

# Bases de datos NoSQL

## Aparición de tecnología NoSQL

- ▶ **Años 80:** Mayoría de sistemas de información almacenan sus datos en **gestores relacionales**.
  - ▶ Aplicaciones de banca
  - ▶ CRM
  - ▶ ERP
  - ▶ .com (90's), sist. sanitarios y académicos

**RDBMS** se caracterizan por:

- ▶ Modelo de datos simple basado en **tablas y relaciones entre tablas**.
- ▶ Herramientas para garantizar la integridad de datos y consistencia de la información: **(ACID)**
  - Atomicity:** Todo o nada
  - Consistency:** Coherencia
  - Isolation:** Serialización transacciones
  - Durability:** Cambios permanentes
- ▶ Lenguaje de consulta estándar, simple y potente.
- ▶ Utilidades para **acceso, manipulación y privacidad** de datos.
- ▶ Utilidades para **auditoría y recuperación** de datos.
- ▶ Garantizar independencia esquema lógico y físico.
- ▶ Facebook, Twitter, Ebay → Tecnología **relacional** no ofrecía el nivel de **escalabilidad ni rendimiento** adecuado

▶ Avances en virtualización → Construcción **nodos computación en nube (cloud computing)** que a su vez requería **nodos almacenamiento (cloud storage)**

▶ Cambios en **arquitectura y requisitos** de las aplicaciones.

## BIG DATA

→ **3 V's (2013)**

1. Volume
2. Velocity
3. Variety → Datos estructurados y no estructurados.

→ **7 V's (2016)**

4. Variability → Significado cambia
5. Veracity
6. Visualisation
7. Value → Toma de decisiones

## Cuellos de botella en RDBMS

- **Joins** en caso de aumento de datos y su complejidad.
- Manejo de concurrencia deja de ser útil por sistemas distribuidos.
- Interfaces CLI ya no son eficientes.

## El término NoSQL

- **NO SQL (Not only SQL)** → Describe un conjunto de B.D. que se diferencian de RDBMS en los sigtes. aspectos:

- ▶ Esquema **prescindible**
- ▶ **Desnormalización**
- ▶ Escalar **horizontalmente**
- ▶ No garantizan **ACID**

**Escalar horizontalmente:** Utilizar varias máquinas de menor costo conectadas entre sí. Así, puedo tener **distribuidos** los datos. Mayor volumen de datos y velocidad.

- **NoSQL y RDBMS** son tipos de **Almacenamiento Estructurado**.
- **Principal diferencia** → Cómo guardar los datos.



- **Caso de factura:** En **NoSQL** se guarda como unidad, sin separar los datos.

## Principios

1. Control **ACID** no es importante
2. Los **JOIN's** tampoco lo son
3. Algunos elementos **relacionales** son necesarios y aconsejables (**keys**)
4. Gran capacidad de **escalabilidad** y **replicación** en múltiples servidores (En **RDBMS** se puede hacer pero son poco flexibles)

## Arquitectura

- **Consistencia débil**
- **Arquitectura distribuida** → Varios servidores.
- **Estructuras de datos simples** → Arrays asociativos o clave-valor.
- Consultas por **Key** o **índice**
- Consultas complejas mediante una infraestructura de procesamiento externo → **MapReduce**

## Teorema CAP

"Es imposible para un sistema computacional **distribuido** ofrecer simultáneamente las sgtes tres garantías"

1. **Consistencia:** Mismos datos al mismo tiempo
2. **Disponibilidad**
3. **Tolerancia a la partición (Partition):** Sistema continúa funcionando a pesar de fallos.

## Modelo BASE

- **Basically Available** → Operativo la mayoría del tiempo
- **Soft state** → Datos en dif. réplicas no tienen que ser mutuamente consistentes a todo momento

- **Eventually Consistent** → Se asegura consistencia solo después de cierto tiempo.

## Aplicaciones

- Volúmenes grandes de datos
- Frecuencia alta de **accesos** de lectura y escritura.
- Cambios frecuentes en esquemas de datos.
- No requieren consistencia **ACID**.

## Tipos de bases de datos NoSQL

### 1. BD clave-valor

- Única **clave (key)** en un **hasn-map**
- Valor asociado con la key es **opaco**, un blob.
- Escalabilidad horizontal a través de **sharding**, tomar pedazos de datos y distribuirlos en shards. Con clave-valor los deposito y busco.
- Amazon DynamoDB

### 2. BD documental

- Datos se almacenan como **documentos** que contienen sus propios **metadatos**.
- Similares a clave-valor pero almacenado en valor es visible.

### → Sharding

- MongoDB

### 3. Familia de columnas

- Familias de columnas: Cada fila tiene una key y una o varias columnas.
- Cassandra

### 4. BD Grafos

- Datos se almacenan como **nodos** y **enlaces**.
- Escalabilidad vertical.
- Neo4j