

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Bases de Datos

Ing. Fernando Arreola Franco

Mariana Daniela Hernández Pérez

Grupo 1

Tarea 1

Generación 2026-1

1. Modelo orientado a objetos

La tecnología de bases de datos vive un momento de lenta transición del modelo relacional a otros modelos. Entre éstos se encuentra el multidimensional para sistemas OLAP, el semiestructurado para bases de datos XML de intercambio electrónico de información, el modelo dimensional para creación de Data Warehouse y el orientado a objetos. En una base de datos orientada a objetos, los componentes se almacenan como objetos y no como datos, tal y como hace una base relacional, cuya representación son las tablas.

Algo importante que debemos resaltar es que hoy en día, las empresas siguen utilizando los manejadores de bases de datos relacionales y no se sabe aún si serán suplantadas por completo, ni cuándo.

Los manejadores de bases de datos orientados a objetos deben tomar en cuenta las siguientes operaciones:

- Ser capaces de definir sus propios tipos de datos.
- El tamaño de los datos puede ser muy grande.
- La duración de las transacciones puede ser muy larga.
- Recuperar rápidamente objetos complejos.
- Lenguajes de consulta de objetos, un ejemplo es OQL (Object Query Language).
- Mecanismos de seguridad basados en la noción de objeto.
- Funciones para definir reglas deductivas.

2. Bases de Datos NoSQL

El uso de las bases de datos NoSQL se hace presente hoy en día de forma frecuente. Las aplicaciones y los softwares que más utilizamos las incluyen de forma directa en su arquitectura. Algún tiempo atrás fue toda una revolución la aparición de las BBDD construidas con SQL. Han cumplido y lo han hecho bien, pero las necesidades del mundo digital avanzan a pasos agigantados y lamentablemente las BBDD SQL tienen algunas limitaciones.

Esta es una de las razones de la aparición de las Bases de datos NoSQL. La evolución tecnológica de la sociedad y la participación masiva de usuarios que producen y alojan contenido cada segundo, exigía una forma más flexible de almacenar, ordenar y captar mayor cantidad de datos. Eso es lo que ha ayudado a solucionar el NoSQL.

2.1. ¿Qué es NoSQL?

El término NoSQL se refiere a la denominación en inglés *Not Only SQL*. Plantea modelos de datos específicos de esquemas flexibles que se adaptan a los requisitos de las aplicaciones más modernas. Tienen un conjunto increíble de características y varios modelos que descubriremos más adelante.

3. Modelo NoSQL Clave-Valor

El uso de las bases de datos NoSQL se hace presente hoy en día de forma frecuente. Las aplicaciones y los softwares que más utilizamos las incluyen de forma directa en su arquitectura. Algún tiempo atrás fue toda una revolución la aparición de las BBDD construidas con SQL. Han cumplido y lo han hecho bien, pero las necesidades del mundo digital avanzan a pasos agigantados y lamentablemente las BBDD SQL tienen algunas limitaciones.

Una de las soluciones a estas limitaciones son las bases de datos NoSQL, y dentro de ellas, el **modelo clave-valor** es uno de los más simples y eficientes. Este modelo almacena la información como un conjunto de pares *clave* \rightarrow *valor*, donde cada clave es única y se asocia directamente a un valor. La estructura es similar a un diccionario o mapa en lenguajes de programación.

El modelo clave-valor destaca por su **velocidad** y **eficiencia** en operaciones de lectura y escritura, ya que permite acceder a los datos directamente a través de su clave sin necesidad de realizar consultas complejas. Es ideal para casos de uso como:

- Almacenamiento de sesiones de usuario.
- Cachés en memoria para aplicaciones web.
- Configuraciones rápidas y persistentes.
- Almacenamiento de datos en tiempo real.

3.1. Ventajas

- Alto rendimiento y baja latencia.
- Escalabilidad horizontal sencilla.
- Simplicidad en la estructura de datos.

3.2. Desventajas

- No es adecuado para consultas complejas.
- Falta de relaciones entre datos.
- Puede requerir más lógica en la aplicación para manejar la información.

3.3. Ejemplos de bases de datos clave-valor

- Redis.
- Amazon DynamoDB.
- Riak.
- Memcached.

4. Modelo NoSQL Documental

El modelo NoSQL documental es uno de los tipos más utilizados dentro de las bases de datos NoSQL. En este enfoque, la información se almacena en documentos, normalmente en formato **JSON** o **BSON**, que contienen pares de clave-valor, arreglos, y estructuras anidadas. Cada documento es una unidad independiente que almacena todos los datos relacionados con una entidad, lo que permite una gran flexibilidad y escalabilidad.

A diferencia de las bases de datos relacionales, no existe un esquema fijo; cada documento puede tener un conjunto diferente de campos. Esto permite adaptar la estructura de los datos a las necesidades específicas de la aplicación sin requerir migraciones complejas.

4.1. Ventajas

- Alta flexibilidad: no requiere un esquema fijo.
- Excelente rendimiento para consultas sobre documentos completos.
- Ideal para aplicaciones con datos semiestructurados o no estructurados.
- Fácil integración con aplicaciones web modernas.

4.2. Desventajas

- No es óptimo para relaciones complejas entre datos.
- Puede generar duplicación de información.
- Algunas operaciones de agregación pueden ser costosas.

4.3. Ejemplos de bases de datos documentales

- MongoDB.
- CouchDB.
- Amazon DocumentDB.
- RethinkDB.

4.4. Casos de uso típicos

- Sistemas de gestión de contenido (CMS).
- Aplicaciones web que manejan datos dinámicos.
- Almacenamiento de catálogos de productos.
- Aplicaciones con datos semiestructurados que cambian con frecuencia.

5. Modelo NoSQL de Grafos

El modelo NoSQL de grafos es un tipo de base de datos diseñado para representar y consultar relaciones complejas entre datos. En este modelo, la información se almacena en forma de **nodos** (entidades), **aristas** (relaciones) y **propiedades** (atributos de nodos o aristas). Es especialmente útil para aplicaciones donde las conexiones y las relaciones entre datos son tan importantes como los datos mismos.

A diferencia de los modelos relacionales o documentales, las bases de datos de grafos permiten recorrer relaciones de forma rápida y eficiente, incluso cuando el volumen de datos y el número de conexiones es muy alto.

5.1. Ventajas

- Excelente rendimiento en consultas de relaciones complejas.
- Modelo intuitivo para representar redes y conexiones.
- Flexibilidad para cambios en la estructura de los datos.

5.2. Desventajas

- No es la mejor opción para datos tabulares simples.
- Puede tener un costo de aprendizaje más alto para usuarios acostumbrados a SQL.
- Escalabilidad horizontal más compleja que en otros modelos NoSQL.

5.3. Ejemplos de bases de datos de grafos

- Neo4j.
- ArangoDB.
- Amazon Neptune.
- OrientDB.

5.4. Casos de uso típicos

- Redes sociales (representación de usuarios y sus conexiones).
- Sistemas de recomendación.
- Análisis de rutas y mapas.
- Detección de fraudes mediante patrones de conexión.

6. Modelo NoSQL Columnar

El modelo NoSQL columnar es un tipo de base de datos diseñado para almacenar y gestionar datos organizados por columnas en lugar de filas. En este enfoque, la información se agrupa por **familias de columnas**, lo que permite un acceso muy rápido a grandes volúmenes de datos cuando se necesitan solo ciertos atributos. Este modelo es especialmente eficiente en consultas analíticas y operaciones de lectura sobre conjuntos de datos masivos.

A diferencia de las bases de datos relacionales, que almacenan todos los valores de una fila juntos, las bases de datos columnares almacenan todos los valores de una misma columna juntos en memoria o en disco, optimizando la compresión y el rendimiento en consultas específicas.

6.1. Ventajas

- Alto rendimiento en consultas de lectura sobre grandes volúmenes de datos.
- Compresión de datos más eficiente.
- Ideal para análisis y procesamiento masivo de información.

6.2. Desventajas

- No es la mejor opción para operaciones transaccionales frecuentes.
- Estructura menos intuitiva para aplicaciones que necesitan registros completos.
- Curva de aprendizaje más pronunciada.

6.3. Ejemplos de bases de datos columnares

- Apache Cassandra.
- HBase.
- ScyllaDB.
- Amazon Keyspaces.

6.4. Casos de uso típicos

- Almacenamiento y consulta de datos de series temporales.
- Sistemas de analítica empresarial (Business Intelligence).
- Procesamiento de grandes volúmenes de datos en tiempo real.

Referencias

- [1] A. Lakshman and P. Malik, “Cassandra: A Decentralized Structured Storage System,” *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, Apr. 2010.
- [2] K. Chodorow, *MongoDB: The Definitive Guide*, 3rd ed. Sebastopol, CA, USA: O’Reilly Media, 2019.
- [3] I. Robinson, J. Webber, and E. Eifrem, *Graph Databases*, 2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2015.
- [4] S. Sanfilippo, Redis Documentation, Redis, 2023. [Online]. Available: <https://redis.io/documentation>. [Accessed: 14-Aug-2025].
- [5] A. George, “HBase: The Definitive Guide,” Sebastopol, CA, USA: O’Reilly Media, 2011.
- [6] Amazon Web Services, “Amazon DynamoDB Documentation,” AWS, 2024. [Online]. Available: <https://aws.amazon.com/dynamodb/>. [Accessed: 14-Aug-2025].
- [7] P. J. Sadalage and M. Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, Boston, MA, USA: Addison-Wesley, 2012.
- [8] “Modelo Orientado a Objetos”. 403 Forbidden. Accedido el 16 de agosto de 2025. [En línea]. Disponible: https://repositorio-uapa.cuaed.unam.mx/repositorio/moodle/pluginfile.php/2731/mod_resource/content/1/UAPA-Modelo-Orientado-Objetos/index.html
- [9] “Bases de Datos NoSQL — Qué son, marcas, tipos y ventajas”. GraphEverywhere. Accedido el 16 de agosto de 2025. [En línea]. Disponible: <https://www.grapheverywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/>