

## Modelo orientado a objetos

Descripción: Almacena datos como objetos con atributos y métodos.

Ventajas: Facilita el manejo de datos complejos y es natural para programadores.

Desventajas: Poco usado y difícil de integrar con bases de datos relacionales.

Casos de uso: Aplicaciones científicas, sistemas CAD y software complejo.

## Modelos NoSQL

### - Clave-valor

Descripción: Almacena datos en pares clave-valor, donde una clave única identifica y permite acceder directamente a un valor completo.

Ventajas: Muy rápido y altamente escalable.

Desventajas: No permite consultas complejas ni relaciones.

Casos de uso: Caché, sesiones de usuario y configuraciones.

### - Documentales

Descripción: Almacena datos en documentos como JSON o XML.

Ventajas: Flexible y fácil de usar.

Desventajas: Poca eficiencia en joins y posible duplicación de datos.

Casos de uso: Aplicaciones web, APIs y CMS.

### - Grafos

Descripción: Representa datos como nodos y relaciones.

Ventajas: Excelente manejo de relaciones y conexiones.

Desventajas: Curva de aprendizaje y no ideal para datos simples masivos.

Casos de uso: Redes sociales, sistemas de recomendación y rutas.

### - Columnar

Descripción: Almacena datos organizados por columnas.

Ventajas: Alto rendimiento en análisis y consultas.

Desventajas: Escrituras más lentas y estructura rígida.

Casos de uso: Big Data, analítica y data warehouses.

## Bibliografía

- Silberschatz, H. F. Korth y S. Sudarshan, *Database System Concepts*, 6th ed. New York, NY, USA: McGraw-Hill, 2011.
- P. J. Sadalage y M. Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Boston, MA, USA: Addison-Wesley, 2013.