

- **Modelo orientado a objetos.**

El modelo de bases de datos orientado a objetos agrupa la información en paquetes relacionados entre sí, los datos de cada registro se combinan en un solo objeto, con todos sus atributos. Así toda la información está disponible en el objeto, ya que sus datos quedan agrupados en lugar de distribuidos en diferentes tablas.

- **Ventajas**

- Los conjuntos de datos complejos pueden guardarse y consultarse de forma rápida y sencilla.
- Los códigos de identificación se asignan automáticamente a cada objeto.
- Buena función con lenguajes de POO

- **Desventajas**

- Modelo muy poco extendido
- En algunos casos la gran complejidad puede acarrear problemas de rendimiento.

- **Casos de uso**

- Investigación científica. Se usan para manejar gran cantidad de datos y crear modelos complejos.
- Sistemas de información geográfica. Se usan para manejar datos espaciales complejos, como mapas y coordenadas.
- Telecomunicaciones, Biología molecular, etc..

## • Modelo NoSQL (clave-valor)

Se basa en una tabla de dos columnas. En una de ellas se guarda un valor y en la otra, una clave que representa una característica identificativa única o clasificadora y almacenan la información en diccionarios.

### - Ventajas

- Son efectivas para consultas y fáciles de escalar
- Gran velocidad de búsqueda debido a la sencilla conexión entre la clave y el valor

### - Desventajas

- No existe otro método de acceso. La búsqueda solo es posible con la clave.

### - Casos de uso

Se usan cuando se requiere una alta velocidad con un gran volumen de datos. Por ejemplo:

- Carritos de compra de tiendas online
- Datos de inicio de sesión

## • Modelo NoSQL (clave-valor)

Este tipo de base de datos almacena los datos en forma de documentos, por lo general en formatos JSON, XML o BSON.

### - Ventajas

- Permite almacenar y consultar información semiestructurada sin una estructura definida
- Modelo flexible, puede albergar numerosos tipos de datos.
- Simplifica tareas de edición o actualización de datos
- Garantiza buen rendimiento al contar con potentes motores de búsqueda e indexación lo que asegura una mayor rapidez en la búsqueda.
- Gran escalabilidad y permite almacenar una gran cantidad de datos.

### - Desventajas

- No utilizan el lenguaje SQL
- Gran cantidad de uso de memoria RAM
- Poca información jerárquica de esta base de datos

### - Casos de uso.

- Administración de contenido como Blogs, y plataformas de video. Permite actualizar una aplicación cambiante en requisitos de una manera intuitiva.
- Catálogos
- Gestión de sensores

## • Modelo NoSQL (grafos)

Es una colección sistemática de datos que enfatiza las relaciones entre diferentes entidades de datos.

Utiliza la teoría matemática de grafos para mostrar las conexiones entre los datos. Almacenan los datos en estructuras de tablas rígidas y almacenan los datos como una red de entidades y relaciones.

### - Ventajas

- Flexibilidad. El esquema y estructura de los modelos de grafos pueden modificarse o agrandarse sin afectar funciones existentes.
- Rendimiento. Este mejora realmente cuando se consultan relaciones; se mantiene constante con el aumento de datos.
- Eficiencia. Consultas cortas y eficientes, aprovecha los nodos enlazados.

### - Desventajas

- Dificultades en autenticidad y patrones de estandarización.
- Dificultades similares a BD relacional al realizar particiones

### - Casos de uso

- Detección de fraude. Util para estudiar patrones relacionales en estrategias de fraude
- Recomendaciones en tiempo real en redes sociales. Util para establecer relaciones entre personas e intereses por sus relaciones entre nodos
- Gestión de volumen de datos. Al implementar esta BD es posible gestionar, monitorizar y optimizar redes físicas sin que el volumen de datos sea un problema.