

Column-Family



Key-Value



# NoSQL

Document



Graph



Ivana Sánchez Pérez

2ºASIR

## **Introducción**

Son muchas las aplicaciones web que utilizan algún tipo de bases de datos para funcionar. Hasta ahora estábamos acostumbrados a utilizar bases de

datos SQL como son MySQL, Oracle o MS SQL, pero desde hace ya algún tiempo, y ante la necesidad de responder a las altas demandas de procesamiento de grandes volúmenes de datos en ambientes distribuidos, han aparecido otras que reciben el nombre de NoSQL, más flexibles y escalables.

## La aparición de la tecnología NoSQL

Estas bases de datos se utilizan para agrupar sistemas de administración diferentes a los tradicionales relacionales. Los conceptos de Big Data, Big Users y Cloud Computing han motivado fuertemente el desarrollo de la tecnología NoSQL. Google, Amazon, Facebook, Mozilla, Adobe y LinkedIn son algunas de las empresas que descubrieron las limitaciones de la tecnología relacional y desarrollaron sus propios sistemas.

Se puede decir que la aparición del término NoSQL, aparece con la llegada de la web 2.0 ya que hasta ese momento sólo subían contenido en la red aquellas empresas que tenían un portal, pero con la llegada de aplicaciones como las nombradas anteriormente, cualquier usuario podía subir contenido, provocando así un crecimiento exponencial de los datos. Este es el momento en el que surgen los primeros problemas de la gestión de toda esa información almacenada en bases de datos relacionales.

En un principio, para solucionar estos problemas de accesibilidad, las empresas optaron por utilizar un mayor número de máquinas, pero además de no solucionar el problema era muy costoso. Por lo que se pasó a crear sistemas pensados para un uso específico que con el paso del tiempo han dado lugar a soluciones robustas, apareciendo así el movimiento NoSQL.

## ¿Qué son?

Las bases de datos NoSQL son un tipo de sistemas de gestión de bases de datos diseñados para trabajar con grandes volúmenes de datos no estructurados o semiestructurados.

Podemos decir que las NoSQL son estructuras que nos permiten almacenar información en aquellas situaciones en las que las bases de datos relacionales generan ciertos problemas debido principalmente a problemas de escalabilidad y rendimiento de las bases de datos relacionales donde se dan cita miles de usuarios concurrentes y con millones de consultas diarias, y también son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación. Tampoco utilizan una estructura de datos en forma de tabla donde se van almacenando los datos, sino que para el almacenamiento hacen uso de otros formatos como clave-valor, mapeo de columnas o grafos.

# Tipos

Dependiendo de la forma en la que almacenen la información, hay varios tipos distintos de bases de datos NoSQL. Los más utilizados son:

1. **Bases de datos de documentos:** Almacenan datos en formato de documentos, generalmente en formato **JSON** , **BSON** , o **XML** . Los datos están organizados como pares clave-valor, pero cada valor es un documento que puede tener estructuras complejas.
  - Ejemplos: **MongoDB**, **CouchDB**
2. **Bases de datos clave-valor :** Almacenan datos como pares clave-valor, donde cada clave es única y está asociada a un valor que puede ser cualquier tipo de dato.
  - Ejemplos: **Redis**, **DynamoDB**
3. **Bases de datos de columna ancha :** Estas bases de datos organizan los datos en tablas, filas y columnas, pero permiten un número variable de columnas por fila. Son ideales para grandes volúmenes de datos distribuidos.
  - Ejemplo: **Apache Cassandra**, **HBase**
4. **Bases de datos de gráficos:** Diseñadas para trabajar con relaciones complejas entre datos, estas bases de datos representan la información como nodos y las relaciones como aristas.
  - Ejemplo: **Neo4j**, **Amazon Neptune**

## Ejemplo: MongoDB (Base de datos de documentos)

Elegiré **MongoDB** como base de datos para profundizar, ya que es una de las más populares y ampliamente utilizadas en el entorno NoSQL.

# Elección

Me he decantado por **MongoDB** como base de datos para profundizar, ya que es una de las más populares y ampliamente utilizadas en el entorno NoSQL.

## Características principales de MongoDB:

1. **Modelo basado en documentos:** MongoDB almacena los datos como **BSON** (una representación binaria de JSON), lo que permite que los datos

tengan una estructura flexible y que se adapten fácilmente a diferentes tipos de información.

2. **Escalabilidad horizontal:** MongoDB está diseñado para escalar horizontalmente, lo que significa que puede distribuirse fácilmente entre varios servidores para manejar grandes volúmenes de datos sin comprometer el rendimiento.
3. **Consultas ricas:** Aunque es una base de datos NoSQL, MongoDB admite consultas avanzadas, indexación y agregaciones. Permite realizar búsquedas por múltiples criterios, y ofrece soporte para índices en varios campos.
4. **Alta disponibilidad:** MongoDB admite la replicación a través de **réplicas** de conjuntos, lo que significa que se puede configurar para que los datos se repliquen en Múltiples servidores, asegurando la disponibilidad incluso en caso de fallos.
5. **Ecosistema robusto:** MongoDB tiene un ecosistema rico en herramientas y servicios, como **MongoDB Atlas** para la administración en la nube, **MongoDB Stitch** para backend-as-a-service y **MongoDB Charts** para visualización de datos.

#### **Ventajas de MongoDB:**

- **Flexibilidad:** El esquema flexible permite almacenar diferentes tipos de documentos dentro de una misma colección, lo que lo hace ideal para proyectos en los que los requisitos de datos cambian con el tiempo.
- **Fácil de usar:** La estructura basada en documentos es intuitiva para desarrolladores, ya que el formato JSON es ampliamente utilizado en la programación web.
- **Gran rendimiento para grandes datos:** MongoDB está optimizado para manejar grandes volúmenes de datos y puede realizar operaciones de escritura rápida, lo que lo hace adecuado para aplicaciones en tiempo real.

#### **Limitaciones de MongoDB:**

- **Consistencia eventual:** En configuraciones distribuidas, MongoDB puede operar en modo de consistencia eventual, lo que significa que no garantiza que todos los nodos tengan los mismos datos en todo momento.
- **No es ideal para transacciones complejas:** Aunque ha mejorado su soporte para transacciones, aún no es tan sólido como en las bases de datos SQL tradicionales cuando se trata de operaciones complejas con Múltiples tablas.

### Casos de uso de MongoDB:

- **Aplicaciones web:** MongoDB es ampliamente utilizado en aplicaciones web que requieren almacenar datos de usuarios, preferencias y contenido generado por el usuario de forma dinámica.
- **Big Data:** MongoDB se utiliza en sistemas de análisis de grandes volúmenes de datos donde los datos pueden estar semiestructurados.
- **Sistemas de gestión de contenido:** Ideal para CMS que requieren manejar grandes cantidades de contenido de forma variada y flexible.

En resumen, MongoDB es una opción poderosa para gestionar datos dinámicos y no estructurados, ofreciendo escalabilidad y flexibilidad para aplicaciones modernas, aunque puede no ser la mejor opción para todas las situaciones, especialmente aquellas que requieren consistencia estricta o transacciones complejas.

