

Modelo de Datos

- Modelo Orientado a Objetos

El modelo de base de datos orientada a objetos agrupa la información en paquetes relacionados entre sí: los datos de cada registro se combinan en un solo objeto, con todos sus atributos. De esta manera, toda la información está disponible en el objeto, ya que sus datos quedan agrupados en lugar de distribuidos en diferentes tablas. En los objetos no solo pueden guardarse los atributos, sino también los métodos, lo que refleja la afinidad de estas bases de datos con los **lenguajes de programación orientados a objetos**: al igual que en estos, cada objeto presenta un conjunto de acciones que pueden llevarse a cabo.[2]

Los manejadores de bases de datos orientados a objetos deben tomar en cuenta las siguientes operaciones:

- Ser capaces de definir sus propios tipos de datos.
- El tamaño de los datos puede ser muy grande.
- La duración de las transacciones puede ser muy larga.
- Recuperar rápidamente objetos complejos.
- Lenguajes de consulta de objetos, un ejemplo es OQL (Object Query Language).
- Mecanismos de seguridad basados en la noción de objeto.
- Funciones para definir reglas deductivas. [1]

Ventajas	Desventajas
Los conjuntos de datos complejos pueden guardarse y consultarse de forma rápida y sencilla.	El uso de las bases de datos orientadas a objetos no está muy extendido.
Los códigos de identificación se asignan automáticamente a cada objeto.	En algunas situaciones, la gran complejidad puede acarrear problemas de rendimiento.
Funciona bien con lenguajes de programación orientados a objetos.	

[2]

- Modelo NoSQL

El término "NoSQL" se refiere a tipos de bases de datos no relacionales que almacenan datos en un formato distinto a las tablas relacionales. Sin embargo, las bases de datos NoSQL se pueden consultar utilizando API de lenguaje natural, lenguajes de consulta estructurados declarativos y lenguajes de consulta mediante ejemplo, por lo que también se les llama bases de datos "no solo SQL".[3]

Las bases de datos NoSQL se utilizan de forma generalizada en aplicaciones web en tiempo real y big data, ya que sus principales ventajas son los elevados niveles de escalabilidad y disponibilidad.[3]

Las bases de datos NoSQL también son la opción preferida entre los desarrolladores, ya que propician por su propia naturaleza un desarrollo ágil dada su rápida adaptación a los requisitos en constante cambio. Las bases de datos NoSQL permiten almacenar los datos de forma más intuitiva y fácil de entender, o más cercanas a la forma en que las aplicaciones utilizan los datos: no necesitan realizar tantas transformaciones cuando almacenan o recuperan datos utilizando interfaces API de NoSQL. Además, las bases de datos NoSQL pueden aprovechar al máximo la nube para evitar por completo el tiempo de inactividad.[3]

Ventajas	Desventajas
Flexibilidad: pueden almacenar de forma más libre datos sin la rigidez de esquemas. Este diseño impulsa la innovación y el rápido desarrollo de aplicaciones	Falta de estandarización: A diferencia de SQL, donde el lenguaje y las operaciones son bastante estándar, en el mundo NoSQL existen muchas soluciones con diferentes formas de operación y consulta.
Escalabilidad: en lugar de escalar verticalmente agregando más servidores, las bases de datos NoSQL pueden escalar horizontalmente utilizando hardware básico	Bases de datos sin esquemas específicos (estructura no fija, por ejemplo)
Alto rendimiento: esta arquitectura garantiza tiempos de respuesta rápidos y predecibles de milisegundos de un solo dígito. Las bases de datos NoSQL también pueden ingerir datos y entregarlos de forma rápida y fiable.	Necesidad de múltiples búsquedas de lectura, todos los datos necesarios se pueden recuperar efectivamente de una vez sin combinación en particular.
Disponibilidad: las bases de datos NoSQL replican datos de forma automática en múltiples servidores, centros de datos o recursos en la nube.	Grandes conjuntos de datos

Altamente funcional: las bases de datos NoSQL están diseñadas para almacenes de datos distribuidos que precisan una capacidad de almacenamiento de datos extremadamente grande.	
---	--

[3][4]

Referencias

- [UNAM, «Modelo Orientado a Objetos,» CUAED, 2017. [En línea]. Available:
1 https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/782/mod_resource/content/8/contenido/index.html. [Último acceso: 1 febrero 2024].
- [IONOS, «Base de datos orientada a objetos: el secreto mejor guardado de los
2 modelos de bases de datos,» 18 enero 2023. [En línea]. Available:
] <https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/base-de-datos-orientada-a-objetos/>. [Último acceso: 1 febrero 2024].
- [ORACLE, «¿Qué es NoSQL?,» [En línea]. Available:
3 <https://www.oracle.com/mx/database/nosql/what-is-nosql/#:~:text=Las%20bases%20de%20datos%20NoSQL%20permiten%20almacenar%20los%20datos%20de,utilizando%20interfaces%20API%20de%20NoSQL..>
] [Último acceso: 1 febrero 2024].
- [DataScientest, «SQL vs NoSQL : diferencias, usos, ventajas y inconvenientes,»
4 2023. [En línea]. Available: <https://datascientest.com/es/sql-vs-nosql-diferencias-usos-ventajas-y-inconvenientes>. [Último acceso: 1 febrero 2024].