

TEMA 1

Introducción



Esta obra está bajo una [licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Icono diseñado por Flaticon

Introducción

Introducción a las Bases de datos NoSQL

¿Qué son los datos?

Corresponden a hechos o realidades del mundo real.

A partir de ellos, intentamos reconstruir la información del mundo real.

Son "almacenados" usando un método de comunicación (ej.: figuras o lenguajes) en un medio semipermanente de "registrarlos" (ej.: piedras o papel).

¿Qué son los datos?

Generalmente, el dato y su interpretación son recogidos juntos, en los lenguajes naturales

- Su altura es 175 cm
 - Dato: 175
 - Significado: altura en centímetros

A veces, los datos son separados de su interpretación

- Hora en un reloj
- Temperatura en un termómetro de la calle

¿Qué son los datos?

Los ordenadores han incrementado la separación entre datos y su significado:

- No se prestan para manipular en lenguaje natural
- El coste de almacenamiento es muy elevado

La interpretación de los datos es inherente a los programas que los utilizan:

- Dato: valores almacenados
- Información: significado de los datos

Almacenamiento de datos

Existen dos aproximaciones para el almacenamiento de los datos utilizados por un programa informático:

- Sistemas basados en ficheros
- Bases de datos

Sistemas basados en ficheros

En los sistemas basados en ficheros cada programa utiliza sus propios datos. Esto provoca una ocupación inútil de memoria, la aparición de inconsistencias y duplicidad de información.

Además, existe dependencia física entre los programas y los datos:

Sistemas basados en bases de datos

Cuando se utilizan bases de datos los programas "*comparten*" los datos:



¿Qué es una base de datos?

Conjunto de información (datos) **homogénea** de una organización, **almacenada** en un ordenador, y que permite realizar **consultas** y **actualizaciones** (inserciones, modificaciones y/o borrados).

¿Qué es una base de datos relacional?

Una base de datos **relacional** es un tipo de base de datos que utiliza un modelo matemático para describir y almacenar los datos. Este modelo se basa en **tablas**, **filas** y **columnas**, donde cada tabla representa un tipo de objeto o **entidad**, las filas son las **instancias de ese objeto** y las columnas son las **propiedades o atributos** de ese objeto.

¿Qué es una base de datos NoSQL?

NoSQL es un acrónimo de "**Not Only SQL**", se refiere a un tipo de base de datos que **no utiliza el modelo relacional** (tablas y filas) para almacenar datos. En su lugar, utiliza **estructuras de datos** más flexibles como **documentos, claves-valores, grafos, columnas**, entre otros. Estos tipos de bases de datos se caracterizan por ser escalables y manejar grandes cantidades de datos y usuarios simultáneos.

¿Cómo y para qué se *utilizan* las bases de datos NoSQL?

Las bases de datos **NoSQL** se utilizan en diferentes ámbitos y casos de uso:

- **Aplicaciones en tiempo real:** Se requiere una alta velocidad de lectura y escritura, como chatbots, juegos en línea, entre otros.
- **Big Data:** Ideales para el almacenamiento y análisis de grandes volúmenes de datos no estructurados, como datos de redes sociales, sensores, entre otros.
- **Microservicios:** Arquitecturas de microservicios para almacenar los datos de forma independiente.
- **Almacenamiento de objetos:** Aplicaciones que requieren almacenar objetos complejos con relaciones complejas entre ellos.
- **Aplicaciones móviles:** Aplicaciones móviles para almacenar y recuperar información de forma rápida y sin necesidad de conexión a un servidor central.

Introducción

Bases de datos SQL vs. Bases de datos NoSQL

¿Qué es una modelo de datos?

Un modelo de datos permite describir las propiedades de la información almacenada en una base de datos:

- Estructuras de datos
- Restricciones
- Dependencias
- Dominios

Los modelos de datos son fundamentales para introducir la abstracción en una base de datos.

¿Qué modelos de almacenamiento utilizan SQL y NoSQL?

- **SQL** utilizan tablas con columnas (atributos) y filas (registros) fijas. Estas bases de datos siguen reglas específicas relativas a la integridad y la coherencia.
- **NoSQL** no se ciñen a un formato rígido y tienden a pertenecer a una de estas cuatro categorías:
 - **Basadas en documentos:** almacenan y codifican los datos en documentos en formatos (JSON, XML, YAML y BSON).
 - **Basadas en grados:** estructuran los datos como nodos y relaciones para mostrar las conexiones entre los distintos elementos de datos.
 - **Basadas en columnas:** almacenan los datos en celdas agrupadas en un número ilimitado de columnas en lugar de filas.
 - **Basadas en clave-valor:** almacenan los datos como pares clave-valor, donde cada clave es un identificador único que corresponde a un valor asociado.



¿Qué esquemas de bases de datos utilizan SQL y NoSQL?

Las bases de datos SQL requiere un esquema *rígido, predefinido, estático o fijo*. Organiza los datos de forma tabular y relacional. Por lo tanto, es necesario *estructurar y organizar los datos* antes de crear una base de datos SQL .

Las bases de datos NoSQL tienen esquemas *flexibles y dinámicos* para datos que *no están estructurados*. Por lo tanto, no hay mucha necesidad de estructurar u organizar los datos antes de colocarlos en una base de datos NoSQL .

¿Hasta qué punto son escalables las bases de datos SQL y NoSQL?

Tanto SQL como NoSQL son escalables, aunque la naturaleza de su escalabilidad es diferente.

- **SQL** pueden escalar "*verticalmente*" si se supera la capacidad actual del servidor lo que significa que se puede aumentar la potencia de procesamiento del hardware actual migrando a un servidor más grande.
- **NoSQL** pueden escalar fácilmente de forma "*horizontal*" añadiendo más servidores para gestionar un mayor tráfico según sea necesario.

¡Atención! - Aunque las bases de datos **SQL** se pueden escalar horizontalmente, no están bien soportadas.

¿Es SQL más rápido que NoSQL?

En general, ni SQL ni NoSQL son más rápidos que el otro. Su velocidad depende más bien del contexto en el que se utilicen.

- Las bases de datos SQL :
 - Se diseñaron cuando el almacenamiento de datos era caro y la duplicación de datos podía hacer perder mucho dinero.
 - Están preparadas para ser más rápidas para consultas, uniones, actualizaciones, etc.
- Las bases de datos NoSQL :
 - Se diseñaron para datos no estructurados, es decir, pueden ser orientadas a columnas, grafos, documentos o tuplas clave-valor.
 - Los datos se almacenan juntos, es decir, es más rápido realizar operaciones de lectura o escritura en una entidad de datos.

Resumen: SQL es excelente para proteger la validez de los datos, mientras que NoSQL es ideal para cuando se necesita una rápida disponibilidad de big data.

Entrando en detalle: Pros y cons SQL

- Fiabilidad: son robustas y confiables, es decir, los datos están seguros y no se pierden fácilmente.
- Modelado de datos estructurado: se usa un modelado de datos estructurados y relacionales.
- Integridad de los datos: la integridad de los datos se garantizan y se mantiene la consistencia de los datos.
- Lenguaje de consulta estándar: SQL es un lenguaje estándar que es ampliamente utilizado y conocido.

Entrando en detalle: Pros y cons SQL

- Complejidad: pueden ser más complejas de implementar y mantener que otras.
- Costo: a menudo se requieren licencias y hardware.
- Rendimiento: pueden tener un rendimiento limitado.
- Dificultad de escalabilidad horizontal: pueden ser más difíciles de escalar horizontalmente que las bases de datos **NoSQL**.

Entrando en detalle: Pros y cons NoSQL

- Flexibilidad de datos: permiten una mayor flexibilidad en la estructura y el modelado de datos.
- Escalabilidad horizontal: son fáciles de escalar horizontalmente para manejar grandes cantidades de datos y usuarios simultáneos.
- Rendimiento: a menudo tienen un mejor rendimiento en situaciones de alta concurrencia o grandes cantidades de datos.
- Coste: suelen ser más económicas, ya que no requieren licencias o hardware especializado.

Entrando en detalle: Pros y cons NoSQL

- Integridad de los datos: pueden no tener las mismas características de integridad de datos que las bases de datos SQL, lo que puede comprometer la exactitud y consistencia de los datos.
- Lenguaje de consulta no estándar: tienen lenguajes de consulta no estándar que pueden ser menos conocidos.
- Dificultad en la realización de consultas complejas: tienen limitaciones en la realización de consultas complejas en comparación con las bases de datos SQL .
- Falta de soporte y recursos: son relativamente nuevas, puede haber menos soporte y recursos disponibles en comparación con las bases de datos SQL .

Curiosidad - ¿Qué usa Google?

Google es un gran ejemplo de empresa que entiende sus objetivos y puede elegir la mejor opción para sus necesidades entre una base de datos `SQL` y una `NoSQL`.

Dado que trabaja con conjuntos de **datos masivos**, ha optado por trabajar con una base de datos `NoSQL`. La empresa utiliza `Bigtable`, que es una base de datos de creación propia.

`Bigtable` es una tabla poco poblada que puede escalar hasta miles de millones de filas y miles de columnas, lo que te permite almacenar petabytes de datos. Se indexa solo un valor de cada fila; este es conocido como la clave de fila. Admite una capacidad de procesamiento de lectura y escritura alta con baja latencia, y es una fuente de datos ideal para las operaciones de **MapReduce**.

Cuando usamos SQL vs NoSQL

Las bases de datos `SQL` son ideales cuando:

- Se necesita un alto nivel de seguridad e integridad de los datos.
- Tiene datos muy estructurados que no cambian con regularidad.
- Necesita realizar solicitudes ad hoc u otras consultas complejas
- No necesita escalar horizontalmente.
- Soporta sistemas transaccionales, como aplicaciones financieras o contables.

Cuando usamos SQL vs NoSQL

Es mejor utilizar bases de datos **NoSQL** cuando:

- No requieres un alto nivel de seguridad e integridad de los datos
- Tiene muchos datos no estructurados o semiestructurados.
- Tiene datos que cambian con frecuencia y necesita la flexibilidad de un esquema dinámico.
- Quiere agilizar el desarrollo y ahorrar dinero utilizando un enfoque estructurado.
- Necesita escalar horizontalmente.

Introducción

Características de bases de datos NoSQL

Bases de datos relacionales

Cumplen con el modelo relacional:

- Normalización

Es el tipo de base de datos más utilizado.

Utilizan el lenguaje `SQL` (*Structured Query Language*) para consultar y manipular datos.

Los datos son almacenados en tablas:

- Es posible "unir" diferentes tablas para recuperar información



Bases de datos **documentales**

1. **Modelo de documento:** información es almacenada en documentos.
2. **Datos no estructurados:** información semi-estructurada (Sin esquema fijo).
3. **Escalabilidad:** en vertical (máquina más potente) y horizontal (más máquinas).
4. **Acceso flexible:** Muy eficientes para la manipulación de datos.
5. **Replicación y distribución:** replican datos y distribuyen sus nodos geográficamente.



Aconsejan duplicar información:

- Mejora el rendimiento de las consultas

Consultas muy limitadas.

Bases de datos **clave-valor**

Almacena toda la información en pares `<clave, valor>`.

- La clave es única.
- El valor puede ser cualquier objeto.
- Ejemplo:
 - Clave: `aa0000`
 - Valor: `nombre = "Juan"; apellidos = "García Torres"`

1. **Escalabilidad:** pueden manejar grandes cantidades de datos y usuarios concurrentes.
2. **Simplicidad:** la estructura de clave-valor es sencilla.
3. **Flexibilidad:** no requieren una estructura fija de datos.
4. **Alta velocidad:** las operaciones de lectura y escritura son rápidas.
5. **Altamente divisibles** (suelen almacenarse en memoria)

Limitaciones como la falta de capacidad de relacionar datos y la dificultad para realizar consultas complejas.



Bases de datos **columnares**

1. Orientadas a columnas:

- Optimizadas para la completa recuperación de datos de columnas de datos (*analítica de datos*).

2. **Escalabilidad horizontal**: escalabilidad lineal y orientadas a ser distribuidas.

3. **Análisis de datos de alta velocidad**: están optimizadas para el análisis de datos y proporcionan velocidades de procesamiento muy rápidas.

4. **Compresión de datos**: reduce el tamaño de los datos y aumenta la eficiencia.

5. **Eficiencia en la recuperación de datos**: muy eficientes en la recuperación de datos en función de columnas específicas.

6. **No se basan en un esquema fijo**: no están limitadas por un esquema fijo y capacidad de adaptarse a los cambios en los datos.

7. **Simplicidad en la gestión**: más simples de gestionar, lo que reduce la complejidad y los costos de mantenimiento.



Bases de datos **orientadas a grafos**

1. **Modelado de relaciones:** representan la información mediante un grafo donde
 - Nodos: entidades
 - Aristas: relaciones
2. **Completamente normalizadas:** No duplican información
3. **Análisis de redes:** adecuadas para analizar y visualizar grandes redes de relaciones.
4. **Escalabilidad:** pueden manejar grandes cantidades de datos y relaciones.
5. **Consultas potentes:** amplia variedad de consultas para acceder a los datos de manera eficiente y un lenguaje de consultas complejo.



Limitaciones como la falta de capacidad para realizar cálculos matemáticos complejos y la dificultad para manejar grandes cantidades de nodos y relaciones en tiempo real.