

Bases de Datos: Modelos Orientados a Objetos y NoSQL

Zúñiga Arreguín Emilio

I. Modelo orientado a objetos

- **Descripción:** Una BDOO (Base de Datos Orientada a Objetos) mezcla la Programación Orientada a Objetos con el almacenamiento de datos de manera persistente. En este modelo, cada objeto mantiene sus atributos y métodos, lo que permite que las operaciones que se definen en el código puedan aplicarse directamente sobre la base de datos sin pasos intermedios.
- **Ejemplo:** *db4o* y *ObjectDB*, que en su momento fueron bastante usados. Hoy en día ya no son tan comunes, pero todavía se aplican en áreas específicas como simuladores o software de diseño CAD.
- **Ventajas:**
 - Es más adecuado en proyectos científicos o técnicos que manejan estructuras de datos complejas.
 - No requiere usar mapeo objeto-relacional (ORM).
- **Desventajas:**
 - Tiene poco soporte en la nube si se compara con SQL o NoSQL.
 - Su ecosistema es reducido y hay pocas herramientas.
- **Casos de uso:** Se utiliza en algunos videojuegos (para guardar personajes u objetos) y también en áreas como la biología computacional.

II. Modelos NoSQL

Clave–Valor

- **Descripción:** Su estructura es clave–valor. Se ocupa cuando no hacen falta relaciones entre datos ni consultas complicadas.
- **Ventajas:** Es rápido para accesos repetitivos y facilita el almacenamiento distribuido.
- **Desventajas:** No es recomendable para reportes o análisis complejos.
- **Casos de uso:**
 - Amazon DynamoDB, que se usa en Alexa y Amazon.com.
 - Redis, que maneja sesiones de millones de usuarios en Twitter.

Documentales

- **Descripción:** Cada registro se almacena como un documento (en JSON, BSON o XML). No necesitan un esquema fijo.
- **Ventajas:** Es flexible y funciona muy bien en arquitecturas de micro-servicios.
- **Desventajas:** Cuando los documentos son muy grandes, el rendimiento puede verse afectado.
- **Casos de uso:**
 - MongoDB, empleado en Uber y eBay.
 - CouchDB, que sirve para sincronizar datos en dispositivos móviles sin conexión.

Grafos

- **Descripción:** Están diseñados para representar relaciones mediante nodos (entidades) y aristas (conexiones).
- **Ventajas:** Facilita consultas en profundidad, como las redes de conexiones (ejemplo: “amigos de amigos”).

- **Desventajas:** Tiene dificultades para escalar de forma horizontal.
- **Casos de uso:**
 - Neo4j, usado en LinkedIn y eBay para recomendaciones.
 - Facebook Graph API, que se inspira en este tipo de modelos.

Columnar

- **Descripción:** Los datos se almacenan por columnas en lugar de filas, lo que resulta útil en tareas de análisis.
- **Ventajas:** Muy eficiente para Big Data y análisis OLAP.
- **Desventajas:** No es tan eficaz en transacciones pequeñas.
- **Casos de uso:**
 - Apache Cassandra, que usa Netflix para historiales de streaming.
 - Google Bigtable, que inspiró proyectos como Cassandra y HBase.

III. Bibliografía IEEE

1. MongoDB Inc., “What is MongoDB?,” [Online]. Available: <https://www.mongodb.com/>.
2. Amazon Web Services, “Amazon DynamoDB,” [Online]. Available: <https://aws.amazon.com/dynamodb/>.
3. Neo4j Inc., “Neo4j Graph Database Platform,” [Online]. Available: <https://neo4j.com/>.
4. Apache Software Foundation, “Apache Cassandra,” [Online]. Available: <https://cassandra.apache.org/>.
5. ObjectDB Software, “Object Database for Java/JPA,” [Online]. Available: <https://www.objectdb.com/>.