

# Tarea 1. Modelos Orientados a Objetos y NoSQL para Bases de Datos

*Lemus González, Javier Issac*  
[javierissaclemusgonzalez@gmail.com](mailto:javierissaclemusgonzalez@gmail.com)  
 Universidad Nacional Autónoma de México

## I. MODELO ORIENTADO A OBJETOS

Las bases de datos orientadas a objetos (BDOO) representan una alternativa a las bases de datos relacionales, especialmente en situaciones donde la gestión de datos complejos o no convencionales se vuelve impracticable con las estructuras relacionales tradicionales, como en el caso de bases de datos multimedia, científicas y sistemas de apoyo al diseño industrial.

El modelo de datos adoptado por las BDOO se basa en los principios de los lenguajes orientados a objetos, lo que permite el uso de estructuras de datos complejas y elimina las barreras entre el desarrollo de aplicaciones y la gestión de datos. En este modelo, todos los elementos manejados son objetos, cada uno correspondiente a una entidad del mundo real con identidad única, propiedades incorporadas y comportamiento definido.

La identidad única de un objeto se representa mediante un Object Identifier (OID), asignado por el sistema en el momento de la creación del objeto y no sujeto a modificaciones. A diferencia de una clave primaria, que se basa en valores dados por los usuarios y puede cambiar, el OID garantiza la unicidad.

Los objetos en una BDOO tienen asignado un valor que expresa sus propiedades, ajustándose a una estructura de datos que puede ser tan compleja como sea necesario. Los tipos de datos básicos se combinan mediante constructores de tipos de datos y tipos de objetos, permitiendo la extensibilidad con nuevos tipos de datos complejos.

Los constructores se dividen en átomos (elementos de dominios básicos) y colecciones (tuplas, conjuntos y listas). Los atributos de un objeto pueden tener un dominio asignado, ya sea un tipo de datos o un tipo de objeto previamente definido.

El comportamiento de los objetos se representa mediante operaciones implementadas sobre ellos. Cada operación tiene una signatura pública y un método oculto, desarrollado en un lenguaje de programación específico. Estas operaciones representan acciones del mundo real y pueden ser utilizadas para crear y eliminar objetos, cambiar valores de atributos y realizar operaciones más complejas.

Las BDOO clasifican objetos en clases, conjuntos de objetos que comparten propiedades y comportamiento. La instanciación es el proceso de generar objetos de una clase, y cada objeto en la base de datos es una instancia de una clase. El conjunto de instancias de una clase se denomina extensión y cuenta con un nombre explícito mediante la cláusula extent.

La interfaz o protocolo de acceso de una clase define su conjunto de operaciones públicas, mientras que los cuerpos de los métodos se implementan utilizando lenguajes de programación orientados a objetos. La flexibilidad de las BDOO radica en su capacidad para admitir tipos de datos multimedia y texto, así como permitir a los usuarios definir sus propios tipos de datos básicos y operaciones.

En resumen, las BDOO ofrecen una aproximación más flexible y eficiente para gestionar datos complejos en comparación con las bases de datos relacionales convencionales. Sin embargo, las bases de datos orientadas a objetos presentan desafíos. La implementación puede ser compleja y requerir una curva de aprendizaje. La falta de un estándar universal dificulta la interoperabilidad, y el rendimiento puede ser inferior en comparación con las bases de datos relacionales. Además, las BDOO suelen requerir más recursos, lo que implica costos adicionales. Aunque ofrecen beneficios, la adopción limitada también puede afectar su aplicabilidad en diversos contextos.

## II. MODELOS NoSQL

La evolución de los paradigmas de gestión de datos ha llevado al surgimiento de las bases de datos NoSQL (Not Only SQL), una alternativa a las tradicionales bases de datos relacionales. Estas bases de datos NoSQL se caracterizan por su capacidad para manejar grandes volúmenes de datos con flexibilidad y escalabilidad, adaptándose a entornos donde la consistencia inmediata puede ser sacrificada en favor de la disponibilidad y la tolerancia a particiones.

En este contexto, es esencial comprender los principios fundamentales de las bases de datos NoSQL y los distintos modelos que ofrecen. Uno de los criterios clave es la persistencia, que se refiere a la capacidad de mantener los datos en un estado permanente, garantizando que no se pierdan incluso en situaciones de fallo. La replicación, por otro lado, implica almacenar réplicas de la base de datos en varios nodos para asegurar la disponibilidad continua. Sin embargo, la

replicación puede plantear desafíos de consistencia que deben abordarse.

La fragmentación es otra característica importante, permitiendo la división de datos en varios nodos para un procesamiento rápido. No todas las bases de datos NoSQL ofrecen esta capacidad, ya que algunos sistemas requieren que los usuarios realicen manualmente la partición de datos. La consistencia eventual es un concepto clave en las bases de datos NoSQL, permitiendo múltiples accesos simultáneos para escribir y leer datos, con la garantía de que los datos serán consistentes eventualmente.

El Teorema del CAP, que aborda la consistencia, la disponibilidad y la tolerancia a particiones, establece que ningún sistema distribuido puede cumplir plenamente con los tres criterios. En cambio, las bases de datos NoSQL se dividen en tres categorías según cómo gestionan estos criterios: CP (Consistent & Partition Tolerance), AP (Availability & Partition Tolerance) y CA (Consistent & Availability).

ACID (Atomicity, Consistency, Isolation, Durability) y BASE (Basically Available, Soft state, Eventually consistent) son dos conjuntos de propiedades que definen los enfoques de las bases de datos relacionales y NoSQL, respectivamente. Mientras que ACID se centra en la integridad de las transacciones, BASE se centra en la disponibilidad y la consistencia eventual, proporcionando un marco más flexible para entornos NoSQL.

En la actualidad, las bases de datos NoSQL abordan diversas necesidades, desde estructuradas hasta semiestructuradas y no estructuradas. Están diseñadas para manejar big data con alta variabilidad, adaptándose a la diversidad de datos provenientes de múltiples fuentes. Estas bases de datos permiten la escalabilidad horizontal y son ideales para aplicaciones con bases de datos jerárquicas, como las que almacenan datos en una estructura de árbol.

Dentro de los modelos de bases de datos NoSQL, destacan tres tipos principales: clave-valor, documentales y orientadas a grafos. Las bases de datos **clave-valor** son tablas hash simples, ideales para acceso mediante una clave primaria, como en la gestión de sesiones de usuarios. Ejemplos incluyen Riak, Redis, Voldemort e Infinispan. Se utilizan en aplicaciones de ingesta rápida de datos, almacenamiento en caché y detección de fraudes.

Las bases de datos clave-valor ofrecen una solución eficiente para casos de uso que requieren un alto rendimiento y una escalabilidad fácil. Un ejemplo común es la gestión de sesiones de usuarios en aplicaciones web. Las sesiones de usuario implican el almacenamiento y recuperación de datos de interacción del usuario, como historial de navegación y preferencias, durante la duración de su sesión. Las bases de datos clave-valor, con su capacidad para recuperar rápidamente datos mediante una clave primaria, son ideales para esta tarea. Al almacenar los datos de sesión, las claves pueden representar identificadores de usuarios, y los valores contienen información relevante para personalizar la experiencia del usuario.

Otro caso de uso es la gestión de anuncios en páginas web. En este contexto, las bases de datos clave-valor pueden almacenar información sobre anuncios dirigidos a usuarios específicos. La clave puede contener detalles como el país de origen, el idioma y las preferencias de compra, mientras que el valor almacena el contenido del anuncio. Esta estructura facilita la recuperación rápida de anuncios personalizados basados en diferentes criterios, lo que es esencial para la eficacia de la publicidad en línea.

Las bases de datos **documentales** almacenan y recuperan documentos, como XML, JSON o BSON. MongoDB y CouchDB son ejemplos notables. Su flexibilidad y capacidad para manejar datos no estructurados las hace ideales para aplicaciones móviles, videojuegos y gestión de contenidos.

Un caso de uso para este tipo es la tecnología blockchain, donde se utiliza para gestionar identidades digitales en una red bancaria ficticia. En este escenario, la base de datos documental puede almacenar de manera eficiente información sobre identidades digitales, transacciones y otras actividades relacionadas. La cadena de bloques, como un gráfico de conocimiento, representa las relaciones entre entidades y ofrece un registro seguro y verificable de todas las transacciones.

Otro caso de uso es el análisis en tiempo real, donde las bases de datos documentales permiten el procesamiento continuo de datos. En entornos como cajeros automáticos, servicios al cliente y sistemas de punto de venta, estas bases de datos facilitan la entrada, procesamiento y salida continuos de datos. La capacidad de almacenar y recuperar documentos de manera eficiente es esencial para lograr resultados analíticos casi instantáneos, como en el análisis de datos de vehículos para transporte y gestión inteligente del tráfico.

Las bases de datos **orientadas a grafos** permiten almacenar entidades y relaciones, destacando Neo4J e InfiniteGraph. Estas bases son fundamentales para casos como grafos sociales, conocimiento graph, detección de fraude y recomendaciones de carritos de compra.

Un caso de uso podría ser el análisis de redes sociales, donde estas bases de datos son fundamentales para almacenar y consultar eficientemente relaciones entre usuarios, sus interacciones y sus intereses comunes. La representación de la red social como un grafo facilita la identificación de comunidades, la recomendación de amigos y la detección de patrones de comportamiento.

Otro caso de uso es la gestión de conocimiento, donde las bases de datos orientadas a grafos son utilizadas para modelar y explorar relaciones entre conceptos. Por ejemplo, en un sistema de recomendación de contenido educativo, los nodos pueden representar temas de aprendizaje, y las relaciones entre ellos indican la relevancia o la secuencia de temas. Esto facilita la navegación y la recomendación personalizada de contenido educativo basada en la estructura de conocimiento.

## REFERENCIAS

- [1] M. J. Aramburu Cabo, Bases de datos avanzadas. Univ. Jaume I, 2013. Accedido el 2 de febrero de 2024. [En línea]. Disponible: <https://doi.org/10.6035/sapientia73J>.
- [2] J. Soriano González, "Estudio sobre herramientas NoSQL," 2021. Accedido el 2 de febrero de 2024. [En línea]. Disponible: <https://dspace.umh.es>.