es su correspondiente función y el como descargar Mongo en Docker.

## **OBJETIVO**

Este taller tiene el objetivo de dará entender las características principales de Mongo, su estructura, formato, su manera de almacenar y como se diferencian las bases de datos creadas en Mongo a las bases de datos creadas en mySQL

## **TALLER 1**

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

Las bases de datos Mongo son un sistema de gestión de BD no orientadas hacia SQL, estás BD están orientadas a los documentos y son de codigo abierto. Estás bases de datos en lugar de almacenar mediante filas y columnas utiliza documentos flexibles en un formato llamado BSON que son similares a los JSON, los que permite a estás bases de datos manejar datos no estructurado d manera más eficiente

Las características principales de Mongo son:

- La utilización de esquemas flexibles: Esto se refiere a que las bases de datos Mongo permiten el manejo de documentos dentro de una colección y que estos tengan una gran adaptación a cambios
- Escalabilidad horizontal: Esto se refiere a que las bases de datos Mongo estan diseñadas para escalar, distribuyendo datos en múltiples servidores para poder grandes cantidades de información
- Consultas avanzadas: Las MongoDB soportan búsquedas por campo, consultas, rango y expresiones regulares
- Alta disponibilidad: Las MongoDB ofrecen una replicación mediante conjuntos d réplicas, asegurando una mayor disponibilidad de los datos

Las principales diferencias entre Mongo y MySQL son

- El modelo de datos
- Lenguajes de consulta
- Esquemas y flexibilidad
- Escalabilidad
- Transacciones y consistencia

Mongo es una base de datos NoSQL que está orientado hacia documentos por otro lado las bases de datos de MySQL son bases de datos relacionales

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

En MongoDB las colecciones son un conjunto de documentos que están agrupados en una base de datos y cada documento es una estructura de datos en el formato BSON

Las diferencias entre una tabla y una colección son:

- Esquemas flexibles: La diferencia entre las tablas que se utilizan en SQL es que dónde las tablas tienen que seguir el orden de las columnas preestablecidas Mongo y sus colecciones no imponen un orden en específico permitiendo que los documentos dentro de las colecciones puedan tener diferentes estructuras
- Estructura de los datos: Mientras que las tablas en SQL almacenan la información en filas y columnas, las colecciones en Mongo almacenan documentos completos que permiten contener datos anidados y complejos

### **3. ¿Cómo se almacena la información en MongoDB y qué formato utiliza?**

Cómo mencioné en el primer punto Mongo se maneja en el formato BSON y este formato es una extensión del formato JSON que permite mostrar de manera más eficiente los datos incluyendo cadenas de texto, números, fechas, booleanos, arreglos, etc.

El orden de cómo se manejan este tipo de datos es:

- Crear la base de datos (si se maneja en docker es similar) y usarla y en esta base de datos se crean los documentos que se tienen que poner en una colección
- Lectura de los documentos, para esto lo primero es consultar todos los documentos que tenemos en una colección mediante `db.usuarios.find()`, también se pueden buscar documentos en específico mediante `db.usuarios.findOne({nombre: "Juan"})`
- Actualización de documentos, este proceso se puede realizar utilizando comandos como `(db.usuario.updateMany({ciudad: "Bogotá"},{$set: {país: "colombia"}})`

- Eliminación de documentos, esto se puede hacer utilizando el comando  
`"d.b.usuarios.delete.One({ nombre: "Juan" })`

#### **4. Explica la diferencia entre JSON y BSON en MongoDB.**

La diferencia es que JSON es un formato de texto que está basado en Javascript, que normalmente se utiliza para almacenar o transmitir datos y los BSON son un formato binario derivado del mismo JSON, que está optimizado para almacenar y procesar en MongoDB. La mayor diferencia es que BSON soporta más tipos de datos y es más rápido para la lectura y escritura en las bases de datos

#### **5. Estructura de los archivos json**

Los JSON (JavaScrip Objeto Notation) Son un formato de texto hecho para el intercambio de datos. Esta basado en la estructura de objetos de JavaScrip y es utilizado normalmente para almacenar y transmitir información entre sistemas

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

- Mongo permite distribuir datos en múltiples servidores mientras que las bases de datos relacionales dependen de una escalabilidad vertical
- Mongo permite realizar réplicas automáticamente para enviar datos a múltiples servidores mientras que en SQL la replicación es más compleja y requiere configurar varias cosas

- Mongo no requiere una estructura fija como lo son las tablas y columnas lo que permiten almacenar documentos con diferentes formatos
- Mongo permite documentos anidados y estructuras jerárquicas lo que permite reducir la necesidad de utilizar varios inner JOIN

## 7. Comandos para realizar CRUD en Mongo

### CREATE

Insertar documentos en una colección:

#### - INSERTAR UN DOCUMENTO

`db.usuarios.insertOne({ nombre: "Juan", edad: 25, correo: juan@gmail.com });` "insertar uno"

`db.usuarios.insertOne([  
{ nombre: "Juan", edad: 25, correo: juan@gmail.com },  
{ nombre: "Juan2", edad: 25, correo: juan2@gmail.com }  
]);` "insertar varios"

### LEER

Consultar documentos de una colección

`Db.usuarios.find();` "obtener todos los documentos"

`Db.usuarios.find({edad:25});` "filtrar documentos"

`Db.usuarios.find({edad:25}).pretty();` "filtrar y mostrar el formateado"

`Db.usuarios.findOne({nombre: "juan" });` "obtener un solo documento"

### ACTUALIZAR

`Db.usuarios.updateMany(  
{nombre:: "juan"}},  
{$set: { edad:20 }  
});` "un documento"

`Db.usuarios.updateMany(  
{edad: {$it: 30}},  
{$set: { activo: true}}  
);` "varios documentos"

## DELETE

Db.usuarios.deleteOne({ nombre: "juan"}); "un solo documento"

Db.usuarios.deleteMany({ edad: { \$gt: 50 } }); "Varios documentos"

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

Ya que en Mongo no existen los joins como en SQL se pueden establecer relaciones entre datos de dos formas

### - INCRUSTADO

Se puede almacenar toda la información relacionada de un documento, el comando seria:

```
db.usuarios.insertOne({
  nombre: "Juan",
  edad: 25,
  direcciones: [
    { ciudad: "Ciudad de México", calle: "Av. Reforma 123" },
    { ciudad: "Guadalajara", calle: "Calle Juárez 456" }
  ]
});
```

### - REFERENCIAS

Se almacena el id de otro documento en lugar de incrustar los datos y el comando seria:

```
db.usuarios.insertOne({
  _id: ObjectId("65a7b7e8d5f1d50c0a0a1234"),
  nombre: "Juan",
  edad: 25
});
```

```
db.pedidos.insertOne({
  usuario_id: ObjectId("65a7b7e8d5f1d50c0a0a1234"),
  producto: "Laptop",
  precio: 1200
});
```

### - EN LUGAR DE LOS JOIN PARA OBTENER LA INFORMACION DE AMBOS SE HACEN DOS CONSULTAS

```
const usuario = db.usuarios.findOne({ nombre: "Juan" });
const pedidos = db.pedidos.find({ usuario_id: usuario._id });
```

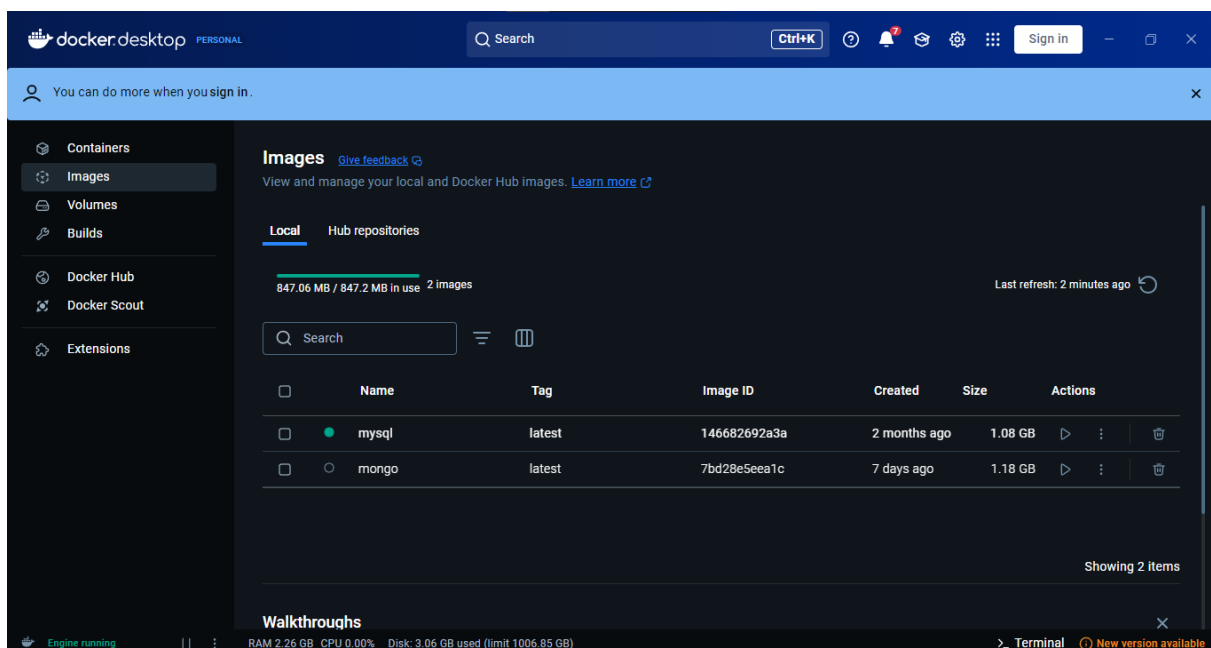
### - REFERENCIAS CON \$lookup



Mongo tiene una herramienta llamada lookup que es equivalente a un JOIN y el comando de esto es:

```
db.usuarios.aggregate([
  {
    $lookup: {
      from: "pedidos",
      localField: "_id",
      foreignField: "usuario_id",
      as: "pedidos"
    }
  }
]);
```

## 9. Descargar imagen de mongo en Docker



```
Simbolo del sistema
C:\Users\janpe>docker pull mongo
Using default tag: latest
error during connect: Post "http://%2F%2F.%2Fpipe%2FdockerDesktopLinuxEngine/v1.47/images/create?fromImage=mongo&tag=latest": open //./pipe/dockerDesktopLinuxEngine: The system cannot find the file specified.

C:\Users\janpe>docker pull mongo: latest
"docker pull" requires exactly 1 argument.
See 'docker pull --help'.

Usage:  docker pull [OPTIONS] NAME[:TAG|@DIGEST]

Download an image from a registry

C:\Users\janpe>docker pull mongo:latest
latest: Pulling from library/mongo
734719e891c0: Download complete
342a4f4728ff: Download complete
7afa02f8c09e: Download complete
4e7ca17a42bd: Download complete
0c492c8e8cfd: Download complete
d67c4ebf9460: Download complete
d5bafdd14f8e8: Download complete
5a7813e071bf: Download complete
Digest: sha256:7bd28e5eea1c5766a084d5818254046f3ebe3b8f20a65e3a274640189e296667
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest

C:\Users\janpe>
C:\Users\janpe>
```

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

**DBraver** - Herramienta gratuita y ligera para consultar MongoDB y otras bases de datos SQL/NoSQL.

**MongoDB Compass** - Aplicación oficial de MongoDB con una interfaz intuitiva para explorar, consultar y analizar datos sin necesidad de escribir código.

**Studio 3T** - Plataforma avanzada con autocompletado de consultas, agregaciones visuales y migración de datos.

**NoSQLBooster** - Soporta JavaScript en consultas y permite trabajar con scripts personalizados.

**Robo 3T (antes RoboMongo)** - Cliente ligero y gratuito con soporte para Shell de MongoDB.

**TablePlus** - Administra múltiples bases de datos, incluida MongoDB, con una interfaz limpia y rápida.

**DataGrip** - Herramienta de JetBrains que permite administrar varias bases de datos, incluidas NoSQL como MongoDB.

## CONCLUSION

Mongo es una herramienta que es parecida y al mismo tiempo diferente a my SQL a pesar de que ambas sirven para crear bases de datos tienen formatos y una manera de estructurarlas muy diferentes además de que mongo tiene la capacidad de almacenar cantidades de datos mayores a las de mySQL

## REFERENCIAS

<https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>

JSON.org. (s.f.). Introducing JSON. Recuperado de <https://www.json.org/json-en.html>

MongoDB, Inc. (s.f.). MongoDB Architecture Guide. Recuperado de <https://www.mongodb.com/architecture>

Harrison, G. (2015). Next Generation Databases: NoSQL, NewSQL, and Big Data. Apress.

MongoDB, Inc. (s.f.). CRUD Operations. Recuperado de <https://docs.mongodb.com/manual/crud/>

Chodorow, K. (2013). MongoDB: The Definitive Guide (2ª ed.). O'Reilly Media.

<https://chatgpt.com/share/67ddbd22-a16c-800e-a858-add63bfbee74>