

Taller MongoDB

Juan Sebastian Dueñas Robayo

833539

Corporación Universitaria Minuto de Dios

Ing. William Alexander Matallana Porras

10-60747: Bases de Datos masivas

2025

Contenido

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

Introducción

MongoDB es una base de datos NoSQL muy empleada en la creación de aplicaciones modernas gracias a su versatilidad y capacidad de expansión. En comparación a bases de datos relacionales como MySQL, MongoDB guarda información en documentos JSON en vez de tablas estructuradas, lo que facilita una gestión más eficiente de grandes cantidades de datos. En este taller, analizaremos sus propiedades fundamentales, organización de datos, órdenes CRUD y otras funciones clave.

Objetivos

- Comprender las comparaciones entre MongoDB y una base relacional de datos
- Examinar la configuración de almacenamiento de MongoDB y su formato de información.
- Comprender los beneficios de MongoDB en cuanto a escalabilidad y adaptabilidad.
- Investigar los comandos CRUD fundamentales en MongoDB.
- Comprender cómo vincular datos en MongoDB sin recurrir a joins.
- Establecer una imagen de MongoDB en Docker.
- Identificar herramientas para la visualización de datos en MongoDB.

DESARROLLO DEL EJERCICIO

¿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 orientada a documentos que apareció a mediados de la década de 2000. Se utiliza para almacenar volúmenes masivos de datos.

A diferencia de una base de datos relacional SQL tradicional, MongoDB no se basa en tablas y columnas. Los datos se almacenan como colecciones y documentos.

Los documentos son pares value/key que sirven como unidad básica de datos. Las colecciones contienen conjuntos de documentos y funciones. Son el equivalente a las tablas en las bases de datos relacionales clásicas.

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

Las Colecciones son agrupaciones de documentos. Son equivalentes a las tablas en bases de datos relacionales, pero NO nos imponen un esquema o estructura rígida para guardar información. En SQL, una tabla tiene un esquema rígido con columnas y tipos de datos predefinidos.

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

MongoDB almacena objetos de datos en colecciones y documentos en lugar de las tablas y filas que se utilizan en las bases de datos relacionales tradicionales. Las colecciones comprenden conjuntos de documentos, que son equivalentes a tablas en una base de datos

relacional. Los documentos consisten en pares clave-valor, que son la unidad básica de datos en MongoDB.

Se puede modificar la estructura de un documento simplemente añadiendo campos nuevos o suprimiendo los ya existentes. Los documentos pueden establecer un identificador principal como clave única y los valores pueden ser diversos tipos de datos, incluyendo otros documentos, matrices y matrices de documentos. Usa formato **BSON** que es una forma binaria de **JSON** que permite acomodar más tipos de datos.

[Explica la diferencia entre JSON y BSON en MongoDB.](#)

- Binario en comparación con texto: BSON es un formato binario de codificación, en cambio, JSON es un formato que se fundamenta en texto. Esto implica que BSON es compacto para ser transmitido en una red, en cambio, JSON es comprensible para los humanos y más sencillo de manejar en diferentes contextos.
- Soporte de datos ampliados: JSON se restringe a los tipos de datos utilizados en JavaScript, tales como cadenas, números, booleanos, nulos, objetos y grupos. Estos tipos de información pueden fusionarse para simbolizar tipos de datos complejos. BSON soporta tipos adicionales de datos (como datos binarios y tipos de fecha) que JSON no soporta.
- Con el apoyo de: BSON sólo está soportado de forma nativa por MongoDB. JSON, por otro lado, está ampliamente soportado y se puede utilizar con sistemas de bases de datos distribuidas lenguajes de programación y plataformas.

- **Huella:** En ciertas circunstancias, los documentos BSON pueden tener un tamaño superior al de los documentos JSON equivalentes debido a que contienen metadatos extra e información que no se encuentra en estos mismos. Esto puede influir en los plazos de transmisión y en las necesidades de almacenamiento, particularmente en situaciones de grandes volúmenes de datos. BSON y JSON pueden obtener beneficios de la compresión.
- **Compatibilidad y complejidad:** BSON es más sofisticado que JSON, lo que complica su manejo en ciertos contextos. Los programadores pueden necesitar adquirir nuevos tipos de datos y técnicas de codificación/decodificación para manejar eficientemente BSON. Además, pueden presentarse dificultades de compatibilidad cuando se comparten datos entre sistemas que no soportan totalmente los tipos adicionales de BSON.

Estructura de los archivos json

```
{  
  'nombre': María,  
  'edad': 30,  
  'ciudad': 'Madrid'  
}
```

Este objeto JSON consta de 3 pares clave / valor:

La clave <<nombre>> tiene el valor de <<María>>

La clave <<edad>> tiene el valor de <<30>>

La clave <<ciudad>> tiene el valor de <<Madrid>>

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

- Escalabilidad horizontal mediante sharding.
- Modelo de datos flexible y sin esquema fijo.
- Mejor rendimiento en lectura y escritura.
- Facilidad para manejar datos semiestructurados y anidados.

Comandos para realizar CRUD en Mongo

- Create (Insertar un documento)

```
db.insertOne({"nombre": "Juan", "edad": "30"})
```

- Read (Leer documentos)

```
db.usuarios.find({})
```

- Update (Actualizar documentos)

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

- Delete (Eliminar documentos)

```
db.usuarios.deleteOne({"nombre": "Juan"})
```

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

Operador Lookup

Mongo nos provee una herramienta para realizar relaciones entre documentos, este es el operador `$lookup`. Este tiene como objetivo fusionar información de un documento a otro, utilizando el metodo de combinación externa izquierda, cabe mencionar que este operador se utiliza como tubería dentro del método `aggregate()`.

Ejemplo del operador lookup

```
{
  $lookup: {
    from: <collection_to_join>,
    localField: <field_from_the_input_documents>,
    foreignField: <field_from_collection>,
    as: <output_array_field>
  }
}
```

Descargar imagen de Mongo en Docker

```
C:\Windows\System32>docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
d67c4ebf9460: Pull complete
7afa02f8c09e: Pull complete
734719e891c0: Pull complete
4e7ca17a42bd: Pull complete
342a4f4728ff: Pull complete
d5bafd14fbe8: Pull complete
0c492c8e8cfd: Pull complete
5a7813e071bf: Pull complete
Digest: sha256:7bd28e5eea1c5766a084d5818254046f3ebe3b8f20a65e3a274640189e296667
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest

C:\Windows\System32>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mongo         latest    7bd28e5eea1c   7 days ago    1.19GB
mysql         latest    146682692a3a   8 weeks ago    1.09GB
```


Herramientas similares a Workbench para visualizar los datos de mongo.

- MongoDB Compass
- Robo 3T (Robomongo)
- Studio 3T

Conclusiones

MongoDB se presenta como una alternativa robusta a las bases de datos relacionales, brindando flexibilidad, escalabilidad y un modelo de datos adaptable a necesidades dinámicas. Su estructura basada en documentos y la capacidad de escalabilidad horizontal lo hacen una excelente opción para proyectos modernos. Además, su integración con herramientas como Docker y MongoDB Compass facilita su implementación y gestión.

Bibliografía

- DataScientest. (s.f.). MongoDB: todo sobre la base de datos NoSQL orientada a documentos. Recuperado de <https://datascientest.com/es/mongodb-todo-sobre-la-base-de-datos-nosql-orientada-a-documentos>
- Platzi. (s.f.). Bases de datos, colecciones y documentos en MongoDB. Recuperado de <https://platzi.com/clases/1533-mongodb-basico/18481-bases-de-datos-colecciones-y-documentos-en-mongodb/>
- Pure Storage. (s.f.). ¿Cómo funciona MongoDB?. Recuperado de <https://www.purestorage.com/es/knowledge/what-is-mongodb.html>

- Couchbase. (s.f.). JSON vs BSON. Recuperado de <https://www.couchbase.com/es/resources/concepts/json-vs-bson/>
- Arsys. (s.f.). Formato JSON: ¿qué es y para qué sirve?. Recuperado de <https://www.arsys.es/blog/formato-json-que-es-y-para-que-sirve>
- Codehoven. (s.f.). MongoDB: Update, Insert, Find, Delete. Recuperado de <https://www.codehoven.com/mongodb-update-insert-find-delete>
- Docker Hub. (s.f.). MongoDB en Docker. Recuperado de https://hub.docker.com/_/mongo
- MongoDB. (s.f.). MongoDB Compass. Recuperado de <https://www.mongodb.com/products/tools/compass>
- Robo 3T. (s.f.). Robo 3T (Robomongo). Recuperado de <https://robomongo.org/>
- Studio 3T. (s.f.). Studio 3T para MongoDB. Recuperado de <https://studio3t.com/es/>