



**Universidad Nacional Autónoma de
México**

Facultad de Ingeniería



BASES DE DATOS

Semestre 2025 - 1

Tarea 1

Alumno: Fernández Herrera Mauricio

Profesor: ING. FERNANDO ARREOLA FRANCO

Grupo: 1

Fecha de entrega: 07/02/2025

BASES DE DATOS ORIENTADAS A OBJETOS

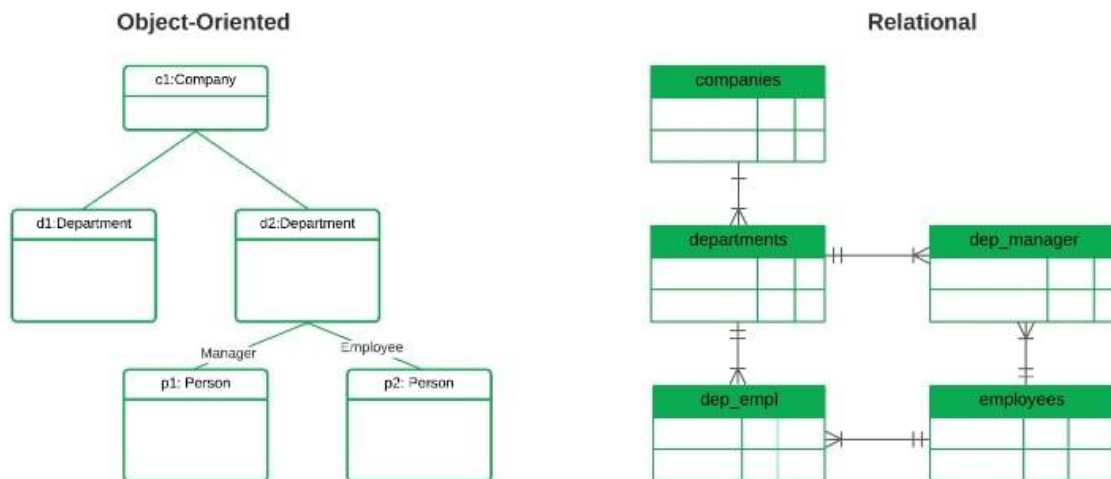
Este tipo de modelado de bases de datos, como su nombre lo sugiere, toma los modelos utilizados en la programación orientada a objetos y los reinterpreta para ser utilizados en la construcción de una base de datos. Esta hace uso de los términos y conceptos propuestos por este paradigma como, por ejemplo, el uso del concepto de clase y atributos de la misma, los cuáles funcionan como tabla y datos respectivamente. Usualmente se utiliza este modelo cuando se desee almacenar información de objetos reales, ya que esto es la misma metodología que tiene en mente la programación orientada a objetos.

Elementos de una base de datos orientada a objetos:

Objeto: Funciona como una entidad dentro de la propia base de datos, pudiendo ser objetos, personas o entes que llevan a cabo una tarea en concreto.

Clase: Agrupación de todos los objetos que tengan las mismas características o datos por almacenar, además de métodos y comportamientos.

Atributos y métodos: Los atributos son todas las características que definen al objeto, mientras que los métodos son los comportamientos que pueden llevar a cabo. Ambos funcionan como datos y un conjunto de información que se debe almacenar.



Características de una base de datos orientada a objetos:

Características principales de las bases de datos orientadas a objetos:

- 1.- Encapsulación:** Aplicación del concepto de transparencia en una base de datos. Los objetos encapsulan sus propiedades y métodos, de modo que esto sea lo único a lo que se pueda acceder.
- 2.- Herencia:** Los objetos pueden heredar características y comportamientos de otros objetos, facilitando la reutilización de código y conservando información útil y confiable.
- 3.- Polimorfismo:** Los objetos pueden responder a un llamado de varias formas en base a los métodos que estos tengan definidos.
- 4.- Abstracción:** Los objetos representan conceptos del mundo real de forma simplificada, ocultando detalles irrelevantes y enfocándose en lo esencial, evitando manejo de información redundante.
- 5.- Persistencia:** Los objetos pueden almacenarse y recuperarse de forma permanente, garantizando la integridad y disponibilidad de la información.

[1] [3]

Ventajas de las bases de datos orientadas a objetos:

- **Modelado natural de datos complejos:** Las BDOO permiten modelar de forma natural relaciones complejas y jerárquicas entre datos, reflejando mejor la estructura del mundo real.
- **Flexibilidad y escalabilidad:** Las BDOO son altamente flexibles y escalables, adaptándose fácilmente a cambios en los requisitos de la aplicación.
- **Reutilización de código:** La herencia y el polimorfismo permiten reutilizar código de manera eficiente, reduciendo el tiempo de desarrollo y mejorando la calidad del software.
- **Mantenimiento simplificado:** La modularidad y encapsulación de los objetos facilitan el mantenimiento y la actualización del código.
- **Integración con lenguajes de programación orientados a objetos:** Las BDOO se integran perfectamente con lenguajes de programación orientados a objetos, como Java, C++ o Python, facilitando el desarrollo de aplicaciones.

[2]

Desventajas de las bases de datos orientadas a objetos:

- **Falta de estandarización y madurez:** Las bases de datos orientadas a objetos no cuentan con un estándar tan consolidado como SQL para las bases de datos relacionales. Esto puede dificultar la interoperabilidad entre sistemas y la migración de datos, además de

limitar la disponibilidad de herramientas y recursos de soporte en comparación con los sistemas relacionales.

- **Complejidad en el modelado y diseño:** La definición de clases, herencias, relaciones y métodos requiere un análisis detallado y, en algunos casos, puede implicar una mayor curva de aprendizaje para desarrolladores acostumbrados a modelos relacionales.
- **Limitado lenguaje de consulta:** A diferencia del lenguaje SQL ampliamente aceptado en el mundo relacional, los sistemas de bases de datos orientadas a objetos suelen depender de lenguajes propios o APIs específicas para la manipulación de datos. Esto puede representar un inconveniente al buscar portabilidad o al integrar con otras aplicaciones y tecnologías.
- **Dificultades en la integración con sistemas existentes:** La integración de una base de datos orientada a objetos en entornos donde predominan los modelos relacionales puede resultar compleja, ya que implica gestionar la transformación y el mapeo de datos entre paradigmas distintos.
- **Menor comunidad y soporte en el mercado:** Debido a que las bases de datos orientadas a objetos tienen una adopción menor en comparación con las bases de datos relacionales, puede haber una comunidad de usuarios y desarrolladores más reducida. Esto se traduce en menos recursos, documentación y soluciones a problemas específicos, lo cual puede dificultar el desarrollo y mantenimiento a largo plazo.

[5]

Referencias:

[1] Navarro, S. (2024, 9 julio). Base de datos orientada a objetos: definición y componentes. KeepCoding Bootcamps. <https://keepcoding.io/blog/base-de-datos-orientada-a-objetos/> Consultado: 2025, 5 febrero

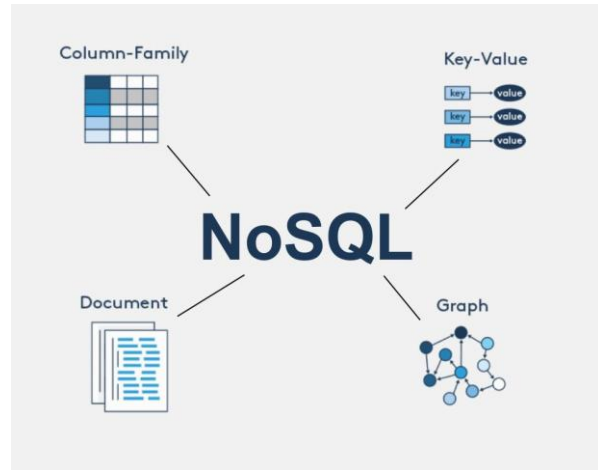
[2] Otalora, & Otalora. (2024, 15 julio). Modelo de base de datos orientado a objetos: + ejemplos. TuProfeDigital.com. <https://tuprofedigital.net/modelo-de-base-de-datos-orientado-a-objetos/> Consultado: 2025, 5 febrero

[3] Hernández, E. D. K. (s. f.). Modelo Orientado a Objetos. https://repositorio-uapa.cuaieed.unam.mx/repositorio/moodle/pluginfile.php/2731/mod_resource/content/1/UPA-Modelo-Orientado-Objetos/index.html Consultado: 2025, 5 febrero

[4] GraphEverywhere, E. (2019, 29 junio). Bases de datos NoSQL | Bases de datos orientadas a objetos. GraphEverywhere. <https://www.grapheverywhere.com/bases-de-datos-orientadas-objetos/> Consultado: 2025, 5 febrero

MODELOS NO SQL

Estos modelos forman parte de una iniciativa que tiene el enfoque de diseñar bases de datos haciendo uso de otras herramientas y estructuras diferentes a las que se encuentran en las bases de datos relacionales. Están pensadas para manejar datos estructurados o semiestructurados, fungiendo como una opción alternativa la cuál no necesariamente hace uso del ya estándar SQL, aunque últimamente han surgido bases de datos que hacen uso de esta filosofía que son compatibles con este lenguaje.



[1][2]

Características principales

- 1.- Escalabilidad horizontal:** Se pueden distribuir los datos en múltiples servidores para mejorar el rendimiento.
- 2.- Flexibilidad de esquemas:** No requieren un esquema predefinido, permitiendo cambios dinámicos en la estructura de los datos.
- 3.- Alta disponibilidad:** Están diseñados para ser tolerantes a fallos y ofrecer tiempos de respuesta rápidos.
- 4.- Optimización para grandes volúmenes:** Pueden manejar grandes cantidades de datos, tanto estructurados como no estructurados.

[3]

Ventajas

- Una base de datos de este tipo almacena información en una estructura flexible y sin un esqueleto predeterminado, concediendo la ventaja de poder adecuarse de mejor manera a los requerimientos que estas tengan.
- Proporcionan una alternativa en cuestiones de escalabilidad adoptando un enfoque horizontal en lugar de vertical a una empresa en constante crecimiento. En un caso convencional, una empresa depende de factores como la memoria al momento de expandirse. En el caso del NoSQL, no se depende de tablas para almacenar información, lo cual concede la oportunidad de fragmentar la base y expandirse mediante el uso de un servidor secundario, por ejemplo.

- No se tiene un esquema de diseño predeterminado, lo cual concede la posibilidad de considerar un enfoque fuera de la rigidez de una base de datos relacional, lo cual dota de este tipo de bases de datos de una adaptabilidad y libertad de diseño, apreciada en casos variados.
- El rendimiento de las bases de datos desarrolladas con este enfoque resulta excepcional, al hacer de operaciones tan básicas como lectura y escritura lo más sencillas posible. Esto se traduce en consulta, fragmentación, y recuperación de la información más rápida.
- Es costoso mantener RDBMS comerciales de alta gama. Requieren la compra de licencias, administradores de bases de datos capacitados y hardware potente para escalar verticalmente. Las bases de datos NoSQL le permiten escalar rápidamente horizontalmente, asignando mejor los recursos para minimizar los costos.

[1] [2]

Desventajas

- La falta de consistencia de esta iniciativa puede afectar a los datos con los que se esté trabajando, ofreciendo una integridad de los mismos variable, la cuál dependerá del equipo que se encarga de implementar la base de datos.
- A diferencia de las bases de datos relacionales, estas no poseen un nivel tan significativo de herramientas y soporte para su desarrollo, lo cuál puede complicar su implementación.
- La consulta y manejo de los datos puede no ser tan directa y dinámica como en las bases de datos más estandarizadas, lo cual puede resultar en una curva de aprendizaje complicada para el desarrollador de SQL acostumbrado al modelo estándar.
- La escalabilidad horizontal puede estar limitada en algunos casos, siendo esta inviable en ciertos casos donde se requiera una expansión considerable del sistema, a diferencia del escalado vertical.

[4]

Casos de uso principales

- **Aplicaciones de Big Data:** Las bases de datos NoSQL están especialmente diseñadas para manejar grandes volúmenes de datos que no necesariamente estén estructurados. Además, concede a la empresa una gran ventaja al ser necesario el análisis en tiempo real de estos datos, como lo requieren las redes sociales o transacciones en línea.
- **Plataformas de comercio electrónico:** Con elevados volúmenes de solicitudes y tráfico constante de datos delicados, las bases de datos NoSQL son ideales para manejar datos dinámicos, diversos, y en grandes volúmenes de forma constante, especialmente en el caso de una plataforma especializada que requiere el uso de perfiles y claves de acceso.
- **Sistemas de gestión de contenido:** Cuando la empresa depende de ofrecer una gran variedad de contenidos de distinta naturaleza como videos, imágenes y publicaciones de texto, este tipo de bases de datos son las idóneas para manejar un flujo constante de información de naturaleza tan diferente.

- **Foros y Redes Sociales:** Aplicaciones con un flujo constante de información en grandes cantidades y en constante actualización. Aunque se trate de texto, reacciones, datos personales, imágenes, o algún otro tipo de información, las bases de datos No SQL se especializan en el manejo rápidamente, lo cuál es de vital importancia en este tipo de aplicaciones.

[1][2]

Referencias:

[1] *Qué son las bases de datos NoSQL: ventajas, tipos y casos de uso.* (2025, 4 febrero). Spain. <https://www.intersystems.com/es/recursos/que-son-las-bbdd-nosql/#advantages>

Consultado: 2025, 6 febrero

[2] Ibm. (2024, 3 septiembre). bases de datos NOSQL. *IGM-opics*. <https://www.ibm.com/es-es/topics/nosql-database> Consultado: 2025, 6 febrero

[3] Stackscale. (2024, 24 enero). Bases de datos NoSQL: características y tipos. *Stackscale*. https://www.stackscale.com/es/blog/bases-de-datos-nosql/#Caracteristicas_y_ventajas_principales Consultado: 2025, 6 febrero

[4] Posadasherschel, & Posadasherschel. (2024, 16 junio). *Si estás interesado en el mundo de las bases de datos, es Leer más.* Herschel González. <https://herschelgonzalez.com/desventajas-de-bases-de-datos-nosql/> Consultado: 2025, 6 febrero