



MongoDB

Walter Martín Lopes

*“Flexibilidad, escalabilidad y rendimiento
para tus datos.”*

ÍNDICE

1. Introducción

- 1.1.** Contexto y motivación
- 1.2.** Objetivos del trabajo

2. Fundamentos de MongoDB

- 2.1.** Historia y evolución
- 2.2.** Características principales
- 2.3.** Comparación con bases de datos SQL

3. Conceptos básicos de MongoDB

- 3.1.** Documentos y colecciones
- 3.2.** **BSON** (*Binary JSON*)
- 3.3.** **CRUD** (*Create, Read, Update, Delete*)

4. Uso básico de MongoDB

- 4.1.** Instalación y configuración
- 4.2.** Interfaz de línea de comandos (mongo shell)

5. Conclusiones

- 5.1.** Reflexiones finales
- 5.2.** Perspectivas y tendencias futuras

6. Referencias bibliográficas

1. Introducción

1.1 Contexto y motivación

MongoDB es una base de datos NoSQL, específicamente de tipo documental, que ha ido ganando popularidad en los últimos años debido a su capacidad para manejar grandes volúmenes de datos, su escalabilidad y flexibilidad. En el ámbito del desarrollo web, el uso de bases de datos NoSQL como MongoDB ha crecido debido a que permite modelar datos de manera más natural y eficiente en comparación con los sistemas de bases de datos relacionales tradicionales.

La motivación de este trabajo voluntario radica en la importancia de conocer y comprender el funcionamiento de MongoDB y cómo este puede ser aplicado en el desarrollo de aplicaciones web. Con la creciente demanda de aplicaciones que puedan manejar grandes volúmenes de datos y ofrecer alta disponibilidad, es crucial para los desarrolladores web estar familiarizados con las bases de datos NoSQL y sus ventajas.

1.2 Objetivos del trabajo

El objetivo principal de este trabajo es proporcionar una visión general de MongoDB, incluyendo sus fundamentos, conceptos básicos y cómo integrarlo con lenguajes de programación como Python y JavaScript (Node.js). A continuación, se detallan los objetivos específicos:

- Presentar el contexto y la motivación detrás del uso de MongoDB en el desarrollo de aplicaciones web.
- Explicar los fundamentos de MongoDB, incluyendo su historia, evolución y características principales.
- Comparar MongoDB con las bases de datos SQL tradicionales y destacar sus diferencias y ventajas.
- Introducir los conceptos básicos de MongoDB, como documentos, colecciones, BSON y operaciones CRUD.
- Reflexionar sobre las perspectivas y tendencias futuras de MongoDB en el ámbito del desarrollo web.

2. Fundamentos

2.1 Historia y evolución

MongoDB fue creado por Dwight Merriman y Eliot Horowitz en 2007, quienes fundaron la empresa 10gen (actualmente conocida como MongoDB Inc.). La idea de desarrollar una base de datos NoSQL surgió debido a las limitaciones que encontraron en las bases de datos relacionales tradicionales al abordar aplicaciones web de gran escala, como escalabilidad y manejo de datos semi-estructurados.

El nombre MongoDB proviene de la palabra *"humongous"* (enorme en inglés), refiriéndose a la capacidad de manejar grandes cantidades de datos. La primera versión pública de MongoDB, la versión 1.0, fue lanzada en agosto de 2009.

Desde su lanzamiento, MongoDB ha experimentado un crecimiento constante en popularidad y adopción en la industria. A lo largo de los años, se han lanzado varias versiones que han mejorado su rendimiento, funcionalidades y facilidad de uso. Hoy en día, MongoDB es una de las bases de datos NoSQL más populares y se utiliza en una amplia variedad de aplicaciones y sectores.

2.2 Características principales

MongoDB posee varias características que lo diferencian de las bases de datos relacionales y lo hacen atractivo para el desarrollo de aplicaciones web. Algunas de las características principales son:

- **Modelo de datos basado en documentos:** Utiliza un modelo de datos flexible basado en documentos, donde cada registro se almacena como un documento en formato **BSON** (*Binary JSON*). Esto permite representar estructuras de datos complejas y jerárquicas, facilitando la adaptación a diferentes esquemas de datos.
- **Escalabilidad horizontal:** Ofrece escalabilidad horizontal a través de la partición automática de datos (*sharding*). Esto permite distribuir los datos en varios servidores, mejorando la capacidad de manejar grandes volúmenes de datos y mantener un alto rendimiento.
- **Alta disponibilidad y tolerancia a fallos:** Utiliza un sistema de réplica llamado "replica sets" para garantizar la disponibilidad y la tolerancia a fallos. Estos consisten en un grupo de servidores que mantienen copias de los datos y pueden asumir el rol de servidor primario en caso de fallo del servidor primario actual.
- **Consultas y agregación flexibles:** Ofrece un lenguaje de consulta y un sistema de agregación potentes y flexibles. Es posible realizar consultas y filtrar datos utilizando campos específicos, rangos, expresiones regulares, entre otros.

2.3 Comparación con BD SQL

MongoDB y las bases de datos SQL tienen diferencias fundamentales en cuanto a su modelo de datos, escalabilidad y casos de uso ideales. A continuación, se presentan algunas comparaciones clave:

- **Modelo de datos:** SQL usa tablas con esquemas fijos, mientras que MongoDB emplea documentos flexibles.
- **Escalabilidad:** SQL escala verticalmente (*más recursos en un único servidor*), MongoDB escala horizontalmente (*distribuye datos en múltiples servidores*).
- **Transacciones:** SQL tiene soporte completo para transacciones *ACID*; MongoDB ha añadido transacciones multi-documento en versiones recientes.
- **Lenguaje de consulta:** SQL utiliza un lenguaje de consulta estandarizado, mientras que MongoDB usa un lenguaje de consulta basado en JSON.

La elección entre ambos dependerá de las necesidades específicas del proyecto, como la flexibilidad del esquema de datos y los requisitos de escalabilidad.

3. Conceptos básicos

3.1 Documentos y colecciones

En MongoDB, los datos se almacenan en documentos, que son estructuras de datos similares a JSON pero en formato BSON (Binary JSON). Un documento puede contener campos y valores de distintos tipos de datos, así como estructuras anidadas como listas y subdocumentos.

Las colecciones son grupos de documentos que se organizan juntos en la base de datos. Una colección es similar a una tabla en una base de datos SQL, pero no requiere un esquema fijo. Esto significa que los documentos dentro de una colección pueden tener estructuras diferentes, lo que permite mayor flexibilidad en el almacenamiento y la gestión de datos.

En resumen, los documentos y las colecciones son elementos fundamentales en MongoDB. Los documentos representan registros de datos individuales y flexibles, mientras que las colecciones agrupan y organizan documentos en la base de datos. Esta organización permite realizar consultas y operaciones de manera eficiente y adaptarse a diferentes necesidades de almacenamiento.

3.2 BSON (Binary JSON)

BSON como hemos explicado anteriormente, es el formato de almacenamiento utilizado por MongoDB para representar los documentos en la base de datos. BSON es una representación binaria de la estructura de datos JSON. Aunque BSON es similar a JSON, ofrece algunas ventajas en el contexto de MongoDB:

- **Mayor eficiencia en el almacenamiento y procesamiento:** BSON utiliza una representación binaria compacta, lo que reduce el espacio de almacenamiento y acelera las operaciones de lectura y escritura.
- **Tipos de datos adicionales:** BSON admite más tipos de datos que JSON, incluyendo tipos binarios, fechas y números decimales, lo que enriquece las posibilidades de representación de datos en MongoDB.

MongoDB utiliza BSON para almacenar documentos y realizar consultas, ya que ofrece un equilibrio entre flexibilidad, eficiencia y capacidad de representación de datos.

3.3 CRUD (Create, Read, Update, Delete)

CRUD es un acrónimo que representa las cuatro operaciones básicas que se pueden realizar sobre los datos en una base de datos: *Crear*, *Leer*, *Actualizar* y *Eliminar*. MongoDB ofrece métodos para llevar a cabo estas operaciones en los documentos y colecciones:

- **Create:** Para crear y agregar documentos a una colección, MongoDB proporciona el método *insertOne()* para insertar un único documento, y *insertMany()* para insertar múltiples documentos en una sola operación.
- **Read:** Para leer documentos en una colección, MongoDB ofrece el método *find()*, que permite realizar consultas y filtros sobre los documentos utilizando diversos criterios y operadores.
- **Update:** Para actualizar documentos en una colección, MongoDB proporciona métodos como *updateOne()* y *updateMany()*, que permiten modificar documentos según criterios de búsqueda y utilizando operadores de actualización.
- **Delete:** Para eliminar documentos de una colección, MongoDB ofrece métodos como *deleteOne()* y *deleteMany()*, que eliminan documentos según los criterios de búsqueda especificados.

Estas operaciones CRUD son esenciales para interactuar con los datos en MongoDB y permiten a los desarrolladores manipular y gestionar la información almacenada en las bases de datos de manera efectiva.

4. Uso básico

4.1 Instalación y configuración

Para instalar MongoDB en tu sistema, debes visitar el sitio web oficial (<https://www.mongodb.com/>) y descargar la versión adecuada para tu sistema operativo (Windows, macOS o Linux). Sigue las instrucciones de instalación proporcionadas en la documentación oficial para completar el proceso.

Una vez instalado, es necesario configurar MongoDB para que se ejecute como un servicio en segundo plano. En sistemas basados en Linux, esto se puede hacer mediante archivos de configuración y comandos del sistema, mientras que en Windows y macOS, puedes utilizar la interfaz gráfica de usuario del administrador de servicios MongoDB.

4.2 Interfaz de línea de comandos

MongoDB incluye una interfaz de línea de comandos llamada “***Mongo shell***”, que permite interactuar con la base de datos mediante comandos. Para acceder a mongo shell, abre una terminal o línea de comandos en tu sistema y ejecuta el comando ‘*mongo*’. Esto iniciará la shell de MongoDB y te conectará a la instancia local de la base de datos por defecto.

Dentro de mongo shell, puedes ejecutar comandos para gestionar la base de datos, como crear y seleccionar bases de datos, crear y consultar colecciones, e insertar, actualizar y eliminar documentos. La sintaxis de los comandos es similar a la de JavaScript, ya que la shell utiliza JavaScript como lenguaje de scripting.

El uso de mongo shell es fundamental para administrar y manipular datos en MongoDB, especialmente durante el desarrollo y la depuración de aplicaciones. También es útil para realizar tareas de mantenimiento y administración de bases de datos en entornos de producción.

5. Conclusiones

5.1 Reflexiones finales

MongoDB es una base de datos NoSQL que ofrece flexibilidad, escalabilidad y rendimiento, lo que la convierte en una excelente opción para muchas aplicaciones modernas, especialmente aquellas que requieren manejar grandes volúmenes de datos y esquemas de datos cambiantes. A través de su modelo de documentos y sus características de escalabilidad horizontal, MongoDB se adapta bien a los desafíos actuales en el desarrollo de aplicaciones web.

5.2 Perspectivas y tendencias futuras

En el futuro, se espera que la popularidad de las bases de datos NoSQL, como MongoDB, continúe creciendo debido a la creciente necesidad de manejar datos no estructurados y semi-estructurados, así como a la demanda de soluciones escalables y de alto rendimiento.

Además, la integración con tecnologías emergentes como la inteligencia artificial, el aprendizaje automático y el análisis de datos en tiempo real también impulsará el desarrollo y adopción de MongoDB.

La evolución y mejora de las funcionalidades de MongoDB, como las transacciones multi-documento y la compatibilidad con diferentes lenguajes de programación, seguirán siendo un área de enfoque para mantenerse al día con las tendencias y necesidades del mercado.

6. Referencias bibliográficas

6. Referencias bibliográficas

A continuación, se enumeran algunas referencias bibliográficas usadas y recursos en línea relevantes:

- **Sitio oficial MongoDB :**
<https://www.mongodb.com/>
- **Youtube:** Parte de la información usada para la elaboración del trabajo ha sido sacada de videos de esta plataforma.
- **Wikipedia:** <https://es.wikipedia.org/wiki/MongoDB>
- **Otras páginas como:**
 - <https://kinsta.com/>
 - <https://openwebinars.net/>
 - <https://datascientest.com/>
 - <https://keepcoding.io/>