

Clase 10: Introducción a MongoDB

Sitio: [Centro de E-Learning - UTN.BA](#)
Curso: Curso de Backend Developer - Turno
Libro: Noche
Clase 10: Introducción a MongoDB

Imprimido
por:
Nilo Crespi
Día:
Friday, 23 de January de 2026,
10:22

Descripción

Objetivos

- Comprender qué es MongoDB y sus fundamentos.
- Aprender a instalar MongoDB y configurarlo.
- Interactuar con la base de datos a través de la línea de comandos y MongoDB Compass.
- Realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar).
- Optimizar consultas mediante índices.

Tabla de contenidos

- 1. Introducción**
- 2. Introducción a MongoDB y su enfoque NoSQL**
- 3. Interacción con la base de datos**
- 4. Creación de bases de datos y colecciones**

1. Introducción

Esta clase se enfoca en brindar una comprensión esencial de MongoDB y cómo encaja en el mundo de las bases de datos NoSQL. Incluye desde una introducción a MongoDB y su historia hasta conceptos fundamentales, aplicaciones, instalación en diversas plataformas y la interacción con la base de datos a través de la línea de comandos y MongoDB Compass.

Además, cubre la creación de bases de datos y colecciones, así como las operaciones CRUD para gestionar datos. Finalmente, se explora la importancia de los índices y las estrategias de optimización de consultas para un rendimiento óptimo en aplicaciones basadas en MongoDB.

2. Introducción a MongoDB y su enfoque NoSQL

Introducción a MongoDB y su enfoque NoSQL

¿Qué es MongoDB?

MongoDB es un sistema de gestión de bases de datos NoSQL (No Structured Query Language) de código abierto ampliamente utilizado. Se caracteriza por su enfoque en bases de datos orientadas a documentos, lo que significa que los datos se almacenan en formato BSON (Binary JSON) en documentos, en lugar de tablas y filas como en bases de datos relacionales.

MongoDB es conocido por su escalabilidad horizontal, su capacidad para manejar grandes volúmenes de datos y su flexibilidad, ya que no requiere un esquema de datos fijo, permitiendo agregar campos a los documentos de manera dinámica. Es una opción popular en el desarrollo de aplicaciones web, móviles y otros sistemas que requieren un almacenamiento de datos eficiente y versátil.

Historia de MongoDB

MongoDB fue desarrollado por la empresa 10gen, que luego cambió su nombre a MongoDB Inc., y su historia comienza en 2007 cuando Dwight Merriman, Eliot Horowitz y Kevin Ryan fundaron la empresa. La inspiración detrás de MongoDB se basó en las limitaciones de las bases de datos relacionales tradicionales en términos de escalabilidad y flexibilidad.

En 2009, se lanzó la primera versión de MongoDB, y desde entonces ha experimentado un crecimiento significativo y una evolución constante. A lo largo de los años, MongoDB ha visto numerosas actualizaciones y mejoras, con la introducción de características clave como la indexación geoespacial, capacidades de agregación avanzadas, replicación, fragmentación (sharding) y la adopción de características de seguridad avanzada.

La comunidad de MongoDB también ha crecido de manera constante, con una amplia base de usuarios y una sólida contribución a la comunidad de código abierto. MongoDB se ha convertido en una de las bases de datos NoSQL más populares y es ampliamente utilizado en una variedad de aplicaciones, desde sitios web hasta aplicaciones móviles y sistemas de análisis de datos.

La evolución de MongoDB sigue en marcha, con el compromiso de brindar una base de datos moderna y versátil para satisfacer las necesidades cambiantes de desarrollo de aplicaciones. Su historia demuestra cómo la innovación y la adaptación constante pueden llevar a una tecnología a la vanguardia de la industria.

Conceptos Fundamentales

Documentos: En MongoDB, un documento es la unidad básica de almacenamiento de datos. Cada documento representa una entidad o un registro y se almacena en formato BSON (Binary JSON). Estos documentos son estructuras flexibles y pueden contener datos de diferentes tipos, lo que permite una fácil adaptación a las necesidades cambiantes de una aplicación.

Colecciones: Las colecciones son contenedores lógicos en los que se agrupan los documentos relacionados. Pueden considerarse análogas a las tablas en una base de datos relacional, pero con la diferencia clave de que no requieren un esquema fijo. Esto significa que los documentos en una colección pueden tener campos diferentes y estructuras de datos variadas.

Documentos BSON

BSON, o Binary JSON, es un formato de intercambio de datos utilizado por MongoDB y diseñado para ser eficiente en términos de espacio y velocidad. Aunque se asemeja a JSON en su estructura, BSON tiene ventajas significativas en aplicaciones que requieren almacenar y recuperar datos de manera eficiente. A continuación, se explican las características clave de BSON:

- **Binario eficiente:** BSON utiliza una representación binaria para los datos en lugar de texto, lo que lo hace más compacto y eficiente en términos de espacio de almacenamiento y ancho de banda de red. Esto es especialmente beneficioso en aplicaciones con grandes volúmenes de datos.
- **Tipos de datos ricos:** BSON admite una amplia variedad de tipos de datos, incluidos enteros, decimales, fechas, cadenas de texto, matrices y documentos anidados. Esto permite representar datos de manera más precisa que en otros formatos de intercambio de datos.
- **Ordenado:** BSON mantiene el orden de los campos en un documento, lo que facilita la búsqueda y recuperación de datos. Esta característica es útil para aplicaciones que necesitan un control preciso sobre la estructura de los datos.
- **Uso en MongoDB:** MongoDB almacena sus datos en formato BSON. Esto significa que los documentos en una colección de MongoDB se almacenan como documentos BSON, lo que permite una integración perfecta entre la base de datos y las aplicaciones que utilizan este formato.

NoSQL y MongoDB

NoSQL es un término que se refiere a "Not Only SQL" o "No solo SQL". Representa un enfoque alternativo para la gestión de datos que se aparta de las bases de datos relacionales tradicionales.

Las bases de datos NoSQL se caracterizan por su capacidad para manejar datos no estructurados o semiestructurados y ofrecer flexibilidad en el diseño de la base de datos. A diferencia de las bases de datos relacionales, no requieren un esquema fijo y permiten agregar campos y estructuras de datos dinámicamente. Esto hace que sean ideales para aplicaciones donde los requisitos cambian con frecuencia y se necesita una rápida adaptación.

Dentro de las bases de datos NoSQL, existen varios tipos, y uno de los más prominentes es el **bases**

de datos de documentos, en el cual MongoDB se destaca. Estas bases de datos almacenan datos en documentos, que son estructuras flexibles que pueden contener datos en formato BSON (Binary JSON). Los documentos se agrupan en colecciones, lo que permite una organización lógica de los datos.

Dónde Utilizar MongoDB

MongoDB se utiliza en una amplia variedad de aplicaciones y casos de uso debido a su flexibilidad y escalabilidad. Aquí hay algunos ejemplos de dónde brilla MongoDB:

- **Aplicaciones Web:** MongoDB es una elección popular para el desarrollo de aplicaciones web, incluidas las redes sociales, sitios de comercio electrónico y portales de noticias, donde la capacidad de manejar grandes volúmenes de datos y adaptarse a cambios en la estructura de datos es esencial.
- **Gestión de Contenido:** Las aplicaciones que gestionan contenido, como sistemas de gestión de contenido (CMS), blogs y portales de noticias, se benefician de la capacidad de MongoDB para almacenar y recuperar contenido de manera eficiente.
- **Análisis de Datos:** Es utilizado en aplicaciones de análisis de datos y business intelligence para almacenar y consultar grandes conjuntos de datos, lo que facilita el descubrimiento de patrones y tendencias.
- **Aplicaciones Móviles:** MongoDB se utiliza en el desarrollo de aplicaciones móviles para almacenar datos del usuario, como preferencias y registros, y para proporcionar una experiencia de usuario personalizada.
- **IoT (Internet de las cosas):** En aplicaciones de IoT, donde se recopilan grandes cantidades de datos de sensores y dispositivos, MongoDB se utiliza para almacenar y analizar estos datos en tiempo real.
- **Catálogos de Productos:** Las empresas de comercio electrónico utilizan MongoDB para gestionar catálogos de productos que contienen información detallada sobre productos y variaciones.
- **Sistemas de Gestión de Identidad:** En aplicaciones de autenticación y autorización, como sistemas de gestión de identidad, MongoDB almacena perfiles de usuario y credenciales de forma segura.
- **Juegos en Línea:** Los juegos en línea con millones de usuarios concurrentes utilizan MongoDB para gestionar el estado del juego, las puntuaciones y la información del jugador.
- **Contenido Geoespacial:** Aplicaciones que requieren almacenamiento y consulta de datos geoespaciales, como mapas y aplicaciones de seguimiento en tiempo real, encuentran útil la capacidad de MongoDB para indexar y buscar datos basados en ubicación.
- **Registros y Bitácoras:** MongoDB se utiliza para almacenar registros y bitácoras de aplicaciones, lo que facilita la depuración y el análisis de problemas.

Instalación y configuración de MongoDB en diferentes plataformas

La instalación y configuración de MongoDB en diferentes plataformas es un proceso esencial para comenzar a utilizar esta base de datos NoSQL. A continuación, te proporcionaré una visión general de los pasos comunes para instalar y configurar MongoDB en varias plataformas:

Instalación en Windows

1. Descarga el instalador de MongoDB para Windows desde el sitio web oficial.
1. Ejecuta el instalador y sigue las instrucciones del asistente de instalación.
2. Durante la instalación, puedes elegir si deseas instalar MongoDB como servicio de Windows para que se inicie automáticamente con el sistema.
3. Configura el directorio de datos y el puerto si es necesario.
4. Verifica que MongoDB se ha instalado correctamente ejecutando `mongod --version` en la línea de comandos.

Instalación en macOS

1. Utiliza el sistema de gestión de paquetes Homebrew para instalar MongoDB en macOS. Ejecuta el siguiente comando en la Terminal:

```
brew tap mongodb/brew
```

```
brew install mongodb-community
```

2. Inicia el servicio de MongoDB con el comando `brew services start mongodb/brew/mongodb-community`.

3. Verifica la instalación ejecutando `mongod --version`.

Instalación en Linux (Ubuntu)

1. Agrega el repositorio oficial de MongoDB a tu sistema con los siguientes comandos:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv  
9DA31620334BD75D9DCB49F368818C72E52529D4
```

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/4.4  
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
```

2. Actualiza el sistema con `sudo apt-get update`.
3. Instala MongoDB con `sudo apt-get install -y mongodb-org`.
4. Inicia y habilita el servicio MongoDB con `sudo systemctl start mongod` y `sudo systemctl enable mongod`.
5. Verifica la instalación con `mongod --version`.

Instalación en contenedores Docker

1. Descarga e instala Docker en tu sistema si aún no lo tienes.
2. Ejecuta MongoDB en un contenedor Docker con el siguiente comando:

```
docker run -d -p 27017:27017 --name mongodb mongo
```

3. Puedes acceder a la instancia de MongoDB en el puerto 27017 en tu máquina host.

Ten en cuenta que estos son pasos generales, y los detalles exactos pueden variar según las versiones específicas de MongoDB y las plataformas. Asegúrate de consultar la documentación oficial de MongoDB para obtener instrucciones detalladas sobre la instalación y configuración en tu plataforma específica.

3. Interacción con la base de datos

Interacción con la base de datos a través de la interfaz de línea de comandos

La interfaz de línea de comandos (CLI) de MongoDB es una herramienta poderosa que te permite interactuar con la base de datos, administrar colecciones, realizar consultas y ejecutar diversas operaciones.

A continuación, se presentan algunas de las operaciones y comandos más comunes que puedes realizar a través de la interfaz de línea de comandos de MongoDB:

Iniciar el Servidor MongoDB:

Para iniciar el servidor de MongoDB, utiliza el siguiente comando:

```
mongod
```

Conectar a MongoDB:

Para conectarte a MongoDB, usa el comando `mongo`. Puedes especificar la dirección IP y el puerto del servidor si no se está ejecutando en el puerto predeterminado (27017). Por ejemplo:

```
mongo --host <dirección_IP>:<puerto>
```

Seleccionar una Base de Datos:

Utiliza el comando `use` para seleccionar una base de datos específica. Por ejemplo:

```
use miBaseDeDatos
```

Listar Bases de Datos:

Puedes listar todas las bases de datos disponibles con el comando `show dbs`.

Listar Colecciones:

Para ver las colecciones en la base de datos actual, ejecuta `show collections`.

Insertar Documentos:

Puedes insertar documentos en una colección utilizando el método `insert` o `insertOne`. Por ejemplo:

```
db.miColeccion.insert({campo1: valor1, campo2: valor2})
```

Consultar Documentos:

Utiliza el método `find` para realizar consultas en una colección. Por ejemplo:

```
db.miColeccion.find({campo: valor})
```

Actualizar Documentos:

Puedes actualizar documentos con el método `update`. Por ejemplo:

```
db.miColeccion.update({campo: valor}, {$set: {nuevoCampo: nuevoValor}})
```

Eliminar Documentos:

Utiliza el método `remove` para eliminar documentos. Por ejemplo:

```
db.miColeccion.remove({campo: valor})
```

Índices:

Puedes crear índices en campos específicos para acelerar las consultas con el comando `createIndex`.

Ayuda:

Para obtener ayuda sobre los comandos disponibles, puedes escribir `help` en la interfaz de línea de comandos de MongoDB. También puedes obtener ayuda específica para un comando, por ejemplo, `help db.find`.

La interfaz de línea de comandos de MongoDB es una herramienta versátil para administrar y trabajar con tus bases de datos MongoDB. Te permite realizar tareas de administración y ejecutar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de manera eficiente. A medida que te familiarices con los comandos y operaciones, podrás aprovechar al máximo esta potente herramienta.

Uso de la interfaz gráfica MongoDB Compass para gestionar la base de datos

MongoDB Compass es una interfaz gráfica de usuario (GUI) que facilita la administración y la interacción con tus bases de datos MongoDB. Ofrece una forma más visual y amigable de trabajar con tus datos.

The screenshot shows the MongoDB Compass interface with the 'Databases' tab selected. On the left, there's a sidebar with 'My Queries' and 'Databases'. Under 'Databases', there's a search bar and a list of databases: admin, config, local (which is highlighted with a red box), and test. Each database entry shows its storage size, the number of collections, and the number of indexes. For example, the 'local' database has a storage size of 471.04 kB, 6 collections, and 8 indexes.

Database	Storage size	Collections	Indexes
admin	40.96 kB	0	2
config	65.54 kB	5	12
local	471.04 kB	6	8
test	24.58 kB	2	2

A continuación, se describen algunas de las tareas que puedes realizar con MongoDB Compass:

- Conexión a una Base de Datos:** Al abrir MongoDB Compass, puedes establecer una nueva conexión a una base de datos MongoDB. Proporciona la dirección IP del servidor, el puerto y las credenciales de autenticación.
- Exploración de Bases de Datos y Colecciones:** Una vez que te hayas conectado a una base de datos, puedes explorar las bases de datos y colecciones disponibles en una interfaz gráfica. Esto facilita la visualización de la estructura de tus datos.
- Consultas Visuales:** permite realizar consultas en tus colecciones mediante una interfaz visual. Puedes crear y personalizar consultas sin necesidad de escribir código manualmente.
- Visualización de Datos:** permite visualizar los datos de tus colecciones en diferentes formatos, incluyendo tablas y gráficos. Esto facilita la comprensión de tus datos.
- Creación de Índices:** Puedes crear índices en campos específicos de tus colecciones con MongoDB Compass para optimizar el rendimiento de tus consultas.
- Importación y Exportación de Datos:** facilita la importación y exportación de datos desde y hacia tus bases de datos. Esto es útil para transferir datos y realizar copias de seguridad.
- Visualización y Modificación de Documentos:** Puedes ver y editar documentos individualmente con una vista detallada que te muestra todos los campos y sus valores.
- Explicación de Consultas:** proporciona detalles sobre cómo se ejecutan tus consultas, lo que es útil para optimizarlas.
- Administración de Usuarios y Roles:** Puedes administrar usuarios y roles en tu base de datos, incluyendo la creación, modificación y eliminación de cuentas.

- **Configuración de Opciones del Servidor:** Puedes configurar opciones del servidor, como límites de recursos y opciones de seguridad.

MongoDB Compass es especialmente útil para aquellos que prefieren una interfaz gráfica para interactuar con MongoDB en lugar de utilizar la línea de comandos. Facilita la administración y visualización de tus datos de manera intuitiva. Puedes descargar MongoDB Compass desde el sitio web oficial de MongoDB y comenzar a trabajar con tus bases de datos MongoDB de manera eficiente.

4. Creación de bases de datos y colecciones

Creación de bases de datos y colecciones

Creación de una Base de Datos:

1. Para crear una nueva base de datos, primero debes conectarte a MongoDB a través de la línea de comandos o una interfaz gráfica como MongoDB Compass.
2. Utiliza el comando `use` seguido del nombre de la base de datos que deseas crear. Por ejemplo, para crear una base de datos llamada "miBaseDeDatos," escribe:

```
use miBaseDeDatos
```

3. MongoDB creará la base de datos si no existe. Sin embargo, no se mostrará en la lista de bases de datos hasta que tenga al menos una colección.

Creación de una Colección:

- Para crear una colección en una base de datos, primero debes seleccionar la base de datos en la que deseas crear la colección utilizando el comando `use`.
- Utiliza el método `createCollection` en la interfaz de línea de comandos o la función equivalente en el lenguaje de programación que estés utilizando. Por ejemplo, para crear una colección llamada "usuarios," puedes usar el siguiente comando:

```
db.createCollection("usuarios")
```

- La colección se creará en la base de datos especificada.

Es importante destacar que en MongoDB, las bases de datos y las colecciones se crean de forma automática cuando se inserta el primer documento en una colección. Esto significa que no es necesario crear explícitamente una base de datos o una colección antes de comenzar a insertar datos.

A medida que insertes documentos en una colección, MongoDB se encargará de crear la colección si no existe en la base de datos especificada. Del mismo modo, si seleccionas una base de datos que no existe, MongoDB la creará al insertar datos en ella.

Operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en MongoDB CRUD

En MongoDB, puedes realizar una serie de operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para

interactuar con tus datos. A continuación, se muestran las operaciones CRUD más comunes en MongoDB:

Crear (Create):

- **Insertar Documentos:** Puedes insertar nuevos documentos en una colección utilizando el método `insert`, `insertOne`, o `insertMany`. Por ejemplo:

```
db.miColeccion.insert({campo1: valor1, campo2: valor2})
```

Leer (Read):

- **Consultar Documentos:** Puedes recuperar documentos que cumplan con ciertos criterios utilizando el método `find`. Por ejemplo, para encontrar todos los documentos en una colección:

```
db.miColeccion.find()
```

- **Consultas con Criterios:** Puedes especificar criterios para filtrar documentos. Por ejemplo, para encontrar documentos con un campo específico:

```
db.miColeccion.find({campo: valor})
```

- **Consultas con Proyección:** Puedes limitar los campos que se devuelven en una consulta utilizando el método `find` con proyección.

```
db.miColeccion.find({}, {campo1: 1, campo2: 1})
```

Actualizar (Update):

- **Actualizar Documentos:** Puedes actualizar documentos existentes utilizando el método `update`. Por ejemplo, para actualizar un documento con un nuevo valor:

```
db.miColeccion.update({campo: valor}, {$set: {nuevoCampo: nuevoValor}})
```

- **Reemplazar Documentos:** También puedes reemplazar completamente un documento con uno nuevo utilizando el método `update` con la opción `multi: false`.

Eliminar (Delete):

- **Eliminar Documentos:** Puedes eliminar documentos que cumplan con ciertos criterios utilizando el método `remove`. Por ejemplo, para eliminar documentos con un campo específico:

```
db.miColeccion.remove({campo: valor})
```

- **Eliminar Todos los Documentos:** Si no especificas criterios, el método `remove` eliminará todos los documentos en la colección.

Recuerda que MongoDB es una base de datos flexible y esquema libre, lo que significa que los documentos en una colección pueden tener diferentes campos. Esto permite una gran versatilidad en las operaciones CRUD.

Además de estas operaciones básicas, MongoDB también ofrece muchas otras operaciones avanzadas y funciones que permiten realizar tareas más complejas, como la agregación, la creación de índices, la geolocalización y más. Las operaciones CRUD son fundamentales para interactuar con tus datos, ya sea a través de la interfaz de línea de comandos o mediante la utilización de bibliotecas en lenguajes de programación que son compatibles con MongoDB.

Índices y optimización de consultas en MongoDB

Los índices en MongoDB son herramientas fundamentales para la optimización de consultas, ya que permiten acelerar la recuperación de datos al proporcionar un acceso más rápido a los documentos en una colección. A continuación, se describen los conceptos clave relacionados con índices y la optimización de consultas en MongoDB.

MongoDB admite varios tipos de índices, incluyendo índices simples, compuestos, geoespaciales, de texto completo y otros.

Creación de Índices:

Puedes crear índices en una o más claves de un documento utilizando el método `createIndex` o declarando índices en el esquema (por ejemplo, en Mongoose para Node.js).

Ventajas de los Índices

- **Mejora del Rendimiento:** Los índices aceleran las operaciones de lectura al permitir una búsqueda más eficiente de documentos.
- **Reducción de la Carga en el Servidor:** Al utilizar índices, MongoDB puede recuperar datos más rápidamente, lo que disminuye la carga en el servidor y mejora la respuesta de la aplicación.

Estrategias de Optimización de Consultas

- **Consulta Eficiente:** Diseña tus consultas de manera eficiente para aprovechar al máximo los índices. Utiliza operadores de comparación y lógica para filtrar datos de manera precisa.
- **Índices Completos:** Asegúrate de que tus consultas coincidan con los índices completos para obtener un rendimiento óptimo.
- **Actualización de Datos:** Ten en cuenta que la creación y el mantenimiento de índices pueden aumentar la sobrecarga al realizar actualizaciones de datos. Es importante equilibrar el rendimiento de lectura y escritura.
- **Monitoriza el Rendimiento:** Utiliza herramientas de monitorización para supervisar el rendimiento de tu sistema y ajustar los índices según sea necesario.

En resumen, el uso efectivo de índices y estrategias de optimización de consultas en MongoDB es

esencial para garantizar un rendimiento óptimo en tus aplicaciones. Debes diseñar y mantener índices adecuados, utilizar consultas eficientes y supervisar el rendimiento para identificar áreas de mejora. La optimización de consultas es un proceso continuo que contribuye significativamente a la eficacia de tu base de datos MongoDB.

Bibliografía utilizada y sugerida

- MongoDB documentation. (s. f.). MongoDB Documentation. <https://www.mongodb.com/docs/>
- MongoDB getting started. (s. f.). https://www.w3schools.com/mongodb/mongodb_get_started.php
- MongoDB Indexing & Search. (s. f.). https://www.w3schools.com/mongodb/mongodb_indexing_search.php
- What is MongoDB Compass? — MongoDB Compass. (s. f.). <https://www.mongodb.com/docs/compass/current/>