



Isaac Jareth Campos Cortés

Investigación de conceptos:

1) Modelo orientado a objetos:

Una base de datos orientada a objetos (BDOO) almacena sus datos de la misma manera que en el paradigma de programación orientado a objetos (POO), es decir, como objetos con datos y métodos para cada uno.[1][2]

Esta característica permite a este modelo manejar de manera más sencilla datos con estructuras complejas, además facilitan la integración con otros lenguajes de programación orientados a objetos.[1][2]

Otra ventaja del hecho de que los datos sean representados como objetos es que conservan características útiles del paradigma: herencia, polimorfismo, persistencia y encapsulación. Específicamente la herencia y el polimorfismo ayudan a la flexibilidad y escalabilidad de este tipo de bases de datos. Además, facilita la representación de relaciones entre los datos.[1][2]

Las principales desventajas de este modelo de base de datos incluyen: una pronunciada curva de aprendizaje si acostumbra a trabajar con bases de datos relacionales, un mayor consumo de recursos a comparación de otros modelos, integración compleja con otro tipo de bases de datos y una adopción reducida a nivel empresarial.[1][2]

Algunos casos en los que se utilizan con mayor frecuencia este tipo de bases de datos son:

- Diseño y manufactura apoyados por computadora: Permiten almacenar diseños geométricos y estructurales como objetos y por ende facilitan manejarlos y editarlos de manera sencilla.
- Telecomunicaciones: Muy útil para representar relaciones entre distintos datos importantes.
- Investigación científica: Permiten almacenar grandes conjuntos de datos de manera sencilla de organizar y manejar para su estudio.

2) Modelos NoSQL:

NoSQL significa Not only SQL, estos modelos de bases de datos buscan una alternativa a la forma tradicional del modelo relacional. [3][4]

Estos modelos buscan brindar ciertas ventajas para ciertos casos de uso en los cuales las bases de datos relacionales no siempre son la mejor opción.[3][4]

A continuación se presentan cuatro tipos de modelos NoSQL:

Clave-valor

Este tipo de bases de datos emparejan una llave única con cada valor, de manera similar a como funciona una tabla Hash.[3][4]

Debido a su simpleza, además de ser sencilla de implementar, tiene una velocidad muy alta de escritura y lectura de datos y también facilita su escalabilidad.[3][4]

Su principal desventaja viene de la mano de estas fortalezas, la capacidad de consulta de este tipo de bases de datos es muy limitada, en muchos casos consistiendo únicamente de la llave dada para cada valor. Además, es complicado modelar datos con mucha complejidad o representar relaciones detalladas entre ellos.[3][4]

Sus principales casos de uso son:

- Aplicaciones web: En el guardado de caché y manejo de sesiones.
- Carritos de compra
- Servicios multijugador

Documentales

Este tipo bases de datos almacenan información semiestructurada junto a descripciones de dicha información en un formato de documento. Los formatos de documento más utilizados son: JSON, BSON y XML.[3][4]

Uno de los beneficios de su utilización es que cuentan con un esquema flexible de modo que pueden actualizar y modificar el esquema con mayor sencillez. También son muy escalables de manera horizontal y soportan nativamente estructuras complejas y anidadas.[3][4]

Un problema que se presenta al usar este modelo es que su consistencia es menos robusta (tiene consistencia eventual) dado que no todos los documentos se actualizan al mismo tiempo. Además, puede llegar a presentar duplicación de datos por la naturaleza del uso de documentos. Finalmente, no tiene un gran soporte para transacciones (aunque varios DBMS comienzan a implementar esto).[3][4]

Sus principales casos de uso son:

- Aplicaciones móviles
- E-commerce
- Analíticas en tiempo real
- Sistemas de manejo de contenido

Grafos

Este modelo mapea los datos como nodos y sus relaciones como aristas, como su nombre lo indica, se basan en el concepto de la estructura de un grafo.[3][4]

La principal utilidad de este modelo es la representación de relaciones entre los datos, es capaz de representar relaciones muy complejas entre muchos nodos. Además, su esquema no es rígido como otros

modelos, este puede cambiar y evolucionar con el tiempo y el uso.[3][4]

Este tipo de base no es útil para el modelado de simples datos tabulares, al igual que suele ser ineficiente para problemas que no se relacionan con grafos. Tiene una complicada curva de aprendizaje de su lenguaje de consulta a comparación de otros modelos y cuenta con un ecosistema más reducido.[3][4]

Algunos de sus usos más comunes son:

- Redes sociales
- Detección de fraude
- Redes de conocimiento
- Motores de recomendación

Columnares

El modelo columnar organiza la información en columnas que a su vez están agrupadas en familias de columnas, a diferencia de filas fijas como el modelo relacional.[3][4]

Esta estructura permite una mucho mayor optimización al momento de trabajar con grupos muy grandes de datos, pues se pueden tomar ciertos valores de todos los datos al mismo tiempo ignorando el resto. Es muy escalable horizontalmente y también muy flexible. También cuentan con una gran eficiencia de compresión, pues esta puede hacerse a nivel columna [3][4]

Sus desventajas incluyen: resulta complicado el modelado de datos complejos, cuentan con consistencia eventual, la consulta es limitada en complejidad y suelen ser más complejas de lo necesario para conjuntos pequeños de datos.[3][4]

Sus usos más comunes incluyen:

- Datos científicos
- Motores de recomendación
- Monitoreo de eventos
- Big Data

Referencias

- [1] MongoDB, “What is an object-oriented database?” <https://www.mongodb.com/resources/basics/databases/what-is-an-object-oriented-database>, 2025, accessed: Feb. 2026.
- [2] TecnoDigital, “Bases de datos orientadas a objetos: Características y usos,” <https://informatecdigital.com/bases-de-datos-orientadas-a-objetos/>, 2025, accessed: Feb. 2026.
- [3] Debraj Maity, “Types of databases: A comprehensive guide,” <https://allaboutdatabase.com/posts/types-of-databases-comprehensive-guide/>, 2025, accessed: Feb. 2026.
- [4] Kinza Yasar and Craig S. Mullins and Jack Vaughan, “What is nosql (not only sql database)?” <https://www.techtarget.com/searchdatamanagement/definition/NoSQL-Not-Only-SQL>, 2024, accessed: Feb. 2026.