



# Universidad Nacional Autónoma de México

## Facultad de Ingeniería

**Asignatura: Bases de Datos (1644)**

*Profesor: Ing. Fernando Arreola Franco*

Semestre 2026-2

**“Tarea 1”**

Grupo: 1

Alumna: Bautista Reyes Sofía

No. de cuenta: 321103891

Fecha de entrega: 6 de febrero de 2026

# Tarea 1

Investigar:

- **Modelo orientado a objetos:**

Las bases de datos orientadas a objetos son una alternativa que es menos utilizada en comparación a las bases de datos relacionales; en este modelo, toda la información se almacena en objetos que agrupan todos sus atributos y métodos (estos organizados en clases y subclases), algo propio de la lógica en la que se basa la programación orientada a objetos.

Aquí cada objeto tiene un identificador único que permite acceder directamente a toda su información y a diferencia de las bases de datos relacionales que distribuyen los datos en tablas y en relaciones, las bases de datos de objetos permiten manejar registros más complejos de forma directa y a veces con un mejor rendimiento, especialmente cuando se trabaja con lenguajes como Java.

Por su forma de organización, entre sus principales ventajas está la facilidad de integración con la programación orientada a objetos y la rapidez en consultas complejas. Sin embargo como desventajas, tiene la escasa adopción, la poca variedad de SGBD disponibles y una menor comunidad y documentación, lo que hace que se usen con mayor frecuencia las bases de datos relacionales en la mayoría de los proyectos [1], [2].

Por ejemplo, en una tienda de electrodomésticos, una lavadora puede representarse como un objeto dentro de la clase “lavadoras”, asociada a la categoría de pequeños electrodomésticos, y este objeto puede incluir características como color, potencia y dimensiones, facilitando la gestión de información compleja en una sola estructura [3].

- **Modelos NoSQL (clave-valor, documentales, grafos y columnar), descripción, ventajas, desventajas, casos de uso...**

Los modelos NoSQL a diferencia del modelo relacional, no tienen un esquema rígido, lo que permite adaptar la estructura de los datos de acuerdo a lo que pide cada aplicación. Esta flexibilidad ayuda mucho en casos donde los datos son altamente variables, no estructurados o que se generan de forma continua.

El objetivo principal de este modelo es optimizar el rendimiento de las consultas y la escalabilidad del sistema y para hacerlo, se hace uso de datos desnormalizados y estructuras diseñadas en función de los patrones de acceso, reduciendo la necesidad de operaciones complejas como uniones entre las tablas. También su arquitectura distribuida facilita el manejar grandes volúmenes de datos de manera eficiente [4].

Dentro de los modelos NoSQL existen varios enfoques de almacenamiento, cada uno para un tipo específico de información:

Los modelos **clave-valor** destacan por su simplicidad y velocidad, esto se debe a que almacenan la información en pares formados por una clave única y un valor asociado. Esta clave se utiliza para acceder directamente al dato, sin necesidad de conocer la estructura interna del valor, lo que permite realizar operaciones de lectura y escritura muy rápidas. Sin embargo, estos modelos no permiten realizar consultas complejas ni búsquedas avanzadas sobre los datos almacenados, ya que todo se recupera únicamente

a partir de la clave. También pueden presentar problemas de integridad y consistencia de los datos, especialmente en sistemas distribuidos.

Como casos de uso estos modelos se utilizan principalmente en almacenamiento en caché, manejo de sesiones, perfiles de usuario, análisis en tiempo real y catálogos de productos, donde se necesita acceso rápido a los datos [5].

Los modelos **documentales**, como su nombre lo indica, organizan la información en colecciones formadas por documentos que manejan pares clave-valor, comúnmente en formatos como JSON, y cada documento puede tener una estructura distinta, lo que facilita la adaptación a cambios en el esquema sin afectar al resto de la colección.

Estos modelos facilitan el desarrollo rápido de aplicaciones gracias a que su esquema flexible permite adaptar la estructura de los datos conforme cambian los requisitos y también ofrecen lenguajes de consulta y APIs completas, así como capacidad de escalamiento y alta disponibilidad, lo que las convierte en una muy buena opción. Aunque tiene la limitación de que muchas de estas bases de datos no admiten transacciones ACID entre múltiples documentos, sin embargo, en la mayoría de las aplicaciones esta funcionalidad no es necesaria, y algunas bases de datos sí la soportan (MongoDB).

Como casos de uso estos modelos se emplean en gestión de datos de clientes, catálogos de productos, sistemas transaccionales y analíticos de propósito general, análisis operativo y en tiempo real, y en aplicaciones móviles [6].

Los modelos orientados a **grafos** se enfocan en representar relaciones entre los datos, este enfoque permite almacenar y analizar de manera eficiente la información que está altamente conectada. Estas bases de datos no tienen esquemas, lo que permite una gran flexibilidad parecida a la de un modelo clave-valor.

Destacan por su alto rendimiento al manejar grandes volúmenes de datos altamente relacionados y ofrecen gran flexibilidad para analizar conexiones complejas, también permiten una rápida recuperación de información, lo que las hace ideales para entornos donde se requieren decisiones en tiempo real. Aunque tiene dificultades relacionadas con la atomicidad (propiedad donde cada instrucción de una transacción se trata como una sola unidad) y la estandarización, y en ciertos esquemas de partición pueden surgir problemas similares a los de las bases de datos relacionales [7].

Estos modelos se utilizan en detección de fraude, recomendaciones en tiempo real en redes sociales y en la gestión de grandes volúmenes de datos, donde es necesario analizar y optimizar relaciones complejas entre entidades [8].

Un modelo de base de datos **columnar**, también como su nombre lo sugiere, organiza la información por columnas en lugar de filas, lo que permite acelerar el procesamiento de consultas, esto es útil cuando se manejan grandes volúmenes de información y se requiere un acceso rápido a ciertos datos, como en aplicaciones de aprendizaje automático.

Entre sus características está la mejora en el rendimiento de las consultas, gracias a técnicas de compresión que reducen el espacio de almacenamiento y agilizan operaciones como sumas o mínimos, y permiten un análisis más rápido de grandes conjuntos de datos, tienen autoindexación, procesan datos de forma vectorizada y evitan almacenar

valores nulos, optimizando el uso del almacenamiento.

Este tipo de modelo es más eficiente cuando las consultas solo requieren algunas columnas específicas, por lo que se utilizan ampliamente en analíticas empresariales, monitoreo de aplicaciones y seguridad, y donde los datos de sensores se analizan para detectar patrones, anomalías o mejorar el rendimiento de los sistemas [9], [10].

## Referencias

- [1] Equipo editorial de IONOS. (2023) Base de datos orientada a objetos: el secreto mejor guardado de los modelos de bases de datos. IONOS. [Online]. Available: <https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/base-de-datos-orientada-a-objetos/>
- [2] CUAED UNAM. (s.f.) Modelo orientado a objetos. [Online]. Available: [https://repositorio-uapa.cuaed.unam.mx/repositorio/moodle/pluginfile.php/2731/mod\\_resource/content/1/UAPA-Modelo-Orientado-Objetos/index.html](https://repositorio-uapa.cuaed.unam.mx/repositorio/moodle/pluginfile.php/2731/mod_resource/content/1/UAPA-Modelo-Orientado-Objetos/index.html)
- [3] Ángel DevOps Consultant. (2023) Base de datos orientada a objetos, ¿qué es y cómo beneficia a tu organización? Incentro. Consultado el 6 de febrero de 2026. <https://www.incentro.com/es-ES/blog/que-es-una-base-de-datos-orientada-a-objetos>.
- [4] A. Rodríguez. (2024) Bases de datos nosql: Guía definitiva. [Online]. Available: <https://pandorafms.com/blog/es/bases-de-datos-nosql/>
- [5] Couchbase. (s.f.) Base de datos clave-valor. [Online]. Available: <https://www.couchbase.com/es/resources/concepts/key-value-database/>
- [6] MongoDB. (s.f.) ¿Qué es una base de datos documental? [Online]. Available: <https://www.mongodb.com/resources/basics/databases/document-databases>
- [7] Databricks. (s.f.) Transacciones acid. [Online]. Available: <https://www.databricks.com/glossary/acid-transactions#:~:text=Atomicity%20%2D%20each%20statement%20in%20a,data%20fails%20mid%2Dstream>
- [8] Graph Everywhere. (s.f.) Bases de datos nosql | bases de datos de grafos. [Online]. Available: <https://www.grapheverywhere.com/nosql-de-grafos/>
- [9] F. G. de Zúñiga. (s.f.) Bases de datos nosql: qué son, tipos y ventajas. arsys. [Online]. Available: <https://www.arsys.es/blog/bases-de-datos-nosql-que-son-tipos-y-ventajas#tree-2>
- [10] Pure Storage. (s.f.) ¿Qué es una base de datos Columnar? [Online]. Available: <https://www.purestorage.com/es/knowledge/what-is-a-columnar-database.html>