

## Bases de datos NOSQL estudiadas en la materia “Bases de datos para Big data”

Nuestro curso consta de 10 temas que cubren los fundamentos de la adquisición y almacenamiento de datos que se caracterizan por su variedad y flexibilidad para **estructurarlos con esquemas flexibles** que no se adecuan a esquemas estrictos como es el caso de los modelos de bases de datos relacionales.

Estudiamos en el curso varios modelos de datos y sus correspondientes sistemas de bases de datos comúnmente llamadas bases de datos NOSQL.

Comenzamos con los modelos de datos orientados a documentos y el manejador de bases de datos mongodb.

### **MongoDB**

MongoDB es una base de datos NoSQL (No Only SQL) de código abierto que se ha vuelto muy popular en los últimos años. A diferencia de las bases de datos relacionales tradicionales, MongoDB se basa en un modelo de datos flexible basado en documentos, lo que lo hace adecuado para una variedad de aplicaciones modernas. A continuación, algunas características de MongoDB:

#### **¿Qué es MongoDB?**

- MongoDB es una base de datos NoSQL que almacena datos en forma de documentos BSON (Binary JSON).
- Desarrollado por MongoDB, Inc., es de código abierto y ampliamente utilizado en aplicaciones web, móviles y empresariales.

#### **Características Principales:**

- Modelo de Datos Basado en Documentos:
- MongoDB almacena datos en documentos BSON, que son estructuras de datos similares a JSON.
- Cada documento puede tener una estructura diferente, lo que facilita la flexibilidad en la representación de datos.

#### **Escalabilidad Horizontal:**

- MongoDB es altamente escalable y permite la distribución de datos en múltiples servidores.
- Ofrece la posibilidad de escalabilidad horizontal, lo que significa que puedes agregar más servidores para manejar cargas de trabajo crecientes.
- Alto Rendimiento:
- MongoDB está diseñado para un alto rendimiento y puede manejar grandes volúmenes de datos y consultas.
- Utiliza técnicas de indexación y almacenamiento en memoria para acelerar las consultas.

**Facilidad de Uso:**

- MongoDB es fácil de usar debido a su modelo de datos flexible y su lenguaje de consulta intuitivo.
- Las operaciones de lectura/escritura son simples y eficientes.

**Consultas Avanzadas:**

- MongoDB admite consultas avanzadas, incluyendo consultas geoespaciales y consultas de texto completo.

**Replicación y Tolerancia a Fallos:**

- MongoDB permite configurar réplicas de datos para garantizar la alta disponibilidad y tolerancia a fallos.
- En caso de fallos, un servidor secundario puede asumir el rol de servidor principal.

**Colecciones y Bases de Datos:**

- Los documentos se organizan en colecciones, que son similares a tablas en bases de datos relacionales.
- Las colecciones se agrupan en bases de datos.

**Indexación:**

- MongoDB admite índices para acelerar la búsqueda de datos.
- Puedes crear índices en campos específicos para mejorar el rendimiento de consultas.

**Seguridad:**

- MongoDB ofrece funciones de seguridad, como autenticación y control de acceso basado en roles (RBAC).

## Apache Cassandra

### ¿Qué es Apache Cassandra?

Apache Cassandra es una base de datos NoSQL de código abierto que se diseñó originalmente en Facebook y luego se convirtió en un proyecto de la Apache Software Foundation.

Está diseñado para gestionar grandes cantidades de datos distribuidos en múltiples servidores y proporciona alta escalabilidad y disponibilidad.

### Características Principales:

#### Modelo de Datos Descentralizado:

- Cassandra utiliza un modelo de datos descentralizado en el que cada nodo en el clúster es igual y no existe un nodo maestro.
- Cada nodo es autónomo y puede atender solicitudes de lectura y escritura.

#### Escalabilidad Horizontal:

- Cassandra es altamente escalable y permite la adición de nuevos nodos para manejar cargas de trabajo crecientes.
- Escalabilidad horizontal: se puede agregar más hardware para mejorar el rendimiento.

#### Alta Disponibilidad:

- Cassandra está diseñado para proporcionar alta disponibilidad y tolerancia a fallos.
- Los datos se replican en múltiples nodos para evitar la pérdida de datos en caso de fallo del nodo.

#### Esquema Dinámico:

- Cassandra permite un esquema dinámico, lo que significa que no se requiere un esquema fijo para los datos.
- Cada fila en una tabla puede tener columnas diferentes sin afectar la consistencia de los datos.

#### Consultas y Lecturas Rápidas:

- Cassandra es eficiente para la lectura y escritura de datos y es adecuado para aplicaciones con requisitos de baja latencia.

#### Sopporte para Datos No Estructurados y Semiestructurados:

- Cassandra es adecuado para datos no estructurados y semiestructurados, como registros de eventos y datos de series temporales.

### **Consistencia Configurable:**

- Puedes configurar el nivel de consistencia de los datos según tus necesidades.
- Ofrece opciones para equilibrar la consistencia y la disponibilidad.

### **Soporte para Consultas y Búsquedas:**

- Cassandra admite consultas utilizando su lenguaje de consulta CQL (Cassandra Query Language).
- También es compatible con la búsqueda de texto completo.

### **Seguridad y Control de Acceso:**

- Ofrece funciones de seguridad, como autenticación y control de acceso.
- Puedes configurar roles y permisos de usuario.

## **Neo4j**

Neo4j es una base de datos de grafos de alto rendimiento que almacena datos en forma de nodos y relaciones, lo que permite modelar y consultar datos de manera intuitiva y eficiente.

Es ampliamente utilizado en aplicaciones que involucran datos altamente conectados y complejas relaciones, como redes sociales, recomendaciones, análisis de fraude y más.

### **Características Principales:**

#### **Modelo de Datos de Grafo:**

- Neo4j se basa en un modelo de datos de grafo en el que los datos se representan como nodos, relaciones y propiedades.
- Esto permite representar y consultar relaciones de manera eficiente.

#### **Lenguaje de Consulta Cypher:**

- Neo4j utiliza un lenguaje de consulta llamado Cypher que permite expresar consultas de grafo de manera intuitiva.
- Cypher facilita la búsqueda de patrones y la navegación de relaciones en el grafo.

#### **Escalabilidad:**

- Neo4j es escalable y puede manejar bases de datos de grafo de diversos tamaños.
- Ofrece opciones para escalar horizontal o verticalmente según las necesidades.

#### **Alta Disponibilidad:**

- Neo4j admite la alta disponibilidad y la tolerancia a fallos a través de la replicación y la comutación por error.
- Garantiza la disponibilidad de los datos en caso de fallas.

#### **Transacciones ACID:**

- Neo4j ofrece garantías de integridad y consistencia de datos al admitir transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad).

#### **Biblioteca de Algoritmos de Grafo:**

- Neo4j proporciona una amplia gama de algoritmos de grafo predefinidos que pueden utilizarse en aplicaciones de análisis y procesamiento de grafos.

#### **APIs y Conectores:**

- Neo4j es compatible con una variedad de lenguajes de programación y ofrece conectores y bibliotecas para facilitar la integración con aplicaciones.

#### **Visualización de Datos:**

Se pueden utilizar herramientas de visualización de datos para representar gráficamente los datos almacenados en Neo4j y comprender mejor las relaciones.

## **Redis**

Redis es un almacén de estructuras de datos en memoria, de código abierto, que se utiliza como base de datos, caché y agente de mensajes. Soporta estructuras de datos como strings, hashes, listas, sets, sorted sets con consultas de rango, bitmaps, hyperloglogs, índices geoespaciales y flujos de datos. Redis tiene replicación incorporada, transacciones Lua scripting, LRU eviction, transacciones y diferentes niveles de persistencia en disco. Es conocido por su alta velocidad, con operaciones que pueden ejecutarse en menos de un milisegundo, ofreciendo un rendimiento excepcional para aplicaciones en tiempo real.

Redis utiliza un modelo de datos basado en llave-valor, donde cada valor puede ser de tipos como string, lista, set, sorted set, hash, bitmap, hyperloglog o geoespacial. Este modelo permite operaciones de alta velocidad sobre los datos, como buscar, añadir o eliminar valores, y es especialmente útil para estructuras de datos complejas utilizadas en aplicaciones web, cachés, colas de mensajes, y más. La simplicidad del modelo de llave-valor combinada con la rica tipología de datos soportada hace de Redis una solución versátil para una amplia gama de problemas de almacenamiento en memoria.

En un clúster de Redis, los datos se distribuyen entre varios nodos, lo que permite escalabilidad horizontal y alta disponibilidad. El clúster automáticamente partitiona los datos y puede manejar fallas de nodos, reequilibrando y replicando los datos según sea necesario. Los clientes se comunican con cualquier nodo; este puede redirigir operaciones si el dato buscado reside en otro nodo. Este modelo soporta operaciones de lectura y escritura en tiempo real a gran escala, distribuyendo la carga y aumentando la capacidad de procesamiento y almacenamiento.