

Alumno: **Ponce de León Reyes Bruno**

Tarea 1

Modelo orientado a objetos M.O.O.

La orientación a objetos es un paradigma de desarrollo de software utilizado para representar el mundo real de manera sencilla. Los sistemas diseñados en este paradigma son definidos por los componentes y la forma en que estos interactúan entre sí.

En el caso de las bases de datos que utilizan este modelo, los datos almacenados se representan mediante objetos. Se les denomina como *Object Database Management System* (ODBMS), y son empleadas principalmente en áreas como la ingeniería, las telecomunicaciones y en diversas áreas científicas.

Ventajas:

- Gran facilidad para integrar con lenguajes de programación orientados a objetos.
- El acceso a datos es más rápido debido al menor uso de Joins, a diferencia de los que ocurre con las bases de datos de relacionales.
- Son eficientes en áreas donde se requieren grandes cantidades de datos sobre un elemento/objeto.

Desventajas:

- La complejidad de su uso puede conllevar a problemas de rendimiento.
- Existencia de pocos SGDB para bases de datos que utilizan este modelo, debido a que su uso está poco extendido.

Casos de uso:

- Este modelo es adecuado si se utilizan lenguajes de programación orientados a objetos, dado a la facilidad para su integración con la bases de datos al usar el mismo paradigma de desarrollo.
- Es utilizada principalmente en áreas donde se requieren grandes cantidades de datos de un elemento, tales como áreas científicas y matemáticas.

Modelos NoSQL (Clave, valor, documentales, grafos y columnar)

Las bases de datos NoSQL utilizan colecciones para almacenar información en lugar de tablas.

- **Clave-Valor**

Tipo de base de datos simple en donde cada artículo tiene una clave única y dicha clave está asociada con un valor individual. Ofrecen alto rendimiento en lecturas y escrituras de datos.

Ventajas:

- Es el modelo más sencillo de todos, lo cual facilita su implementación.
- Ofrecen una alta velocidad en la escritura y lectura de datos.
- Son eficientes en el uso de datos de acceso frecuente o temporales.

Desventajas:

- No es posible establecer relaciones entre datos.
- Dada a la estructura del modelo, no es posible realizar consultas complejas de datos.

Casos de uso:

- Dada a su velocidad de escritura y lectura de datos se suelen usar en sistemas de almacenamiento en caché.
- Igualmente, es óptima para la gestión de sesiones.

- **Documentales**

La base de datos almacena datos en documentos, usualmente estos son objetos JSON o similares (XML, BSON, etc.). Ofrece un modelo de datos flexible, adaptado para conjuntos de datos semiestructurados y no estructurados. Son utilizados para representar relaciones complejas o datos jerárquicos.

Ventajas:

- Son flexibles, permiten cambiar el esquema de datos de una manera ágil.
- Permiten la representación de relaciones complejas y jerarquías de los datos.
- Se pueden incluir una gran variedad de campos para ingresar datos en un documento.

Desventajas:

- El uso de documentos puede generar redundancia de los datos.
- Si no se tiene un buen control, el uso de varios documentos pueden generar inconsistencias.

Casos de uso:

- Su uso es recomendable en la industria, esto incluye aplicaciones web, aplicaciones móviles, etc.
- Catálogos de productos de una tienda web.

- **Grafos**

Bases de datos que almacenan datos en forma de nodos y aristas. Los nodos son utilizados para almacenar datos, mientras que las aristas almacenan información de las relaciones entre los nodos. Son utilizados cuando se trabaja con datos que tienen varias relaciones y patrones complejos entre entidades.

Ventajas:

- Ideal para la representación de relaciones entre datos.
- La consulta de conexiones entre datos es eficiente.
- La manera en que se representan las relaciones entre datos es intuitiva y parecida al mundo real.

Desventajas:

- Si no se realiza un correcto análisis de los datos a emplear y sus relaciones, el uso de este modelo puede resultar desfavorable.
- La escalabilidad del modelo puede ser compleja si se tienen grafos muy grandes.

Casos de uso:

Es ideal cuando los datos a emplear suelen tener una gran cantidad de relaciones entre sí, por ejemplo:

- Redes sociales.
- Bases de conocimiento.
- Inteligencia artificial.

- **Columnar (columnas anchas)**

Estas bases de datos utilizan tablas con filas y columnas dinámicas, esto significa que las columnas son flexibles. Las filas y columnas permiten la recuperación eficiente de datos dispersos y amplios. Este sistema puede ser más complejo de gestionar. Suelen utilizarse en aplicaciones de big data.

Ventajas:

- Alta eficiencia para almacenar una cantidad masiva de datos, facilitando su búsqueda y recuperación.
- Suelen garantizar una alta disponibilidad de los datos, lo que permite el manejo de sistemas en tiempo real que requieren actualizaciones constantes y rápidas.
- Permiten guardar datos que se generan constantemente.

Desventajas:

- El diseño de una base de datos que utiliza este modelo puede ser complejo.
- Su uso es poco rentable para sistemas o aplicaciones pequeñas en donde no se generan una gran cantidad de datos.
- Suelen requerir mayor espacio de almacenamiento a diferencia de otros modelos.

Casos de uso:

Son empleadas para almacenar cantidades masivas de datos, lo cual las hace útiles en áreas como:

- Big data.
- IoT.
- Logs de un sistema.

Bibliografía (IEEE)

- “Base de datos orientada a objetos: la información en unidades,” *IONOS Digital Guide*, Jan. 18, 2023. <https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/base-de-datos-orientada-a-objetos/>
- KYOCERA Document Solutions, “Conceptos sobre base de datos orientada a objetos | Kyocera,” *KYOCERA Document Solutions*. <https://www.kyoceradocumentsolutions.es/es/smarter-workspaces/business-challenges/paperless/conceptos-sobre-base-de-datos-orientada-a-objetos.html>
- Arsys, “Bases de datos NoSQL: qué son, tipos y ventajas,” Arsys, Feb. 07, 2025. <https://www.arsys.es/blog/bases-de-datos-nosql-que-son-tipos-y-ventajas>
- “¿Qué es NoSQL? Descripción de las bases de datos NoSQL,” *MongoDB*. <https://www.mongodb.com/es/resources/basics/databases/nosql-explained>