



TALLER MONGO

Laura Jhuliana Velasco Gómez

Ingeniería de sistemas, Corporación Universitaria Minuto de Dios,
Sede Cundinamarca, C.R. Zipaquirá

NRC: 60747 Bases de Datos Masivas

Ing. William Matallana Porras

21 marzo 2025

Contenido

INTRODUCCIÓN.....	3
TIPO DE DATO MONGO	3
DIFERENCIA DE UNA BASE DE DATOS RELACIONAL COMO MySQL.....	3
COLECCIÓN DE MONGO.....	4
DIFERENCIA DE UNA TABLA DE SQL.....	4
CÓMO SE ALMACENA LA INFORMACIÓN EN MONGODB Y QUÉ FORMATO UTILIZA	4
JSON VS BSON	5
ESTRUCTURAS DE LOS ARCHIVOS JSON.....	6
VENTAJAS DE MONGO	6
• Plataforma de datos para desarrolladores completamente basada en la nube:	6
COMANDOS PARA REALIZAR EL CRUD EN MONGO	9
CÓMO SE PUEDEN RELACIONAR DATOS EN MONGO SIN USAR JOINS COMO EN SQL	9
BIBLIOGRAFÍA WAMP-28	11

INTRODUCCIÓN

MongoDB se ha convertido en una herramienta esencial para desarrolladores y empresas que buscan rendimiento, escalabilidad y facilidad de uso en la gestión de sus datos. A lo largo de este documento, profundizaremos en sus funcionalidades y beneficios para comprender por qué es una de las bases de datos NoSQL más utilizadas.

TIPO DE DATO MONGO

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.

DIFERENCIA DE UNA BASE DE DATOS RELACIONAL COMO MySQL

Son dos sistemas de administración de bases de datos que puede utilizar para almacenar y administrar datos. MySQL es un sistema de base de datos relacional que almacena datos en un formato tabular estructurado. Por el contrario, MongoDB almacena los datos como documentos JSON en un formato más flexible. Ambos ofrecen rendimiento y escalabilidad, pero brindan un mejor rendimiento según el caso de uso.

COLECCIÓN DE MONGO

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.

DIFERENCIA DE UNA TABLA DE SQL

- En SQL, debe crear una tabla y definir los tipos de datos; en MongoDB, no puede crear una colección, se crea automáticamente al insertar datos.
- En SQL, debe insertar valores según los tipos de datos; en MongoDB, puede insertar valores de cualquier tipo.
- En SQL, no se puede crear una columna en el momento de inserción o actualización; en MongoDB, es posible.
- En SQL, casi nada distingue entre mayúsculas y minúsculas; en MongoDB, todo distingue entre mayúsculas y minúsculas.

CÓMO SE ALMACENA LA INFORMACIÓN EN MONGODB Y QUÉ FORMATO UTILIZA

Usa un formato de almacenamiento de documentos llamado BSON, que es una forma binaria de JSON (JavaScript Object Notation o notación de objetos de JavaScript) que puede acomodar más tipos de datos.

JSON VS BSON

- **Binario frente a texto:** BSON es un formato de codificación binaria, mientras que JSON es un formato basado en texto. Esto significa que BSON es compacto para transmitirse a través de una red, mientras que JSON es legible por humanos y más fácil de trabajar en diversos contextos.
- **Soporte de datos ampliado:** JSON se limita a los tipos de datos de JavaScript, como cadenas, números, booleanos, nulos, objetos y matrices. Estos tipos de datos pueden combinarse para representar tipos de datos complejos. BSON admite tipos de datos adicionales (como datos binarios y tipos de fecha) que JSON no admite.
- 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 algunas situaciones, los documentos BSON pueden ser más grandes que los documentos JSON equivalentes porque incluyen metadatos adicionales e información de tipo que no está presente en JSON. Esto puede afectar a los tiempos de transmisión y a los requisitos de almacenamiento, especialmente en el caso de grandes conjuntos de datos. Tanto BSON como JSON pueden beneficiarse de la compresión.

- **Complejidad y compatibilidad:** BSON es más complejo que JSON, lo que dificulta el trabajo con él en determinados contextos. Los desarrolladores pueden tener que aprender nuevos tipos de datos y métodos de codificación/decodificación para trabajar con BSON de forma eficaz. También pueden surgir problemas de compatibilidad cuando se intercambian datos entre sistemas que no soportan completamente los tipos adicionales de BSON.

ESTRUCTURAS DE LOS ARCHIVOS JSON

La estructura de los datos JSON se basa en dos elementos: una colección de pares clave / valor y una lista ordenada de valores. Esto es similar a los tipos de datos utilizados en muchos lenguajes de programación, como los objetos y los arrays.

<https://www.arsys.es/blog/formato-json-que-es-y-para-que-sirve#:~:text=La%20estructura%20de%20los%20datos,los%20objetos%20y%20los%20arrays.>

VENTAJAS DE MONGO

- **Plataforma de datos para desarrolladores completamente basada en la nube:**
Es mucho más que una base de datos. Es una plataforma completa de datos para desarrolladores. Con MongoDB Atlas, la solución en la nube de MongoDB, tiene acceso a una colección de servicios que se integran perfectamente con su base de datos.
- **Esquemas de documentos flexibles:**

El modelo de documentos de MongoDB permite modelar y manipular prácticamente cualquier estructura de datos fácilmente. El formato de datos BSON de MongoDB, inspirado en JSON, permite tener objetos en una colección con diferentes conjuntos de campos (por ejemplo, el segundo nombre de un usuario solo cuando corresponda, o información específica de una región que solo se aplica a algunos registros). MongoDB permite la creación de esquemas explícitos y la validación de datos. Esta flexibilidad es una ventaja fundamental al gestionar datos reales y cambios en los requisitos o el entorno.

- **Acceso nativo al código y con amplio soporte:**

La mayoría de las bases de datos obligan al uso de envoltorios pesados, como los ORM (mapeadores objeto-relacionales), para convertir los datos en objetos y usarlos en programas. La decisión de MongoDB de almacenar y representar los datos en formato de documento permite acceder a ellos desde cualquier lenguaje, en estructuras de datos nativas de ese lenguaje (p. ej., diccionarios en Python, objetos en JavaScript, mapas en Java, etc.).

- **Diseño que facilita el cambio**

Si está acostumbrado a tener que cerrar su sitio o aplicación para cambiar la estructura de sus datos, está de suerte: MongoDB está diseñado para el cambio.

- **Consultas y análisis potentes**

MongoDB está diseñado para facilitar el acceso a los datos y rara vez requiere uniones o transacciones, pero cuando se necesitan consultas complejas, cumple con

creces la tarea. La API de consultas de MongoDB le permite realizar consultas profundas en documentos e incluso ejecutar análisis complejos con solo unas pocas líneas de código declarativo.

- **Escalabilidad horizontal sencilla con fragmentación**

MongoDB está diseñado desde cero para ser una base de datos distribuida. Cree clústeres con replicación en tiempo real y fragmente colecciones grandes o de alto rendimiento en varios clústeres para mantener el rendimiento y escalar horizontalmente .

- **Alto rendimiento (velocidad)**

Gracias al modelo de documento de MongoDB, la información se puede incrustar en un solo documento en lugar de recurrir a costosas operaciones de unión de bases de datos relacionales tradicionales. Esto agiliza considerablemente las consultas y devuelve toda la información necesaria en una sola llamada a la base de datos. En cuanto al rendimiento de escritura, MongoDB ofrece funciones para insertar y actualizar varios registros simultáneamente con insertMany y updateMany . Estas dos funciones ofrecen una mejora significativa del rendimiento en comparación con las escrituras por lotes en bases de datos tradicionales.

- **Instalación sencilla**

Con MongoDB Atlas, crear y configurar un clúster de MongoDB es más fácil que nunca. Con solo unos clics en la intuitiva interfaz de usuario, puede implementar

una nueva instancia gratuita para siempre . En cuestión de minutos, podrá conectarse a su base de datos mediante la cadena de conexión proporcionada.

- **Rentable**

MongoDB ofrece múltiples enfoques flexibles. Al usar MongoDB Atlas en la nube, puede elegir un tamaño de instancia que se ajuste a sus necesidades. También puede ajustar su clúster para que escale automáticamente cuando sea necesario.

COMANDOS PARA REALIZAR EL CRUD EN MONGO

COMANDOS	FUNCIONALIDAD
Show dbs	Muestra toda la base de datos
use database_name	Crear base de datos o usar una ya existente
db.help()	Muestra las funciones que podemos hacer en la base de datos en uso
db.collectionName()	Muestra todas las funciones que le podemos hacer a una colección
db.collectionName.insertOne({})	Crea documentos dentro de una colección
insertMany()	Inserta diferentes documentos en una colección
find() y findOne().	Búsqueda de documentos

CÓMO SE PUEDEN RELACIONAR DATOS EN MONGO SIN USAR JOINS COMO EN SQL

En MongoDB se pueden relacionar datos mediante el operador \$lookup, que permite fusionar información de un documento a otro.

El operador \$lookup se utiliza como tubería dentro del método aggregate().

- \$lookup (Agregación) crea una unión externa izquierda con otra colección.
- \$lookup ayuda a filtrar datos de los datos fusionados

IMAGEN DE MONGO

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

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview mongo

C:\Users\Laura Jhuliana velas>
```

<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	mongo b81a621037ef	latest	Unused	7 days ago	887.43 MB	
<input type="checkbox"/>	mysql 5568fddd4f66	latest	In use	2 months ago	796.97 MB	
<input type="checkbox"/>	mysql 65f3f983cb08	<none>	In use (dangling)	1 year ago	632.37 MB	

Showing 4 items

Walkthroughs

BIBLIOGRAFÍA WAMP-28

<https://datascientest.com/es/mongodb-todo-sobre-la-base-de-datos-nosql-orientada-a-documentos#:~:text=ingenier%C3%ADa%20de%20datos.-,MongoDB%20es%20una%20base%20de%20datos%20NoSQL%20orientada%20a%20documentos,almacenar%20como%20colecciones%20y%20documentos.>

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

<https://platzi.com/clases/1533-mongodb-basico/18481-bases-de-datos-colecciones-y-documentos-en-mongodb/>

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

<https://www.purestorage.com/es/knowledge/what-is-mongodb.html#:~:text=MongoDB%20usa%20un%20formato%20de,acomodar%20m%C3%A1s%20tipos%20de%20datos.>

<https://www.couchbase.com/es/resources/concepts/json-vs-bson/>

<https://www.mongodb.com/resources/compare/advantages-of-mongodb#:~:text=MongoDB%20ofrece%20muchas%20ventajas%20sobre,amplio%20soporte%20y%20c%C3%B3digo%20nativo>

<https://platzi.com/tutoriales/1533-mongodb-basico/4102-cosas-basicas-de-un-crud-en-mongodb/>

