

Taller Mongo

Paula Alejandra Martínez Huertas

Ingeniería de sistemas, Corporación Universitaria Minuto de Dios

NRC 60747: Base de datos masivas

Ing. William Matallana Porras

Marzo 21, 2025

Contenido

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

Introducción

En el mundo del manejo de datos, las bases de datos NoSQL han ganado popularidad por su capacidad de adaptarse a estructuras dinámicas y su eficiencia en aplicaciones modernas. Entre ellas, MongoDB destaca como una de las más utilizadas por su enfoque orientado a documentos, lo que representa una alternativa poderosa frente a las bases de datos relacionales tradicionales como MySQL.

A lo largo de este trabajo se abordará qué tipo de base de datos es MongoDB y cómo se diferencia de un modelo relacional. Se explicará el concepto de colecciones en MongoDB y su relación con las tablas en SQL, así como el formato en el que se almacena la información y cómo se estructura internamente usando BSON. Durante este documento se hablará sobre la estructura de los archivos JSON y elementos clave en la representación de los datos.

Asimismo, se analizarán las ventajas que ofrece MongoDB en cuanto a escalabilidad y flexibilidad, aspectos que lo hacen ideal para proyectos con grandes volúmenes de datos o estructuras variables. Se presentarán los comandos básicos para realizar operaciones CRUD, y se explorará cómo es posible manejar relaciones entre datos en MongoDB sin utilizar los clásicos JOIN de SQL.

Finalmente, se explicará cómo obtener la imagen oficial de MongoDB a través de Docker, y se mencionarán algunas herramientas gráficas que facilitan la administración y visualización de los datos, similares a lo que ofrece MySQL Workbench en el entorno relacional.

Esta visión general permitirá comprender cómo funciona MongoDB, sus principales características y su aplicación práctica en el desarrollo de sistemas actuales.

Objetivos

1. Identificar el tipo de base de datos que representa MongoDB, así como sus diferencias fundamentales con una base de datos relacional como MySQL.
2. Explicar la estructura y el rol de las colecciones en MongoDB, comparándolas con las tablas utilizadas en SQL.
3. Describir el formato de almacenamiento que utiliza MongoDB, enfocándose en el uso de documentos BSON y su relación con JSON.
4. Evaluar las ventajas que ofrece MongoDB en términos de escalabilidad y flexibilidad, especialmente en contextos de grandes volúmenes de datos o estructuras cambiantes.
5. Presentar los comandos esenciales para realizar operaciones CRUD en MongoDB y su aplicación práctica.
6. Demostrar el proceso de instalación de MongoDB mediante Docker, utilizando la imagen oficial del sistema.

Desarrollo

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

Es un sistema de gestión de bases de datos (DBMS) no relacional de código abierto que utiliza documentos flexibles en lugar de tablas y filas para procesar y almacenar diversas formas de datos.

Como solución de base de datos NoSQL, MongoDB no requiere un sistema de gestión de bases de datos relacionales (RDBMS), por lo que proporciona un modelo de almacenamiento de datos flexible que permite a los usuarios almacenar y consultar tipos de datos multivariados con facilidad. Esto no únicamente simplifica la gestión de bases de datos para los desarrolladores, sino que también crea un entorno altamente escalable para aplicaciones y servicios multiplataforma.

La principal diferencia con una base de datos relacional es cómo almacenan y organizan los datos:

- En MongoDB, los datos se almacenan en documentos con formato similar a JSON, lo que permite tener una estructura flexible. No es necesario definir un esquema fijo antes de guardar los datos. Cada documento puede tener diferentes campos, y se agrupan en colecciones.
- En MySQL, los datos se organizan en tablas, con filas y columnas. Cada tabla tiene un esquema rígido, lo que significa que debes definir de antemano qué columnas tendrá cada fila y qué tipo de datos se permiten.

Otra diferencia es:

- MongoDB no usa relaciones entre datos como claves foráneas; en su lugar, suele guardar datos relacionados dentro del mismo documento (lo que se llama *embebido*).
- MySQL sí usa relaciones entre tablas mediante claves primarias y foráneas, lo que permite organizar los datos de manera más estructurada, pero también más rígida.

2. ¿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. Por su parte se diferencia de una tabla SQL en los siguientes aspectos:

- **Estructura de los datos**

MongoDB (colección): Los documentos en una colección son de tipo BSON (Binary JSON), lo que significa que son flexibles y no requieren un esquema fijo. Cada documento puede tener una estructura diferente, con distintos campos y tipos de datos.

SQL (tabla): Las tablas tienen un esquema rígido, donde todas las filas deben seguir la misma estructura de columnas con tipos de datos predefinidos.

- **Relaciones**

MongoDB: No tiene un sistema de relaciones integrado como en SQL. En lugar de usar claves foráneas, MongoDB permite anidar documentos o usar referencias entre colecciones.

SQL: Las tablas pueden relacionarse entre sí mediante claves primarias y foráneas, lo que permite un modelo de datos relacional.

- **Escalabilidad**

MongoDB: Está diseñado para ser altamente escalable horizontalmente, lo que significa que puede distribuir datos en múltiples servidores fácilmente.

SQL: La escalabilidad horizontal es más compleja en bases de datos relacionales, ya que están optimizadas para escalabilidad vertical.

- **Lenguaje de consulta**

MongoDB: Utiliza consultas basadas en documentos, con operadores como find, aggregate, etc.

SQL: Utiliza el lenguaje SQL estándar, con consultas como SELECT, JOIN, WHERE, etc.

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

Almacena y maneja mayores cantidades de datos que las bases de datos relacionales tradicionales. MongoDB 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. Estos documentos se agrupan dentro de colecciones (que serían como las “tablas” en una base de datos relacional), por lo tanto cada documento es independiente y puede tener una estructura diferente (esquema flexible).

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

BSON está diseñado como una representación binaria de datos JSON , con extensiones específicas para aplicaciones más amplias y optimizado para el almacenamiento y el recorrido de datos. Al igual que JSON, BSON admite la incrustación de objetos y matrices.

Una forma particular en la que BSON se diferencia de JSON es en que asistancia técnica algunos tipos de datos más avanzados. JSON no diferencia, por instancia, entre enteros (que son números redondos) y números en coma flotante (que tienen varios grados de precisión decimal).

5. Estructura de los archivos json

- **Objetos**

Los objetos en JSON son una colección de pares clave-valor. Se encierran entre llaves «{}» y cada clave se separa de su valor correspondiente por dos puntos «:». Los pares clave-valor se separan entre sí por comas «,». Por ejemplo:

- La clave «nombre» tiene el valor «Facundo»
- La clave «edad» es 30
- La clave «ciudad» tiene el valor «Madrid»

- **Arreglos**

Los arreglos o arrays JSON son simples colecciones de elementos. Estos elementos pueden ser datos como números o cadenas, pero también otros arreglos u objetos JSON.

Los arrays se encierran entre corchetes «[]» y los valores se separan entre sí por comas «,».

- **Tipos de datos**

El formato JSON soporta un número limitado de tipos de datos. Son los siguientes:

- Cadenas
- Números
- Booleanos (true / false)
- Objetos JSON
- Arrays, booleanos (verdadero / falso)
- null (vacío)

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

7. Comandos para realizar CRUD en Mongo

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

9. Descargar imagen de mongo en docker

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

Conclusiones

Bibliografía

<https://www.ibm.com/mx-es/topics/mongodb>

[https://platzi.com/clases/1533-mongodb-basico/18481-bases-de-datos-](https://platzi.com/clases/1533-mongodb-basico/18481-bases-de-datos-colecciones-y-documentos-en-mongodb/#:~:text=Las%20Colecciones%20son%20agrupaciones%20de,registros%20dentro%20de%20las%20colecciones.)

[colecciones-y-documentos-en-](https://platzi.com/clases/1533-mongodb-basico/18481-bases-de-datos-colecciones-y-documentos-en-mongodb/#:~:text=Las%20Colecciones%20son%20agrupaciones%20de,registros%20dentro%20de%20las%20colecciones.)

[mongodb/#:~:text=Las%20Colecciones%20son%20agrupaciones%20de,registros%2](https://platzi.com/clases/1533-mongodb-basico/18481-bases-de-datos-colecciones-y-documentos-en-mongodb/#:~:text=Las%20Colecciones%20son%20agrupaciones%20de,registros%20dentro%20de%20las%20colecciones.)

[0dentro%20de%20las%20colecciones.](https://platzi.com/clases/1533-mongodb-basico/18481-bases-de-datos-colecciones-y-documentos-en-mongodb/#:~:text=Las%20Colecciones%20son%20agrupaciones%20de,registros%20dentro%20de%20las%20colecciones.)

DeepSeek. Diferencia de una colección en una tabla en SQL.

<https://chat.deepseek.com/a/chat/s/af02ed24-a809-4d11-a7b5-e32f00dde86d>

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

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

[%C3%A1s%20tipos%20de%20datos.](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.mongodb.com/es/resources/basics/json-and-bson>

<https://www.arsys.es/blog/formato-json-que-es-y-para-que-sirve#tree-4>