

UPIICSA

# BASE DE DATOS

## NoSQL





# INTRODUCCIÓN

# Base de datos NoSQL

El término "NoSQL" se refiere a un enfoque de diseño de bases de datos que difiere del modelo relacional tradicional.



NoSQL, que significa "Not Only SQL" o "No Solo SQL", abarca una variedad de modelos de datos que no siguen el formato tabular y las relaciones estrictas que se encuentran en las bases de datos SQL tradicionales. En lugar de utilizar un esquema fijo y tablas, NoSQL permite la flexibilidad en la estructura de los datos..



**Modelo de Datos No Relacional**





# Base de datos NoSQL

A DJI Mavic 2 drone is shown from a front-facing perspective, slightly angled. The drone is white with grey arms and a black camera module. The camera module has "HASSELBLAD" and "f/2.8 EQV 28mm" printed on it. The drone's arms are extended, and the propellers are visible. The background is a blurred cityscape.

## Escalabilidad Horizontal

Una característica fundamental de las bases de datos NoSQL es su capacidad para escalar horizontalmente. Esto significa que puedes agregar más servidores o nodos a medida que aumenta la carga de trabajo, facilitando la gestión del crecimiento del sistema sin depender únicamente de un servidor más potente.



# Base de datos NoSQL

A DJI Mavic 2 drone is shown from a front-facing perspective, flying against a clear blue sky. The drone is white with grey arms and a black camera module. The camera module has a silver Hasselblad logo and the text "f/2.8 EQV 28mm". The drone's arms are extended, and the propellers are visible. The text "MAVIC 2" is printed on the right arm. The background is a soft, out-of-focus blue sky.

## Rendimiento y Velocidad

NoSQL a menudo se elige por su capacidad para manejar grandes volúmenes de datos y ofrecer un rendimiento rápido. Al eliminar la necesidad de realizar operaciones complejas de unión (join) y transacciones, algunas bases de datos NoSQL pueden ser más eficientes en ciertos casos de uso específicos.



# Base de datos NoSQL



“

Es importante destacar que no es una solución universal y que su elección debe basarse en los requisitos y características específicas del proyecto o aplicación en cuestión.

”

# Porque usar NoSQL

**El uso de bases de datos NoSQL se justifica en diversas situaciones debido a sus características particulares.**





# Porque usar NoSQL

NoSQL permite adaptarse fácilmente a cambios en los requisitos del esquema de datos, ya que no requiere una estructura fija y predefinida. Esto es especialmente útil en entornos donde los requisitos evolucionan rápidamente.

**Flexibilidad en  
el Esquema**



# Porque usar NoSQL

Las bases de datos NoSQL están diseñadas para escalar horizontalmente, lo que significa que puedes agregar más nodos o servidores a medida que la carga de trabajo aumenta. Esto facilita la gestión de grandes volúmenes de datos y el manejo de un mayor tráfico sin depender de un único servidor potente.

**Escalabilidad  
Horizontal**



# Porque usar NoSQL

Algunos modelos de datos, como los documentos JSON o los grafos, se adaptan mejor a ciertos tipos de aplicaciones. Por ejemplo, bases de datos documentales como MongoDB son ideales para almacenar datos semi-estructurados.

**Modelos de  
Datos  
Específicos**





# Porque usar NoSQL

En ciertos casos, las bases de datos NoSQL pueden ofrecer un rendimiento superior en operaciones específicas, como lecturas o escrituras masivas, debido a su diseño optimizado para tareas particulares.

**Rendimiento en  
Lectura/Escritura  
Específico**



# Porque usar NoSQL

Las bases de datos NoSQL a menudo están diseñadas para operar en entornos distribuidos y son capaces de tolerar fallos en nodos individuales. Esto garantiza una mayor disponibilidad y confiabilidad del sistema.

## Datos Distribuidos y Tolerancia a Fallos



# Porque usar NoSQL

Escalar horizontalmente añadiendo nodos suele ser más económico que invertir en un servidor más potente para escalar verticalmente. Esto es particularmente beneficioso en términos de costos en el escenario de crecimiento del sistema.

## Economía de Escala





# Porque usar NoSQL

## Casos de Uso Específicos

Algunos casos de uso, como aplicaciones WEB, análisis de Big Data, y sistemas que manejan datos semi-estructurados o no estructurados, pueden beneficiarse significativamente del uso de bases de datos NoSQL.



# Porque usar NoSQL

**Es importante destacar que la elección entre bases de datos SQL y NoSQL depende de los requisitos específicos del proyecto y del escenario de uso. En muchos casos, una combinación de ambas tecnologías (SQL y NoSQL) puede ser la solución más apropiada, dependiendo de los distintos aspectos y necesidades del sistema.**



# Tipos de Bases de Datos NoSQL



Hay varios tipos de bases de datos NoSQL, cada una diseñada para satisfacer necesidades específicas.

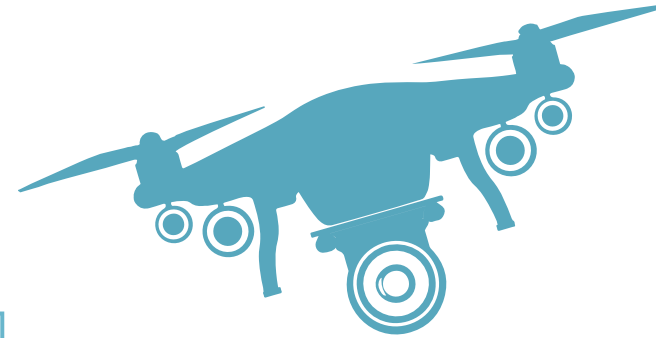


## CLAVE-VALOR

Almacenan datos como pares clave-valor. Ejemplos: Redis, DynamoDB

## DOCUMENTALES.

Almacenan datos en documentos, como JSON o BSON. Ejemplos: MongoDB, CouchDB



## GRAFO

Modelan datos como nodos y relaciones. Ejemplos: Neo4j, ArangoDB.

## COLUMNARES

Almacenan datos en columnas en lugar de filas. Ejemplos: Apache Cassandra, HBase.





# Bases de Datos NoSQL



MongoDB:

**Tipo:** Base de datos de documentos (Document Store).

**Características:** Almacena datos en documentos BSON (formato binario JSON). Es muy utilizado en aplicaciones web y móviles.



Cassandra:



**Tipo:** Base de datos de columna ancha (Wide Column Store).

**Características:** Escalabilidad horizontal, alta disponibilidad y tolerancia a fallos. Es comúnmente utilizado en entornos distribuidos.



# Bases de Datos NoSQL



CouchDB:

**Tipo:** Base de datos de documentos (Document Store).

**Características:** Almacena datos en documentos JSON, soporta consultas mediante MapReduce y es conocido por su capacidad de replicación.



Redis



**Tipo:** Base de datos en memoria (In-Memory Data Store).

**Características:** Almacena datos clave-valor en memoria, lo que proporciona un rendimiento extremadamente rápido. Se utiliza comúnmente como caché y para gestionar sesiones en aplicaciones web.



# Bases de Datos NoSQL



Neo4j:

**Tipo:** Base de datos de grafos (Graph Database).

**Características:** Modela datos como nodos y relaciones, ideal para representar y consultar datos relacionados en redes complejas.



Amazon DynamoDB



**Tipo:** Base de datos de documentos y clave-valor.

**Características:** Servicio de base de datos completamente gestionado por Amazon Web Services (AWS). Es escalable y se utiliza comúnmente en aplicaciones en la nube.





# Bases de Datos NoSQL



Riak

**Tipo:** Base de datos clave-valor distribuida.

**Características:** Escalabilidad horizontal, alta disponibilidad y tolerancia a fallos. Se utiliza en aplicaciones distribuidas y sistemas que requieren alta disponibilidad.



ArangoDB



**Tipo:** Base de datos de documentos y grafo (Multi-model Database).

**Características:** Admite tanto documentos como grafos en un solo sistema, ofreciendo flexibilidad para diferentes modelos de datos.



# Tabla comparativa entre bases de datos SQL y NoSQL



Característica	SQL (Relacional)	NoSQL
Modelo de Datos	Tablas con relaciones (normalización)	Documentos, Clave-Valor, Columnares, Grafos
Esquema	Estructura fija (esquema rígido)	Esquema flexible (sin esquema fijo)
Escalabilidad	Vertical (escalabilidad a través de hardware)	Horizontal (escalabilidad mediante nodos)
Transacciones	ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)	No siempre ACID, puede depender del sistema específico
Lenguaje de Consulta	SQL (Structured Query Language)	Varía según la base de datos específica
Ejemplos Comunes	MySQL, PostgreSQL, Oracle	MongoDB, Cassandra, Redis, Neo4j
Aplicaciones Típicas	Sistemas ERP, CRM, Aplicaciones Transaccionales	Aplicaciones web, Redes sociales, Big Data
Escalabilidad Horizontal	Difícil escalar horizontalmente	Diseñado para escalar horizontalmente
Flexibilidad en el Esquema	Estructura fija requiere cambios en el esquema con cuidado	Esquema flexible, fácil de adaptar a cambios
Ejemplo de Uso	Aplicaciones con relaciones complejas, Sistemas tradicionales	Aplicaciones modernas, Big Data, Proyectos con requisitos cambiantes



# NoSQL



Gaming



Social



IoT



Web



Mobile



Enterprise



Key/value store



Document database



Column family store

# SQL



Web



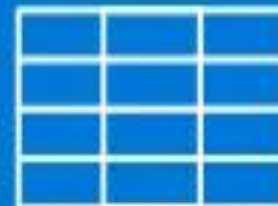
Mobile



Enterprise



Data mart



Relational table storage



Relationships use joins





# **BASES DE DATOS**

## **NoSQL**