



Universidad Nacional
Autónoma de México
Facultad de Ingeniería



Bases de Datos

Semestre: 2026-2

Tarea 1

Profesor: Arreola

Alumno: Parra Bello Carlos Enrique

Número de cuenta: 321135344

Grupo: 01

El modelo de bases de datos orientado a objetos se fundamenta en los principios de la programación orientada a objetos. En este enfoque, la información se almacena como **objetos**, los cuales encapsulan tanto los datos (atributos) como las operaciones que pueden realizarse sobre ellos (métodos). Estos objetos se agrupan en **clases**, permitiendo el uso de conceptos como herencia, encapsulamiento y polimorfismo.

Este modelo busca reducir la brecha existente entre las bases de datos y los lenguajes de programación orientados a objetos, evitando la necesidad de transformar los datos a estructuras relacionales.

Ventajas

- Permite una integración directa con lenguajes orientados a objetos.
- Facilita el manejo de datos complejos y estructurados.
- Promueve la reutilización de código mediante herencia.
- Reduce la conversión entre datos almacenados y objetos en memoria.

Desventajas

- Carece de una estandarización amplia.
- Presenta una curva de aprendizaje elevada.
- Tiene menor adopción comercial en comparación con bases de datos relacionales.
- No es la opción más eficiente para sistemas altamente distribuidos.

Las bases de datos **NoSQL** (Not Only SQL) surgieron como respuesta a las limitaciones de los sistemas relacionales tradicionales frente a escenarios que requieren alta escalabilidad, flexibilidad y rendimiento. Estas bases de datos no siguen un esquema rígido y están diseñadas para trabajar con grandes volúmenes de datos distribuidos.

Dentro de NoSQL existen distintos submodelos, cada uno orientado a un tipo específico de problema.

2.1 Modelo Clave-Valor

Descripción

Este modelo almacena la información mediante pares **clave-valor**, donde cada clave es única y permite acceder directamente al valor asociado.

Ventajas

- Alta velocidad de acceso.
- Diseño simple y eficiente.
- Excelente escalabilidad horizontal.

Desventajas

- No permite realizar consultas complejas.
- No soporta relaciones entre datos.

Casos de uso

Se utiliza comúnmente para almacenar sesiones de usuario, datos en caché, tokens de autenticación y configuraciones de sistemas.

2.2 Modelo Documental

Descripción

El modelo documental almacena la información en **documentos**, generalmente en formatos como JSON o BSON. Cada documento puede tener una estructura diferente, lo que proporciona gran flexibilidad.

Ventajas

- No requiere un esquema fijo.
- Fácil adaptación a cambios en los datos.
- Ideal para aplicaciones web y servicios REST.

Desventajas

- Puede generar duplicación de datos.
- Manejo limitado de relaciones complejas.
- Menor consistencia en estructuras extensas.

Casos de uso

Es ampliamente utilizado en aplicaciones web, sistemas de gestión de contenido (CMS), plataformas de comercio electrónico y APIs.

2.3 Modelo de Grafos

Descripción

Este modelo representa los datos mediante **nodos y relaciones**, lo que permite modelar de forma natural conexiones complejas entre entidades.

Ventajas

- Excelente rendimiento en consultas basadas en relaciones.
- Modelo intuitivo para representar redes.
- Muy eficiente para análisis relacional.
- Búsquedas sencillas

Desventajas

- Escalabilidad horizontal más compleja.
- No es ideal para datos simples.
- Requiere conocimientos especializados.
- Lenguaje de consulta

Casos de uso

Se emplea en redes sociales, sistemas de recomendación, detección de fraudes y análisis de relaciones complejas.

2.4 Modelo Columnar (Familias de Columnas)

Descripción

El modelo columnar organiza los datos en **familias de columnas**, en lugar de filas, lo que permite un acceso más eficiente a grandes volúmenes de información.

Ventajas

- Alta disponibilidad y escalabilidad.
- Excelente rendimiento en análisis de grandes datos.
- Diseñado para sistemas distribuidos.

Desventajas

- Diseño y mantenimiento complejos.
- Menor flexibilidad en consultas.
- No recomendado para transacciones pequeñas.
- Poco eficientes en modelos transaccionales

Casos de uso

Es utilizado en sistemas de Big Data, análisis de registros (logs), aplicaciones IoT y plataformas de monitoreo.

El modelo orientado a objetos almacena la información como objetos, de forma similar a los lenguajes de programación orientados a objetos

Modelo	Estructura	Ventaja principal	Caso típico
Orientado a objetos	Objetos	Datos complejos	Simulación
Clave–Valor	Par clave-valor	Velocidad	Caché
Documental	Documentos	Flexibilidad	Web
Grafos	Nodos y relaciones	Relaciones	Redes sociales
Columnar	Familias de columnas	Big Data	Analítica

IBM, “**NoSQL Databases Explained**,” IBM Cloud Learn Hub, 2023. [En línea]. Disponible en: <https://www.ibm.com/cloud/learn/nosql-databases>

MongoDB Inc., “**What is a Document Database?**,” MongoDB Documentation. [En línea]. Disponible en: <https://www.mongodb.com/document-databases>