

Conceptos Claves de Modelo NoSQL y Orientado a Objetos

Miguel Ángel Ruíz Sánchez

15 de agosto de 2023

1. Modelo NoSQL

a) Conceptos Clave:

Las bases de datos NoSQL (Not Only SQL) han revolucionado el panorama de la gestión de datos al abordar eficazmente la necesidad de almacenar y acceder a grandes volúmenes de datos no estructurados. Estas bases de datos se destacan por su capacidad de escalabilidad horizontal y por permitir flexibilidad en el esquema de datos, lo que las hace ideales para aplicaciones modernas que requieren adaptabilidad constante.

b) Origen:

Las bases de datos NoSQL surgieron como una respuesta a las limitaciones de las bases de datos tradicionales. Estos sistemas emergieron como alternativas a las bases de datos SQL tradicionales, que a menudo se enfrentaban a limitaciones en términos de escalabilidad y manejo de datos no convencionales.

c) Ejemplos de Manejadores:

- **MongoDB:** Es un claro ejemplo de un manejador NoSQL orientado a documentos. Sus documentos BSON (similar a JSON) permiten flexibilidad en el esquema y facilitan la gestión de datos semiestructurados.
- **Cassandra:** Como base de datos de tipo columna, se destaca por su escalabilidad horizontal. Ideal para entornos donde se requiere manejar grandes volúmenes de datos distribuidos.
- **Redis:** Actuando como almacén de datos en memoria clave-valor, Redis es esencial para aplicaciones que priorizan la velocidad, como la gestión de caché y sesiones en aplicaciones web.
- **Neo4j:** Una base de datos de grafos NoSQL especializada en almacenar y consultar relaciones entre datos. Se destaca en aplicaciones que involucran interconexiones de datos, como redes sociales y sistemas de recomendación.

d) Ventajas:

- **Escalabilidad Adaptativa:** Las bases de datos NoSQL ofrecen una escalabilidad horizontal eficiente, permitiendo la distribución y el manejo de cargas de trabajo en constante crecimiento.
- **Esquema Flexible:** Este tiene la capacidad de adaptar el esquema de datos sin restricciones, permitiendo realizar cualquier cambio con facilidad.
- **Rendimiento:** Al evitar las complejidades de las bases de datos SQL en términos de integridad y relaciones, las bases de datos NoSQL pueden lograr un alto rendimiento en operaciones específicas según el caso de uso.

e) Desventajas:

- **Deficiencia de Multiplataforma:** Las bases de datos NoSQL carecen de una multiplataforma.
- **Adopción en Evolución:** Aunque las bases de datos NoSQL han ganado terreno, todavía están en proceso de adopción en comparación con las bases de datos SQL tradicionales, lo que puede impactar en términos de recursos y soporte disponibles.
- **Complejidad en Consultas:** Las bases de datos NoSQL pueden enfrentar dificultades al lidiar con consultas que involucran relaciones complejas.

f) BD'S basadas en NoSQL:

- **Clave-Valor:** Una base de datos clave-valor se asemeja a un sistema de almacenamiento en el que cada elemento se identifica con una etiqueta única (clave) y está acompañado por su contenido correspondiente (valor). Esta estructura permite la inclusión de diversos tipos de datos, ya sean numéricos, textuales u otros.
- **Documentales:** Una base de datos documental es una variante importante de las bases de datos NoSQL. Que esta es capaz de almacenar información en registros individuales, llamados documentos. Cada documento es una unidad completa de información. Estas bases de datos utilizan documentos para guardar registros y sus datos asociados. Los documentos pueden contener diferentes tipos de información. Además, estos documentos pueden tener diversos formatos, como archivos JSON, XML o documentos de texto, lo que ofrece flexibilidad en cómo se almacena la información.
- **Grafos:** Las bases de datos orientadas a grafos (BDOG) destacan por su forma de representar información usando nodos y conexiones entre ellos, siguiendo conceptos de Teoría de Grafos.

2. Modelo Orientado a Objetos

a) Conceptos Clave:

Las bases de datos orientadas a objetos son sistemas de gestión de bases de datos diseñados para almacenar y recuperar datos de manera similar a cómo funcionan los objetos en la programación orientada a objetos. En lugar de utilizar tablas y filas, estas bases de datos almacenan datos en forma de objetos que pueden contener atributos y métodos.

b) Origen:

El concepto de bases de datos orientadas a objetos surge como una extensión natural de la programación orientada a objetos. Se desarrollaron en respuesta a la necesidad de unir la forma en que se almacenan los datos en los sistemas de bases de datos con la forma en que se modelan y manipulan en la programación.

c) Ejemplos de Manejadores:

- **db4o:** Un sistema de bases de datos orientado a objetos que permite a los desarrolladores almacenar y recuperar objetos directamente en la base de datos, eliminando la necesidad de traducciones entre objetos y registros.
- **Versant:** Enfocado en aplicaciones que requieren un alto rendimiento y una escalabilidad sólida, Versant es un sistema de bases de datos orientado a objetos que puede manejar grandes volúmenes de datos y consultas complejas.
- **OrientDB:** Un sistema de bases de datos orientado a objetos y de grafos que permite combinar ambos enfoques para modelar y consultar datos de manera eficiente.

d) Ventajas:

- **Modelo Coherente:** Las bases de datos orientadas a objetos reflejan directamente el paradigma de la programación orientada a objetos, lo que facilita la persistencia de objetos complejos.
- **Eficacia en Consultas:** Para aplicaciones que requieren consultas y relaciones complejas, las bases de datos orientadas a objetos ofrecen un enfoque eficiente para modelar y acceder a los datos.
- **Rendimiento Mejorado:** La integración directa de objetos en la base de datos puede resultar en un mejor rendimiento para aplicaciones con uso intensivo de objetos.

e) Desventajas:

- **Falta de Estandarización:** A menudo, las bases de datos orientadas a objetos carecen de un estándar común, lo que puede dificultar la migración entre sistemas.

- **Complejidad en Escalabilidad Horizontal:** En algunas situaciones, la escalabilidad horizontal puede ser más compleja en comparación con otros modelos de bases de datos.
- **Curva de Aprendizaje:** Tanto para desarrolladores como para administradores, las bases de datos orientadas a objetos pueden requerir una curva de aprendizaje más pronunciada debido a su naturaleza especializada.

Referencias

- [1] Bases de Datos NoSQL: Marcas, Tipos, Ventajas. Recuperado de <https://www.grapheverywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/>
- [2] Bases de Datos NoSQL. Recuperado de <https://mexico.unir.net/ingenieria/noticias/bases-de-datos-nosql/>
- [3] Análisis Comparativo de Sistemas de Gestión de Bases de Datos SQL y NoSQL. Recuperado de https://ebuah.uah.es/dspace/bitstream/handle/10017/38612/TFG_Diaz_Moreno_2019.pdf?isAllowed=y&sequence=1
- [4] Bases de Datos Documentales. Recuperado de <https://ayudaleyprotecciondatos.es/bases-de-datos/documentales/>
- [5] Bases de Datos Orientada a Grafos. Recuperado de <https://www.sngular.com/es/que-son-bases-datos-a-grafos/>
- [6] Bases de Datos Orientadas a Objetos. Recuperado de https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/782/mod_resource/content/8/contenido/index.html
- [7] Conceptos Básicos sobre Modelo de Datos Orientado a Objetos. Recuperado de <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/conceptos-basicos-sobre-modelo-de-datos-orientado-a-objetos>