

Universidad Nacional Autónoma De México

Facultad de Ingeniería

Grupo: 01

Curso: Base de Datos

Alumno: Ortiz Valles Joaquín Rafael

Profesor: Ing. Fernando Arreola Franco

Tarea: 1

Modelo orientado a objetos

Un modelo orientado a objetos es una forma de representar sistemas complejos mediante la simulación de entidades del mundo real como objetos digitales. Cada objeto encapsula tanto información (estado) como funcionalidad (comportamiento), lo que permite construir software más modular, reutilizable y fácil de mantener.

Este modelo no solo organiza el código, sino que también estructura el pensamiento del desarrollador, facilitando la resolución de problemas mediante analogías con el entorno físico o social.

Concepto	Descripción breve
Clase	Es una plantilla o molde que define atributos y métodos comunes para un tipo de objeto.
Objeto	Instancia concreta de una clase. Tiene valores específicos y puede ejecutar acciones.
Encapsulamiento	Oculto los detalles internos del objeto y expone solo lo necesario. Protege la información.
Herencia	Permite que una clase herede atributos y métodos de otra, promoviendo la reutilización.
Polimorfismo	Capacidad de usar métodos con el mismo nombre pero comportamientos distintos según el contexto.
Abstracción	Simplifica la complejidad mostrando solo lo esencial del objeto.

Facilita el mantenimiento y escalabilidad del software.

Promueve la reutilización de código.

Permite modelar sistemas complejos de forma más intuitiva.

Modelos NoSQL

NoSQL (Not Only SQL) es un conjunto de modelos de bases de datos que no siguen el esquema rígido de tablas y relaciones como en SQL. Están diseñados para ofrecer alta disponibilidad, escalabilidad horizontal y flexibilidad en el manejo de datos semiestructurados o no estructurados, como documentos, registros de eventos, redes sociales o sensores IoT.

Ventajas de NoSQL

Escalabilidad horizontal: Se adapta fácilmente a sistemas distribuidos.

Flexibilidad de esquemas: Puedes cambiar la estructura de los datos sin afectar toda la base.

Alto rendimiento: Optimizado para grandes volúmenes y consultas rápidas.

Ideal para Big Data y aplicaciones en tiempo real.

Casos en Los cuales usar NoSQL

Cuando los datos no tienen una estructura fija.

Si necesitas alta disponibilidad y tolerancia a fallos.

Para proyectos que crecen rápidamente y requieren escalabilidad dinámica.

Principales de modelos NoSQL

Tipo de modelo	Descripción breve	Ejemplo
Documental	Almacena datos como documentos (generalmente en JSON o BSON). Cada documento es autónomo.	MongoDB
Clave-valor	Guarda pares clave-valor, como un diccionario. Muy rápido y simple.	Redis, DynamoDB
Columnar	Organiza los datos por columnas en lugar de filas, ideal para análisis masivo.	Cassandra, HBase
Grafos	Representa relaciones entre entidades como nodos y aristas. Perfecto para redes sociales o rutas.	Neo4j, ArangoDB

Desventajas de NoSQL

1. Falta de estandarización

No existe un lenguaje universal como SQL.

Cada sistema (MongoDB, Cassandra, Redis, etc.) tiene su propia sintaxis y comandos.

2. Consistencia eventual

Muchos modelos NoSQL priorizan disponibilidad y partición (según el teorema CAP), sacrificando consistencia inmediata.

Esto puede generar lecturas desactualizadas en sistemas críticos.

3. Dificultad para relaciones complejas

NoSQL no maneja bien relaciones entre entidades como lo hace SQL con JOIN.

Si necesitas modelar relaciones entre estudiantes, becas y evaluaciones, puede volverse complicado.

4. Menor soporte para transacciones

Algunos motores NoSQL no ofrecen transacciones ACID completas.

Esto puede ser un problema si necesitas garantizar integridad en operaciones múltiples.

5. Curva de aprendizaje

Requiere entender nuevos paradigmas de modelado de datos.

No basta con saber cómo guardar datos; hay que diseñar estructuras que se adapten al tipo de consulta.

6. Menor madurez en algunos casos

Aunque hay motores muy robustos, otros aún están en desarrollo o tienen comunidades pequeñas.

Esto puede limitar el soporte técnico o la documentación disponible.

Bibliografía

[1] W. Walther, "SQL vs NoSQL: Ventajas y Desventajas Detalladas de Cada Sistema," Dongee, 24 ago. 2023. [En línea]. Disponible en: <https://www.dongee.com/tutoriales/sql-vs-nosql-ventajas-y-desventajas-detalladas-de-cada-sistema>

[2] A. Rodríguez, "Bases de datos NoSQL: Guía con las ventajas y desventajas," Pandora FMS, 19 mar. 2024. [En línea]. Disponible en: <https://pandorafms.com/blog/es/bases-de-datos-nosql/>

[3] Oracle México, "¿Qué es NoSQL?," Oracle, 18 jun. 2021. [En línea]. Disponible en: <https://www.oracle.com/mx/database/nosql/what-is-nosql/>

[4] IBM, "¿Qué es una base de datos NoSQL?," IBM Think, 12 dic. 2022. [En línea]. Disponible en: <https://www.ibm.com/es-es/think/topics/nosql-databases>

[5] Stackscale, "Bases de datos NoSQL: características y tipos," Stackscale Blog, 2 ene. 2023. [En línea]. Disponible en: <https://www.stackscale.com/es/blog/bases-de-datos-nosql/>

[6] Programación Pro, "Tipos de Bases de Datos NoSQL: Una Guía Completa," Programación Pro. [En línea]. Disponible en: <https://programacionpro.com/tipos-de-bases-de-datos-nosql-una-guia-completa/>