

MODELO ORIENTADA A OBJETOS

Propone organizar el programa en objetos que intervienen en los procesos que se representan y sus interacciones, concentra las características de los objetos, las acciones que realizan y sus interacciones [1].

Las características más importantes del paradigma son:

1. Abstracción: identifica los elementos más importantes para representar un objeto.
2. Polimorfismo: capacidad de un objeto de realizar una acción de diferentes formas.
3. Herencia: pasar características entre objetos
4. Encapsulamiento: puede o no ocultar los atributos y métodos de una clase,

Ventajas:

1. Modularidad: facilita la organización de código en clases y objetos.
2. Reutilización del código: permite volver a usar las clases en diferentes proyectos mediante la herencia y el polimorfismo.
3. Mantenimiento: los posibles cambios pueden ser locales sin afectar otras partes del programa

Desventajas:

1. Sobre carga de memoria: los objetos y sus atributos puedes consumir más memoria.
2. Dependencia: cambios en las clases pueden afectar a las clases derivadas.
3. Rendimiento bajo: la POO puede ser menos eficiente en velocidad y consumo de recursos que otros paradigmas.

Casos de uso:

- Aplicaciones móviles
- Simulación y modelado
- Desarrollo web.
- Automatización de pruebas

MODELO NoSQL

Es un enfoque para el diseño de bases de datos que permiten el almacenamiento y la consulta de datos fuera de las estructuras relacionales [2]. Ofrendes estructuras más flexibles que permite una escalabilidad horizontal más eficiente, distribuyendo la carga de trabajo en múltiples servidores.

Tipos de bases NoSQL

- Clave-valor: almacena datos como pares clave-valor.
- Bases de datos de columnas: organizan en columnas en vez de filas.
- Bases de datos de documentos: gestionan datos semiestructurados almacenados en formatos como JSON o XML.
- Almacenes de grafos: gestionan redes de relaciones y conexiones como Neo4j.

Ventajas:

1. Rentabilidad: la escalabilidad horizontal optimiza el uso de recursos.
2. Flexibilidad: capacidad de manejar datos que cambian rápidamente.

3. Velocidad: procesamiento ágil que demandan respuestas rápidas.

Desventajas:

1. Falta de estandarización, no existe un estándar universal para NoSQ, por lo que varían muchísimo entre sí.
2. Menor soporte para consultas complejas.
3. Escalabilidad compleja: gestionar clústeres distribuidos y mantenerlos sincronizados puede ser complicado.

Casos de uso:

- Aplicaciones web y móviles.
- Análisis de big data.
- Sistemas de gestión de contenido.
- Redes sociales y aplicaciones [2].

BIBLIOGRAFÍA

[1] IBM, "Programación orientada a objetos - IBM Documentation", IBM, [en línea]. Disponible: <https://www.ibm.com/docs/es/spss-modeler/saas?topic=language-object-oriented-programming>. [Accedido: 8-feb-2025].

[2] IBM, "¿Qué es una base de datos NoSQL?", IBM, [en línea]. Disponible: <https://www.ibm.com/mx-es/think/topics/nosql-databases>. [Accedido: 8-feb-2025].