

Taller MongoDB

Cristian Camilo Tique Tapiero

Ingeniería de sistemas, Corporación Universitaria Minuto de Dios

60747: Bases de datos masivas

Prof. William Alexander Matallana

21 de marzo de 2025

Tabla de Contenido

Introducción -----	1
Objetivos -----	1
Desarrollo del Taller -----	2
¿Qué tipo de base de datos es MongoDB y en qué se diferencia de una base de datos relacional como MySQL? -----	2
¿Qué es una colección en MongoDB y en qué se diferencia de una tabla en SQL? -----	2
¿Cómo se almacena la información en MongoDB y qué formato utiliza? -----	3
Explica la diferencia entre JSON y BSON en MongoDB -----	4
Estructura de los archivos JSON -----	4
¿Qué ventajas tiene MongoDB sobre una base de datos relacional en términos de escalabilidad y flexibilidad? -----	5
Comandos para realizar CRUD en Mongo -----	6
Cómo se pueden relacionar datos en Mongo sin usar joins como en SQL-----	6
Descargar imagen de Mongo en Docker -----	7
Herramientas similares a Workbench para visualizar los datos de Mongo -----	8
Referencias -----	9

Introducción

MongoDB es una base de datos que no es de SQL la cual permite almacenar datos en un formato flexible y escalable. A diferencia de las bases de datos relacionales, MongoDB usa documentos en lugar de tablas, lo que facilita la manipulación de datos sin una estructura fija.

Objetivos

- Comprender la estructura y funcionamiento de MongoDB.
- Identificar sus diferencias con bases de datos relacionales.
- Aplicar comandos CRUD en MongoDB.
- Explorar herramientas para visualizar datos en MongoDB.

Desarrollo del taller

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 no SQL, lo que significa que no usa tablas ni relaciones fijas como las bases de datos relacionales ya conocidas; mas bien, almacena la información en documentos flexibles dentro de colecciones.

La principal diferencia con MySQL es que MySQL organiza los datos en tablas con filas y columnas, mientras que MongoDB usa documentos en formato JSON o BSON, lo que permite mayor flexibilidad, ya que no es necesario definir un esquema estricto antes de almacenar los datos.

Otra gran diferencia es la forma en que manejan el crecimiento de los datos. MongoDB se escala horizontalmente, es decir, se pueden distribuir los datos en varios servidores sin perder rendimiento, mientras que en MySQL se escala principalmente de forma vertical, aumentando la capacidad de un solo servidor.

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

En MongoDB, una colección es el equivalente a una tabla en una base de datos relacional. Sin embargo, la diferencia principal es que una colección no requiere una estructura fija; es decir, en MySQL, todas las filas de una tabla deben seguir la misma estructura con los mismos tipos de datos; en cambio, en MongoDB, cada documento dentro de una colección puede tener diferentes campos y estructuras sin problemas.

Un ejemplo, en MySQL, si una tabla de clientes tiene las columnas “nombre”, “apellido” y “edad”, todas las filas deben cumplir con esa estructura. En MongoDB, en una colección de clientes, un documento podría tener “nombre” y “edad”, mientras que otro podría tener “nombre” y “dirección”, sin que esto cause errores.

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

Mongo almacena la información en documentos que utilizan el formato BSON (Binary JSON), BSON es una versión optimizada de JSON, lo que mantiene la estructura de clave-valor típica de JSON pero con una codificación binaria que mejora la velocidad y el almacenamiento. Cada documento en MongoDB se parece a un objeto JSON, por ejemplo:

```
{  
  
  "nombre": "Camilo",  
  
  "edad": 21,  
  
  "correo": "camilo12@gmail.com"  
}
```

Estos documentos se agrupan en colecciones, que funcionan como las tablas en bases de datos relacionales.

4. Explica la diferencia entre JSON y BSON en MongoDB

JSON (JavaScript Object Notation) es un formato de texto ligero y fácil de leer para representar datos estructurados. Por el contrario, BSON (Binary JSON) es una versión binaria de JSON que MongoDB usa internamente. La principal diferencia es que BSON es más eficiente para la manipulación de datos dentro de MongoDB porque permite una lectura y escritura más rápidas. Además, BSON soporta tipos de datos adicionales como fechas y datos binarios, mientras que JSON se limita a texto y números. Por lo que , mongoDB usa BSON debido a que es más rápido y permite almacenar más tipos de datos que JSON. Sin embargo, cuando se extraen datos de MongoDB, se pueden convertir a JSON fácilmente para que sean más legibles.

5. Estructura de los archivos JSON.

Un archivo JSON está basado en pares de clave-valor y sigue una estructura jerárquica con llaves { } y corchetes [] para representar objetos y listas.

Ejemplo de un archivo JSON:

```
{  
  
  "nombre": "Arellys",  
  
  "edad": 20,  
  
  "correo": "aris11@email.com",  
  
  "direccion": {  
  
    "ciudad": "Cajica",
```

```
"Barrio": "Canelon"

},

"telefono": ["3117654321"]

}
```

En este caso, nombre, edad y correo son claves con valores simples, mientras que direccion es un objeto dentro del JSON y telefono es una lista de valores.

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

Escalabilidad horizontal: MongoDB permite distribuir los datos en múltiples servidores (sharding), lo que facilita manejar grandes volúmenes de información sin afectar el rendimiento.

Estructura flexible: Aquí es necesario definir un esquema fijo, por lo que cada documento puede tener distintos campos sin problemas. Esto facilita la adaptación de la base de datos en caso de cambios en los requisitos del proyecto.

Mayor velocidad en operaciones (lectura y escritura): Al no usar relaciones complejas entre tablas, MongoDB accede y almacena datos de manera más rápida en comparación con una base de datos relacional.

7. Comandos para realizar CRUD en Mongo.

CRUD son las operaciones básicas en bases de datos: Crear, Leer, Actualizar y Eliminar. En MongoDB, estos son los comandos principales:

Crear (Insertar un documento):

```
Db.usuarios.insertOne({nombre: "Ana", edad: 30})
```

Leer (Obtener documentos de una colección):

```
Db.usuarios.find()
```

Actualizar (Modificar datos de un documento):

```
Db.usuarios.updateOne({nombre: "Ana"}, {$set: {edad: 31}})
```

Eliminar (Borrar un documento):

```
Db.usuarios.deleteOne({nombre: "Ana"})
```

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

En SQL, las relaciones entre tablas se hacen con JOINS. En MongoDB, hay dos formas principales de relacionar datos sin usar JOINS:

1. Referencias a documentos

Se usa un campo en un documento para almacenar el ID de otro documento.

Luego, se hace una consulta para obtener los datos de la ciudad usando el `ciudad_id`.

2. Incrustar documentos

Se almacenan datos relacionados dentro del mismo documento. Esto evita hacer varias consultas y mejora el rendimiento.

9. Descargar imagen de Mongo en Docker.

El comando para descargar e MongoDB en Docker es:

```
docker pull mongo
```

1. Ejecuto cmd
2. Ejecuto el comando

```
Microsoft Windows [Versión 10.0.22631.4890]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\trcri>docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
734719e891c0: Download complete
4e7ca17a42bd: Download complete
d67c4ebf9460: Download complete
5a7813e071bf: Download complete
342a4f4728ff: Download complete
7afa02f8c09e: Download complete
d5bafd14fbe8: Download complete
0c492c8e8cfd: Download complete
Digest: sha256:7bd28e5eealc5766a084d5818254046f3ebe3b8f20a65e3a274640189e296667
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest

C:\Users\trcri>
```

3. Ejecuto el comando `Docker images` para visualizar mis imágenes

```
C:\Users\trcri>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mongo         latest    7bd28e5eea1c   7 days ago    1.19GB
mysql         latest    146682692a3a   8 weeks ago    1.09GB

C:\Users\trcri>
```

- Si voy a Docker, deberá de aparecer la imagen

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	mysql	latest	146682692a3a	2 months ago	1.08 GB	
<input type="checkbox"/>	mongo	latest	7bd28e5eea1c	7 days ago	1.18 GB	

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

Para administrar y visualizar datos en MongoDB, se pueden usar herramientas como:

- **MongoDB Compass** = Herramienta oficial de MongoDB con interfaz gráfica.
- **Robo 3T** = Cliente ligero para trabajar con bases de datos MongoDB.
- **NoSQLBooster** = Interfaz avanzada con soporte para consultas en JavaScript.

Estas herramientas facilitan la administración de MongoDB sin necesidad de usar solo la línea de comandos.

Referencias

Aws (n/a). ¿Cuál es la diferencia entre MongoDB y MySQL?. Aws.

<https://aws.amazon.com/es/compare/the-difference-between-mongodb-vs-mysql/>

Stackoverflow(2019). ¿Diferencia entre una tabla (SQL) y una colección (Mongo)?.

Stackoverflow. <https://stackoverflow.com/questions/38581139/difference-between-a-table-sql-and-a-collection-mongo>

mongoDB (n/a). Qué es MongoDB?. mongoDB.

<https://www.mongodb.com/es/company/what-is-mongodb>

Jeffrey Erickson (2024). ¿Qué es MongoDB? Guía experta. Oracle.

<https://www.oracle.com/mx/database/mongodb/>

mongoDB (n/a). Operaciones CRUD de MongoDB. mongoDB.

<https://www.mongodb.com/docs/manual/crud/>

estibadorcentro. (s.f.). *mongo*. Recuperado el 21 de 03 de 2025, de hub.docker:

https://hub.docker.com/_/mongo

Gayatri. (18 de 11 de 2024). *dronahq*. Recuperado el 21 de 03 de 2025, de Top

MongoDB GUI tools you can explore in 2025: <https://www.dronahq.com/top-mongodb-guis/>