

Informe Bases de Datos No Relacionales

Harold Mosquera

Adrián Ordoñez

Universidad del Putumayo

Índice

Índice.....	2
Resumen.....	3
Introducción.....	4
Metodología	5
Desarrollo del informe.....	6
Consultas.....	7
Análisis y Discusión	9
Conclusiones	10
Referencias	11

Resumen

El siguiente informe de MongoDB como una base de datos NoSQL que es flexible y eficiente ya que es altamente versátil, que está diseñado para manejar grandes volúmenes de datos no estructurados y semiestructurados, MongoDB nos permite almacenar los datos de forma jerárquica y anidada lo que es bueno para aplicaciones web modernas y también para el análisis de datos.

En este informe se destaca las ventajas de usar MongoDB como una solución moderna y eficiente para las bases de datos en la era digital.

Introducción

En la era digital actual, la gestión eficiente de grandes volúmenes de datos se ha convertido en un desafío crucial para las organizaciones. Las bases de datos no relacionales, comúnmente conocidas como NoSQL, han surgido como una solución viable para manejar datos no estructurados y semiestructurados. A diferencia de las bases de datos relacionales tradicionales, que organizan los datos en tablas con filas y columnas, las bases de datos NoSQL utilizan diversos modelos de datos, como documentos, pares clave-valor, columnas anchas y grafos (MongoDB, 2023).

MongoDB es una de las bases de datos NoSQL más populares y ampliamente adoptadas. Esta base de datos orientada a documentos almacena datos en un formato similar a JSON (JavaScript Object Notation), lo que permite una mayor flexibilidad y capacidad para manejar estructuras de datos complejas y jerárquicas (DataCamp, 2023). La flexibilidad del esquema, la escalabilidad horizontal y el alto rendimiento son algunas de las características distintivas que hacen de MongoDB una opción preferida para desarrolladores y empresas que buscan soluciones de bases de datos modernas y eficientes.

El presente documento explora las características principales de MongoDB, sus ventajas en comparación con las bases de datos relacionales y los casos de uso más comunes en los que se aplica esta tecnología. A través de este análisis, se pretende proporcionar una comprensión integral de cómo MongoDB puede contribuir a la gestión eficaz de datos en diversas aplicaciones.

Metodología

En la realización de este informe sobre la gestión de datos en MongoDB, se emplearon dos herramientas clave: MongoDB Compass y Mongo Shell. Estas herramientas permitieron llevar a cabo las operaciones necesarias para la creación, consulta, actualización y eliminación de colecciones y documentos de forma eficiente.

MongoDB Compass: Es una herramienta gráfica que facilita la visualización y gestión de los datos en MongoDB. Se utilizó para explorar las colecciones y realizar consultas básicas sin la necesidad de escribir código, lo que permitió una rápida validación de los datos almacenados en las colecciones clientes y facturas. Además, MongoDB Compass ofreció una visión clara del rendimiento de las consultas y permitió analizar los índices creados en las colecciones.

Mongo Shell: Esta herramienta de línea de comandos de MongoDB fue fundamental para la ejecución de comandos más complejos y personalizados, como la creación de colecciones, la inserción masiva de documentos, la ejecución de consultas de agregación, y la actualización o eliminación de documentos. El uso de Mongo Shell permitió un control detallado sobre las operaciones de base de datos y la posibilidad de realizar tareas avanzadas de manipulación de datos.

Ambas herramientas complementaron el proceso de desarrollo y análisis del esquema de base de datos, facilitando tanto la manipulación visual como la programación directa de los comandos utilizados.

Desarrollo del informe

La base de datos NoSQL está diseñada para manejar datos no estructurados y semiestructurados de manera eficiente. En este caso, la base de datos contiene dos colecciones principales: clientes y facturas.

Colección: Clientes

La colección clientes almacena información sobre los clientes de una empresa. Cada documento en esta colección representa un cliente individual y puede incluir los siguientes campos:

nombre: El nombre completo del cliente.

email: La dirección de correo electrónico del cliente.

teléfono: El número de teléfono del cliente.

dirección: Un subdocumento que contiene la calle, ciudad, estado y código postal del cliente.

fecha_registro: La fecha en que el cliente se registró en el sistema.

Colección: Facturas

La colección facturas almacena información sobre las transacciones de facturación. Cada documento en esta colección representa una factura individual y puede incluir los siguientes campos:

id_cliente: El identificador único del cliente al que se le emitió la factura (relacionado con la colección de clientes).

fecha_factura: La fecha en que se emitió la factura.

monto_total: El monto total de la factura.

items: Un subdocumento que contiene una lista de los productos o servicios facturados, incluyendo la cantidad y el precio de cada uno.

estado: El estado de la factura (por ejemplo, pagada, pendiente, vencida).

Consultas

A continuación se presentan algunos ejemplos de consultas realizadas con MongoDB.

Buscar en clientes: La consulta find se utiliza para recuperar documentos de una colección. Por ejemplo, para buscar un cliente por su correo electrónico:

```
db.clientes.find({ "email": "juan.perez@example.com" })
```

Buscar en facturas: Similar a la anterior, la consulta find permite buscar documentos en la colección de facturas, por ejemplo, aquellas que están en estado pendiente:

```
db.facturas.find({ "estado": "pendiente" })
```

Actualizar clientes: La consulta updateOne permite modificar documentos existentes en una colección. En este ejemplo, se actualiza el número de teléfono de un cliente específico:

```
db.clientes.updateOne(  
  { "email": "juan.perez@example.com" },  
  { $set: { "teléfono": "+54123456789" } }  
)
```

Actualizar facturas: De manera similar, se puede actualizar el estado de una factura de pendiente a pagada:

```
db. facturas. updateOne (  
  
  {"id_cliente": "2", "estado": "pendiente"},  
  
  {$set: {"estado": "pagada"}}  
  
)
```

Eliminar documentos en clientes

La consulta deleteOne permite eliminar un documento específico de una colección. En este caso, se elimina un cliente en función de su correo electrónico:

```
db. clientes.deleteOne ({ 'email': 'juan.perez@example.com' })
```


Análisis y Discusión

Los resultados obtenidos en el manejo de las colecciones de MongoDB muestran la eficiencia y flexibilidad de esta base de datos NoSQL para almacenar y gestionar grandes volúmenes de datos no estructurados. A lo largo de las consultas realizadas (búsqueda, actualización y eliminación de documentos), se evidenció que MongoDB permite una manipulación ágil y escalable de los datos. Además, el uso de herramientas como MongoDB Compass y Mongo Shell facilita tanto la interacción visual como la ejecución de comandos avanzados, proporcionando al desarrollador un control total sobre las operaciones de la base de datos.

Conclusiones

El objetivo principal de este informe era explorar y demostrar cómo MongoDB puede ser una solución eficaz para gestionar datos no estructurados, comparándolo con las bases de datos relacionales tradicionales. Los resultados confirman que MongoDB es una herramienta muy útil para trabajar en entornos donde los datos cambian mucho y necesitan estructuras flexibles. Además, su capacidad para manejar grandes volúmenes de datos de manera rápida y eficiente la convierte en una excelente opción para empresas que buscan mejorar cómo manejan sus datos.

El análisis de las colecciones "clientes" y "facturas" permitió demostrar cómo MongoDB maneja datos relacionales a través de identificadores como `id_cliente`, mostrando la capacidad de gestionar datos complejos mediante subdocumentos anidados (por ejemplo, los campos "dirección" e "items"). Estas características son especialmente valiosas para aplicaciones web modernas que requieren alta flexibilidad y velocidad de procesamiento.

Referencias

MongoDB. (2023). *NoSQL Explained*. Recuperado de
<https://www.mongodb.com/es/resources/basics/databases/nosql-explained>

DataCamp. (2023). *Tutorial de NoSQL con MongoDB*. Recuperado de
<https://www.datacamp.com/es/tutorial/nosql-tutorial>