

# BD NoSQL

BDDA

UT7

Laurent/Rubén



# Introducción

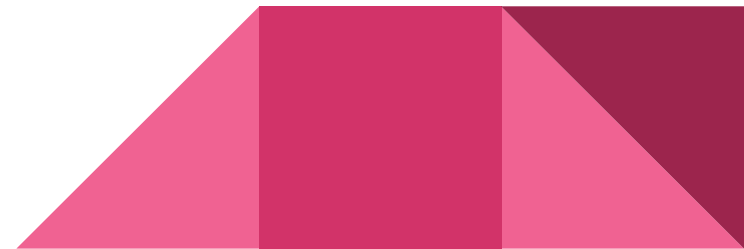
## UT 7 BD NoSQL

Definición y características

CAP

Motivos elección NoSQL

Tipos de BD NoSQL

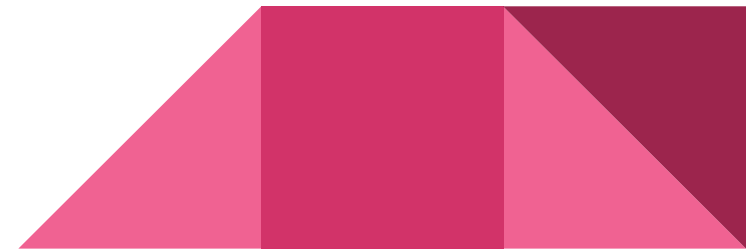


# Definición

Las bases de datos **NoSQL** (Not Only SQL) son sistemas de gestión de bases de datos diseñados para manejar grandes volúmenes de información de manera flexible, escalable y distribuida, sin requerir una estructura rígida como en las bases de datos relacionales.

Las bases de datos NoSQL nacen de la necesidad de:

- Simplicidad en los diseños
- Escalado horizontal
- Mayor control en la disponibilidad



# Características

## Modelo de datos flexible:

- No requieren una estructura fija de esquemas.
- Pueden almacenar datos en distintos formatos como documentos JSON, clave-valor, columnas o grafos.

## Alta escalabilidad:

- Diseñadas para manejar grandes cantidades de datos distribuidos en múltiples servidores.
- Soportan escalabilidad horizontal (añadir más servidores en lugar de mejorar uno solo).

## Rendimiento optimizado:

- Diseñadas para consultas rápidas y eficientes, sin necesidad de hacer JOINS como en SQL.
- Optimizadas para casos de uso específicos, como lecturas rápidas o escrituras masivas.

# Características

Alta disponibilidad y tolerancia a fallos:

- Implementan replicación de datos y particionamiento para evitar pérdida de información.
- Funcionan en arquitecturas distribuidas con múltiples nodos.

Consistencia flexible (Teorema CAP, Consistencia, Disponibilidad y Particionado, o BASE):

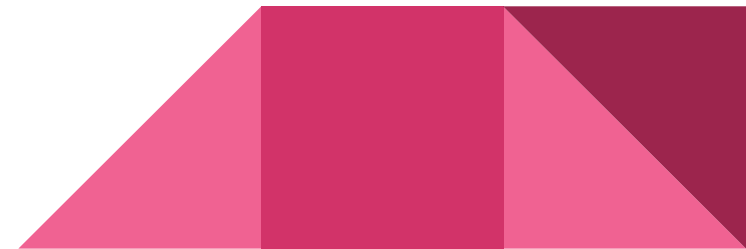
- Dependiendo del tipo de NoSQL, pueden priorizar disponibilidad, consistencia o tolerancia a particiones.
  - a. MongoDB, por ejemplo, sacrifica disponibilidad
  - b. Cassandra, por ejemplo, sacrifica consistencia.
- Algunas bases de datos NoSQL sacrifican consistencia a cambio de rendimiento y escalabilidad.

Optimización para Big Data y aplicaciones web:

- Se utilizan en entornos como redes sociales, comercio electrónico, análisis en tiempo real y almacenamiento de datos en la nube.

# CAP

- Consistencia – todos los nodos ven los mismos datos al mismo tiempo
- Disponibilidad (Availability) – garantiza que cada petición recibe una respuesta acerca de si tuvo éxito o no
- Tolerancia a la partición (Partition) – el sistema continúa funcionando a pesar de la pérdida de mensajes



# ACID VS BASE

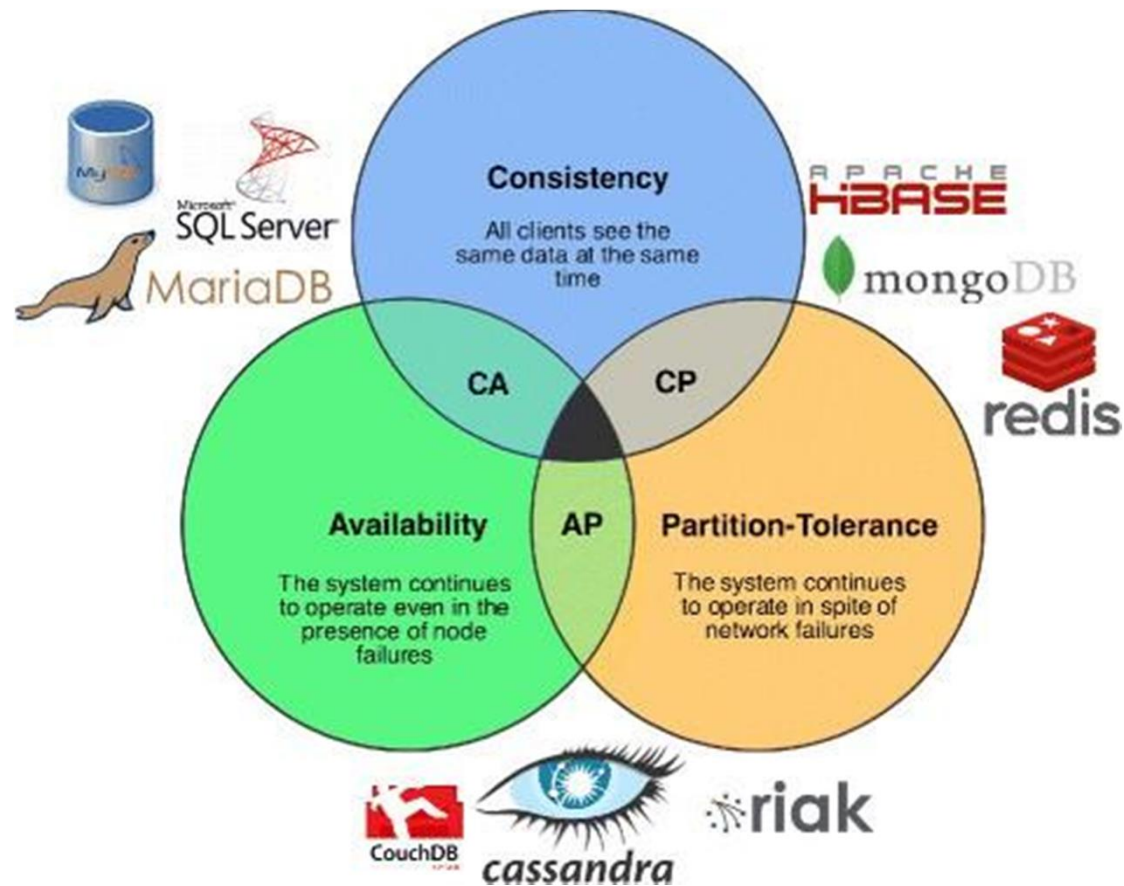
- En el mundo relacional estamos familiarizados con las transacciones ACID, que garantizar la consistencia y estabilidad de las operaciones pero requieren lockings sofisticados:
  - a. ACID = Atomicidad, Consistencia, (Isolation) aislamiento y Durabilidad
- Las BBDD NoSQL siguen el modelo.
  - a. BASE:
    - i. Basic availability: el almacén funciona la mayoría del tiempo incluso ante fallos gracias al almacenamiento distribuido y replicado
    - ii. Soft-sate: los almacenes no tienen porque ser consistentes ni sus réplicas en todo momento.
      - 1. • El programador puede verificar esa consistencia.
    - iii. – Eventual consistency: la consistencia se da eventualmente
- • BASE es una alternativa flexible a ACID para aquellos almacenes de datos que no requieren un adherencia estricta a las transacciones

# Comparación SQL - NoSQL

Característica	SQL	NoSQL
<b><i>Estructura</i></b>	Tablas, filas y columnas	Clave-valor, documentos, grafos, columnas anchas.
<b><i>Esquema</i></b>	Estricto y definido	Flexible y dinámico
<b><i>Escalabilidad</i></b>	Vertical	Horizontal (más servidores)
<b><i>Consistencia</i></b>	Alta (ACID)	Variable (CAP)
<b><i>Uso ideal</i></b>	Aplicaciones transaccionales	Big Data, apps en tiempo real

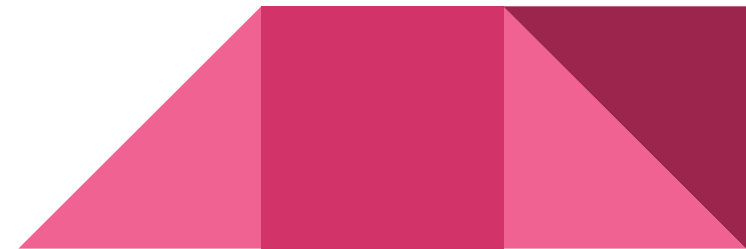


# CAP EJEMPLOS



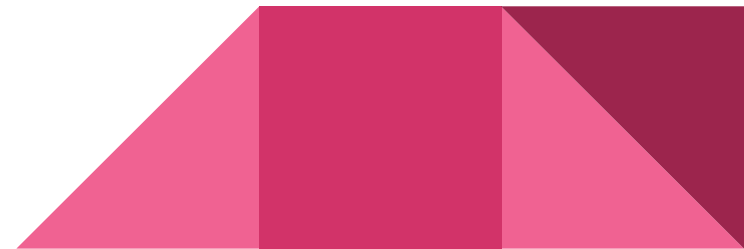
# Razones para escoger NoSQL

- Analítica
- Gran cantidad de escrituras, análisis en bloque
- Escalabilidad
- Tan fácil como añadir un nuevo nodo a la red, bajo coste.
- Redundancia
- Están diseñadas teniendo en cuenta la redundancia
- Rápido desarrollo
- Al ser schema-less o schema on-read son más flexibles que schema on-write
- Flexibilidad en el almacenamiento de datos
- Almacenan todo tipo de datos: texto, imágenes, BLOBs
- Gran rendimiento en consultas sobre datos que no implican relaciones jerárquicas
- Gran rendimiento sobre BBDD desnormalizadas
- Tamaño
- El tamaño del esquema de datos es demasiado grande
- Muchos datos temporales fuera de almacén principal



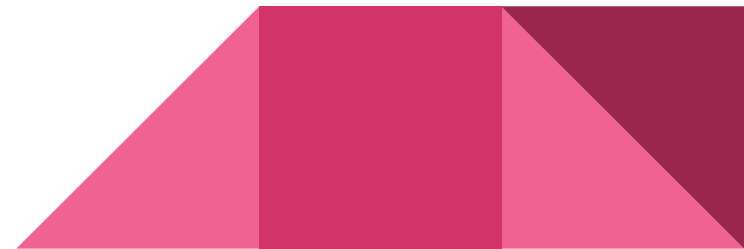
# Casos donde se usa NoSQL

- Redes sociales
- Comercio electrónico y marketplaces
- Big data y análisis en tiempo real
- Internet de las cosas
- Sistemas de recomendaciones (Youtube, spotify...)
- Videojuegos y aplicaciones multijugador
- Blockchain y criptomonedas



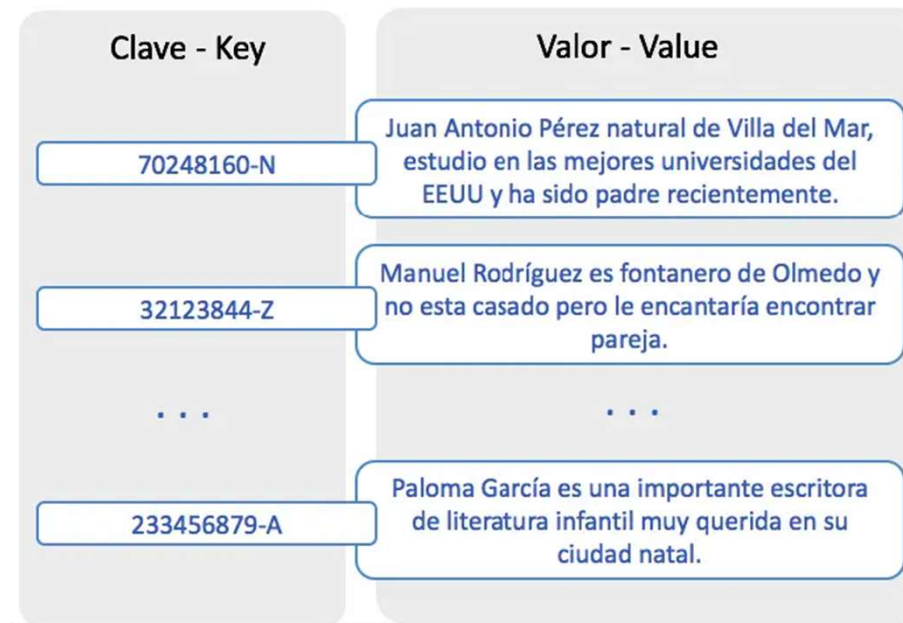
# Tipos de BD NoSQL

- BD Clave-Valor
- Base de datos de documentos
- Base de datos de columnas anchas
- Base de datos de grafos



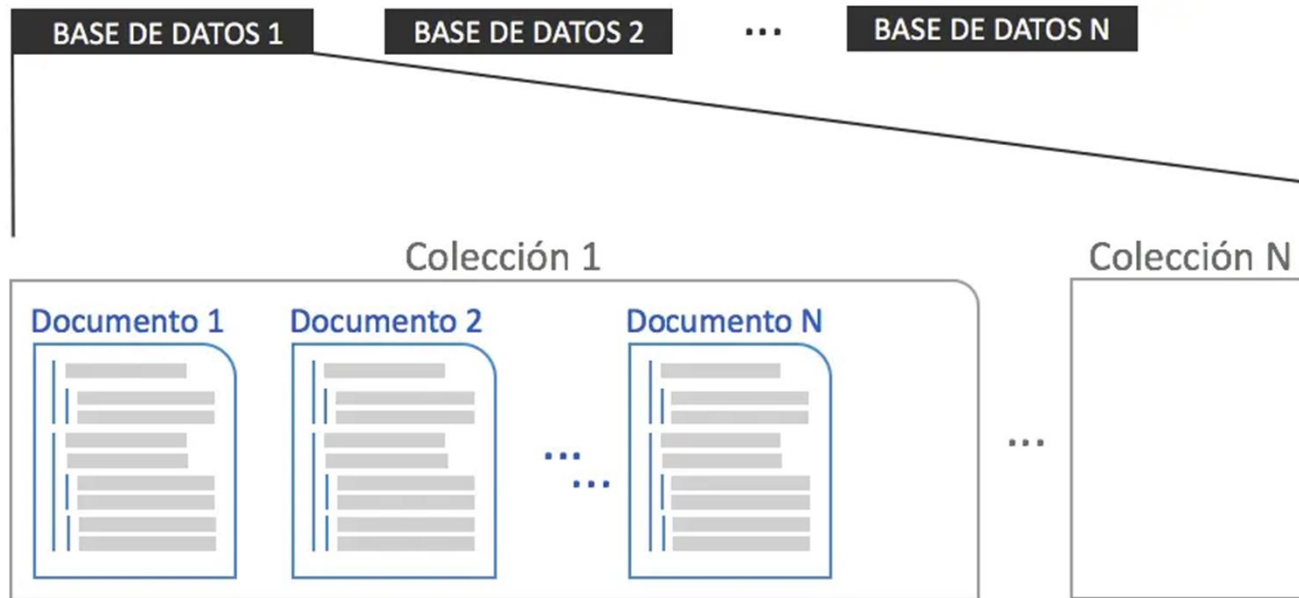
# Clave-Valor

- Las bases de datos **clave-valor** son un tipo de base de datos NoSQL que almacenan datos en un formato simple de **clave** y **valor**, similar a un diccionario o un mapa hash. Son ideales para aplicaciones que requieren accesos rápidos y eficientes a la información. Ej Redis



# Documentos

- Las bases de datos de **documentos** son un tipo de **base de datos NoSQL** que almacenan datos en formatos estructurados como **JSON, BSON o XML**, lo que permite manejar información de manera más flexible que las bases de datos relacionales. Son ideales para aplicaciones que requieren escalabilidad, rendimiento y facilidad de modificación del esquema. Ej MongoDB
- <https://aws.amazon.com/es/nosql/document/>



# Columnas anchas

- Las bases de datos de **columnas anchas** son un tipo de **base de datos NoSQL** diseñadas para almacenar y consultar grandes volúmenes de datos distribuidos en múltiples servidores. Se basan en un modelo de almacenamiento orientado a **columnas en lugar de filas**, lo que permite realizar consultas analíticas eficientes sobre conjuntos de datos masivos. Ej Cassandra

Row-oriented			
ID	Name	Grade	GPA
001	John	Senior	4.00
002	Karen	Freshman	3.67
003	Bill	Junior	3.33

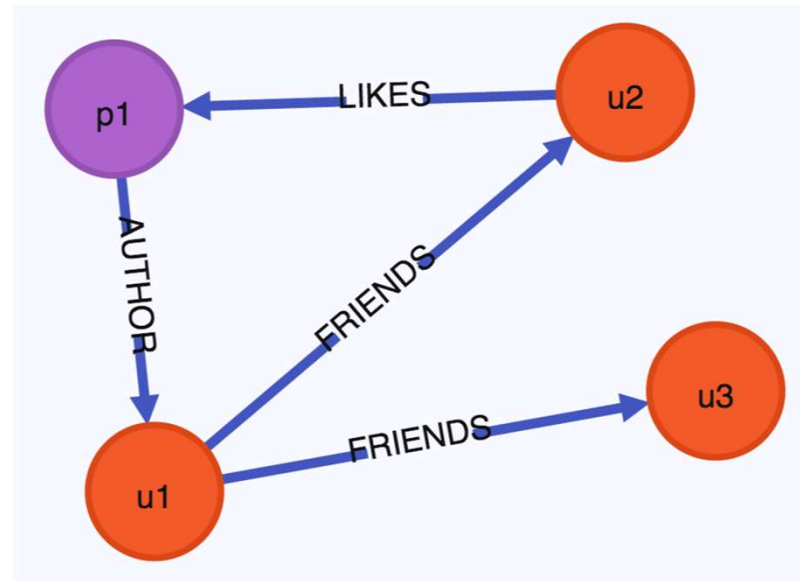
  

Column-oriented			
Name	ID	Grade	ID
John	001	Senior	001
Karen	002	Freshman	002
Bill	003	Junior	003

GPA	ID
4.00	001
3.67	002
3.33	003

# Grafos

- Las **bases de datos de grafos** son un tipo de base de datos NoSQL diseñadas para manejar y almacenar datos altamente conectados. Utilizan la teoría de grafos para representar relaciones y entidades. En este modelo, los datos se almacenan como **nodos** (entidades) y **aristas** (relaciones), lo que permite modelar relaciones complejas de manera eficiente. Son especialmente útiles cuando las relaciones entre los datos son tan importantes como los datos mismos. Ej Neo4j





## Algunas web con información

- <https://phoenixnap.com/kb/nosql-database-types>
- <https://www.acens.com/comunicacion/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>
- <https://ocw.unican.es/pluginfile.php/2444/course/section/2483/Tema%201.%20NoSQL%20introduccio%CC%81n.pdf> (Presentación Universidad de Cantabria)

# Elementos fundamentales BD NoSQL

