UNIVERSIDAD AMERICANA



***Fundamentos de base de datos***

*Investigacion de bases de datos NoSQL*

**Estudiante:**

* Alexander Vado Barkalov.

**Docente:**

- Yader Rodriguéz

Tabla de contenidos

[Introducción 3](#_Toc178343634)

[Características Principales de las Bases de Datos NoSQL: 4](#_Toc178343635)

[Ventajas y desventajas 5](#_Toc178343636)

[Casos de uso típicos: 6](#_Toc178343637)

[Aplicaciones web y móviles con gran cantidad de usuarios: 6](#_Toc178343638)

[Uber - Cassandra 6](#_Toc178343639)

[Big Data 7](#_Toc178343640)

[1. Netflix - MongoDB 7](#_Toc178343641)

[Internet de las Cosas (IoT) 8](#_Toc178343642)

[Smart Homes con Amazon Web Services (AWS IoT) 8](#_Toc178343643)

# Introducción

En la era de la información, los sistemas de almacenamiento y gestión de datos han adquirido una importancia estratégica para las empresas y organizaciones de todo tipo. Tradicionalmente, las bases de datos relacionales han dominado este campo, proporcionando un modelo estructurado y altamente confiable para manejar grandes volúmenes de datos. Sin embargo, con el auge de la web, el big data, las redes sociales y el Internet de las Cosas (IoT), las necesidades de almacenamiento han cambiado drásticamente. Las bases de datos NoSQL, también conocidas como bases de datos no relacionales, han emergido como una alternativa para satisfacer estas nuevas demandas.

Las bases de datos NoSQL permiten gestionar datos no estructurados y semiestructurados con una flexibilidad y escalabilidad que las bases de datos tradicionales no pueden ofrecer. Su capacidad para manejar grandes volúmenes de datos distribuidos de manera eficiente las ha hecho populares en aplicaciones como redes sociales, comercio electrónico, y análisis en tiempo real, donde el rendimiento y la disponibilidad son críticos.

Este informe tiene como objetivo explorar el concepto de bases de datos NoSQL, sus características principales, los diferentes modelos de datos que utilizan, así como sus ventajas y desventajas en comparación con las bases de datos relacionales. Además, se presentarán ejemplos reales de su implementación en sistemas como Netflix, Uber y aplicaciones de IoT, que ilustran el impacto de estas tecnologías en el manejo de datos a gran escala en entornos modernos.

# Características Principales de las Bases de Datos NoSQL:

1. **No utilizan un esquema fijo**: A diferencia de las bases de datos relacionales, las bases de datos NoSQL no requieren que los datos se estructuren en tablas con esquemas rígidos. Pueden almacenar datos en formatos no estructurados o semiestructurados como JSON, BSON, o documentos XML, lo que permite una mayor flexibilidad a la hora de agregar o modificar los datos.

2. **Escalabilidad horizontal**: Las bases de datos NoSQL están diseñadas para escalar horizontalmente, lo que significa que se pueden agregar más servidores para manejar grandes volúmenes de datos y de tráfico. Esto es diferente de las bases de datos relacionales, que suelen escalar verticalmente, añadiendo más potencia a un único servidor.

3. **Modelos de datos variados:** Existen diferentes tipos de bases de datos NoSQL, cada una adaptada a diferentes necesidades:

* Documentales: Almacenan datos en formato de documentos (JSON, BSON, XML). Ejemplo: MongoDB.
* Clave-valor: Almacenan datos como pares de clave-valor, ideales para sistemas con acceso rápido a grandes cantidades de datos. Ejemplo: Redis.
* Columnar: Organizan los datos en columnas en lugar de filas, lo que las hace más eficientes en el procesamiento de grandes conjuntos de datos. Ejemplo: Cassandra.
* Grafos: Utilizan nodos y aristas para representar y almacenar datos que están relacionados entre sí, ideal para analizar relaciones complejas. Ejemplo: Neo4j.

4. **Alto rendimiento para grandes volúmenes de datos:** Las bases de datos NoSQL están diseñadas para manejar grandes cantidades de datos a velocidades rápidas. Son comúnmente utilizadas en aplicaciones que requieren almacenar datos en tiempo real, como redes sociales, análisis de big data, y aplicaciones IoT.

5. **Alta disponibilidad y tolerancia a fallos:** Estas bases de datos suelen distribuirse en clústeres de servidores, lo que permite mantener la disponibilidad del sistema incluso si uno o más nodos fallan.

# **Ventajas y desventajas**

**Ventajas**:

- Flexibilidad en el manejo de datos no estructurados o semiestructurados.

- Escalabilidad horizontal, facilitando el crecimiento sin una reconfiguración masiva.

- Optimización para lecturas y escrituras rápidas.

- Adaptabilidad a aplicaciones modernas y distribuibles.

**Desventajas**:

- La falta de esquemas rígidos puede generar inconsistencias en los datos.

- No tienen soporte nativo para transacciones complejas (aunque algunas bases NoSQL han mejorado en este aspecto).

- A menudo, la falta de estandarización entre diferentes proveedores.

Este tipo de bases de datos se ha popularizado en la última década debido a la expansión de la web, el big data y la necesidad de sistemas más flexibles y rápidos para manejar datos no estructurados. A continuación se mencionara cada uno.

# Casos de uso típicos:

## **Aplicaciones web y móviles con gran cantidad de usuarios:**

Las bases de datos NoSQL permiten gestionar el gran volumen de datos generados por millones de usuarios, especialmente en aplicaciones con datos en tiempo real, como redes sociales, plataformas de streaming o comercio electrónico.

### ****Uber - Cassandra****

Uber utiliza **Apache Cassandra**, una base de datos NoSQL orientada a columnas, para gestionar grandes volúmenes de datos en tiempo real, como las ubicaciones de los conductores, los detalles de los viajes y las tarifas dinámicas. Dado que Uber opera a nivel global, necesita un sistema de almacenamiento de datos que sea altamente escalable y disponible en todo momento. Cassandra permite que los datos se distribuyan en múltiples centros de datos en todo el mundo, garantizando que la plataforma de Uber sea rápida y esté disponible incluso durante picos de demanda, como en eventos masivos o festividades.

## Big Data

Su capacidad para manejar grandes volúmenes de datos no estructurados es ideal para aplicaciones de análisis de datos y Big Data, como sistemas de recomendación, análisis de registros de usuarios o predicciones.

### 1. Netflix - MongoDB

Netflix utiliza **MongoDB**, una base de datos documental NoSQL, para almacenar y analizar grandes volúmenes de datos de comportamiento de los usuarios en tiempo real. Estos datos incluyen el historial de visualizaciones, las búsquedas, las preferencias y las interacciones dentro de la plataforma. Gracias a MongoDB, Netflix puede personalizar la experiencia de cada usuario recomendando películas y series basadas en su historial y comportamiento. Además, MongoDB permite a Netflix escalar rápidamente para gestionar su vasta base de usuarios en todo el mundo.

## Internet de las Cosas (IoT)

Dado que los dispositivos IoT generan una gran cantidad de datos a gran velocidad, las bases de datos NoSQL son capaces de almacenar y procesar esta información eficientemente.

### ****Smart Homes con Amazon Web Services (AWS IoT)****

Un ejemplo claro en el ámbito del **Internet de las Cosas (IoT)** es el uso de **AWS IoT** junto con **Amazon DynamoDB**, una base de datos NoSQL basada en el modelo de clave-valor, para gestionar dispositivos conectados en hogares inteligentes (smart homes).

**Cómo funciona:** Los dispositivos conectados en una casa inteligente, como termostatos, luces, cámaras de seguridad, sensores de movimiento y electrodomésticos, generan continuamente datos. Estos dispositivos necesitan comunicarse entre sí y con servidores centrales para realizar acciones en tiempo real, como ajustar la temperatura, encender/apagar luces, o alertar al usuario sobre movimientos detectados.

**AWS IoT**, junto con **DynamoDB**, gestiona el almacenamiento y procesamiento de estos grandes volúmenes de datos generados por los sensores y dispositivos del hogar. Los datos de los dispositivos IoT se almacenan de manera distribuida en **DynamoDB**, que es capaz de escalar automáticamente y procesar estos datos en tiempo real. Además, la arquitectura distribuida de DynamoDB garantiza alta disponibilidad, lo que es crucial para sistemas de IoT que necesitan estar siempre en funcionamiento.