



**Universidad Nacional Autónoma de México**  
**Facultad de Ingeniería**

**Tarea 1**

**Bases de Datos (1644)**

**Profesor: Ing. Fernando Arreola Franco**

**Semestre 2026-2**

Grupo: 1

Alumna: Cruz Basilio Ximena Carolina

No. de cuenta: 321116424

Fecha de entrega: 6 de febrero de 2026

## I. MODELO ORIENTADO A OBJETOS

La orientación a objetos es un paradigma que busca representar el mundo real a través de objetos, los cuales pueden ser físicos o digitales. Desde este enfoque, un sistema se entiende como un conjunto de componentes que interactúan entre sí, ya que “un sistema sólo se define por sus componentes y la manera en que éstos interactúan”. Este paradigma permite modelar sistemas de manera clara y cercana a la realidad.

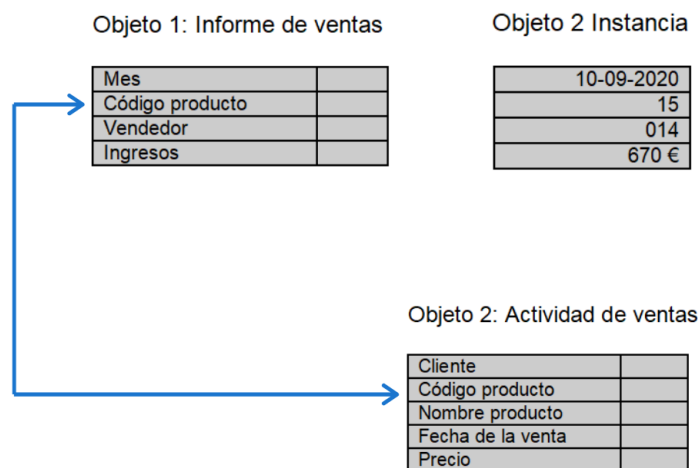
La orientación a objetos se basa en la idea de que todo sistema está formado por objetos. Cada objeto tiene un nombre, atributos y operaciones, además de poseer un estado, un comportamiento y una identidad. Esto permite que cada elemento del sistema sea único y fácilmente identificable.

Este enfoque utiliza conceptos como clases, herencia y polimorfismo, los cuales facilitan la creación de sistemas más organizados, reutilizables y fáciles de mantener. Además, permite crear modelos que reflejan un dominio específico utilizando la terminología propia del área de aplicación.

El modelo de datos orientado a objetos organiza la información utilizando objetos en lugar de tablas. En este modelo, los datos y las relaciones se almacenan juntos dentro de una misma estructura. Este modelo sirve como base para los sistemas de gestión de bases de datos orientados a objetos y ofrece una mayor flexibilidad frente a las bases de datos relacionales tradicionales.

Una de sus principales ventajas es que permite manejar datos complejos, ya que “los OODM permiten almacenar tipos de datos complejos, como imágenes, audio y video”, los cuales son difíciles de gestionar mediante tablas.

Los modelos de datos orientados a objetos son especialmente útiles en aplicaciones que manejan información compleja, como sistemas multimedia, comercio electrónico, inteligencia artificial y sistemas de ingeniería. Por ejemplo, en un sistema de comercio electrónico, los productos y los vendedores pueden representarse como objetos relacionados, lo que permite mostrar información del vendedor y filtrar productos de forma eficiente.



## II. TIPOS DE BASES DE DATOS NOSQL

### II-A. Almacén Clave-Valor

El modelo clave-valor es uno de los más simples dentro de las bases de datos NoSQL. En este modelo, cada dato se almacena como un par compuesto por una clave única y su valor correspondiente. Su principal ventaja es la rapidez en el acceso a la información.

Este tipo de base de datos se utiliza comúnmente para almacenar información temporal, como sesiones de usuario, configuraciones o carritos de compra. Sin embargo, no es ideal para realizar consultas complejas sobre los datos almacenados.

#### II-A1. Ventajas:

1. **Simplicidad:** Utilizan un modelo de datos sencillo que reduce la complejidad en el diseño de la base de datos y en las consultas.
2. **Alto rendimiento:** Están optimizadas para operaciones rápidas de lectura y escritura, lo que permite un acceso veloz a los datos.
3. **Escalabilidad:** Permiten escalar horizontalmente agregando nodos para manejar mayores volúmenes de datos y tráfico.
4. **Flexibilidad:** No requieren un esquema fijo, lo que facilita la adaptación a cambios en la estructura de los datos.
5. **Almacenamiento eficiente:** Son ideales para almacenamiento en caché gracias a su acceso directo a los datos.

#### II-A2. Desventajas:

1. **Consultas limitadas:** No son adecuadas para consultas complejas que involucren múltiples relaciones.
2. **Integridad de los datos:** Mantener la coherencia y la integridad puede ser complicado, especialmente en sistemas distribuidos.
3. **Curva de aprendizaje:** El paradigma NoSQL puede resultar difícil de adoptar para quienes están acostumbrados a bases de datos relacionales.

#### II-A3. Casos de uso de bases de datos clave-valor:

Las bases de datos clave-valor se utilizan ampliamente en aplicaciones modernas debido a su simplicidad, velocidad y capacidad de escalabilidad. A continuación, se describen algunos de sus principales casos de uso.

##### 1. Almacenamiento en caché

Las bases de datos clave-valor son ideales para el almacenamiento en caché, ya que permiten acceder de forma rápida a los datos que se consultan con mayor frecuencia. Esto mejora el rendimiento del sistema y reduce los tiempos de respuesta de las aplicaciones.

2. **Perfiles de usuario** Este tipo de bases de datos se emplea para almacenar información básica de los usuarios, como nombre, correo electrónico y preferencias. Su estructura sencilla facilita la consulta y actualización de los datos, incluso cuando existe una gran cantidad de usuarios.
3. **Almacenamiento de sesión** Las bases de datos clave-valor son muy eficientes para la gestión de sesiones de usuario. Permiten guardar y actualizar información relacionada con inicios de sesión, autenticaciones e interacciones, garantizando un acceso rápido y continuo.

4. **Análisis en tiempo real** Gracias a su alta velocidad de acceso, estas bases de datos son adecuadas para aplicaciones que requieren análisis inmediato de la información, como precios dinámicos, recomendaciones personalizadas o evaluaciones en tiempo real.
5. **Catálogos de productos** En el comercio electrónico, las bases de datos clave-valor permiten administrar catálogos de productos con múltiples atributos. Su flexibilidad facilita la actualización de la información y la recuperación rápida de los datos cuando los usuarios realizan búsquedas.

## *II-B. Documentales*

Las bases de datos orientadas a documentos almacenan la información en forma de documentos, generalmente en formatos como JSON o XML. Cada documento puede tener una estructura distinta, lo que proporciona una gran flexibilidad en el manejo de los datos.

Este modelo es muy utilizado en aplicaciones web, sistemas de gestión de contenido, comercio electrónico y perfiles de usuario, ya que permite modificar la estructura de los datos sin afectar a otros documentos.

### *II-B1. : Ventajas*

1. Permiten almacenar y consultar información semiestructurada sin necesidad de definir una estructura fija, lo que facilita el manejo de datos variados.
2. Ofrecen un modelo muy flexible, capaz de albergar distintos tipos de datos dentro de una misma base de datos.
3. Simplifican las tareas de adición y actualización de datos, lo cual es especialmente útil en aplicaciones web y móviles que cambian constantemente.
4. Priorizan la rapidez en la escritura de los datos, garantizando alta disponibilidad incluso ante fallos de hardware o de red.
5. Proporcionan un buen rendimiento gracias a sus motores de búsqueda y sistemas avanzados de indexación.
6. Presentan una alta escalabilidad, siendo adecuadas para el almacenamiento de grandes volúmenes de información.

### *II-B2. : Desventajas*

1. No utilizan SQL como lenguaje estándar de consulta, ya que cada sistema puede manejar su propio método de acceso a los datos.
2. No siempre garantizan las propiedades ACID, lo que puede afectar la consistencia e integridad de los datos.
3. Existe una menor comunidad y menos documentación disponible en comparación con las bases de datos relacionales.
4. El uso de índices puede consumir una gran cantidad de memoria RAM, especialmente cuando se manejan grandes volúmenes de datos.

### *II-B3. Casos de uso de bases de datos documentales:*

Las bases de datos de documentos son bases de datos de propósito general que sirven para una variedad de casos de uso tanto para aplicaciones transaccionales como analíticas:

1. **Vista única o centro de datos**

2. **Gestión y personalización de datos de clientes**
3. **Internet de las cosas (IoT) y datos de series temporales**
4. **Catálogos de productos y gestión de contenidos**
5. **Procesamiento de pagos**
6. **Aplicaciones móviles**
7. **Descarga del mainframe**
8. **Análisis operativo**
9. **Análisis en tiempo real**

## *II-C. Bases de Datos Columnares*

Las bases de datos orientadas a columnas almacenan la información agrupándola por columnas en lugar de filas. Esto permite un acceso más eficiente cuando se realizan consultas que involucran grandes volúmenes de datos.

Son especialmente útiles en sistemas de análisis de datos, minería de información y detección de patrones, aunque su implementación y mantenimiento pueden resultar más complejos que otros modelos NoSQL.

### *II-C1. Ventajas:*

1. **Alto rendimiento en análisis de datos:** Las bases de datos columnares son muy eficientes cuando se trabajan grandes volúmenes de información, ya que permiten leer únicamente las columnas necesarias. Esto reduce el acceso al disco, que suele ser el principal cuello de botella en las bases de datos.
2. **Optimización para big data:** Este tipo de bases de datos es ideal para entornos de big data y análisis intensivo, donde se requieren evaluaciones rápidas sobre grandes conjuntos de datos históricos.
3. **Alta capacidad de compresión:** Al almacenar datos del mismo tipo en una sola columna, es posible aplicar técnicas de compresión más eficientes, lo que reduce el espacio de almacenamiento y mejora la velocidad de lectura.

### *II-C2. Desventajas:*

1. **Bajo rendimiento en aplicaciones transaccionales:** En aplicaciones donde se realizan muchas inserciones o actualizaciones, los datos deben distribuirse entre varias columnas, lo que hace que las bases de datos columnares sean más lentas que las bases de datos tradicionales orientadas a filas.
2. **No ideales para operaciones frecuentes de escritura:** Este modelo prioriza la lectura y el análisis de datos, por lo que no es la mejor opción cuando se requiere modificar información constantemente.

*II-C3. Casos de uso de las bases de datos columnares:* Las bases de datos columnares son especialmente útiles cuando se realizan consultas que solo requieren analizar algunas columnas específicas de grandes volúmenes de datos. A diferencia de las bases de datos tradicionales, este modelo agrupa la información por columnas, lo que mejora el rendimiento y reduce el tiempo de respuesta en consultas analíticas.

#### **1. Analíticas empresariales**

En el análisis de información empresarial, normalmente solo se necesitan ciertos datos clave, como ventas, ingresos o métricas de desempeño. Las bases de datos columnares permiten consultar estas

columnas de forma rápida, lo que las hace ideales para generar reportes, predicciones y modelos de análisis o aprendizaje automático.

## 2. Supervisión de seguridad y aplicaciones

Las bases de datos columnares se utilizan para almacenar eventos generados por aplicaciones, como errores, accesos fallidos o tiempos de respuesta. Estos datos pueden analizarse de manera eficiente para detectar fallos, mejorar el rendimiento del sistema o identificar posibles ciberataques en tiempo real.

## 3. Internet de las Cosas (IoT)

En sistemas de IoT, los sensores recopilan grandes cantidades de datos que suelen almacenarse en columnas específicas, como temperatura, presión o ritmo cardíaco. Las bases de datos columnares facilitan el análisis de estos datos para detectar anomalías en maquinaria industrial o cambios importantes en la salud de las personas.

### II-D. Bases de Datos de Grafos

Las bases de datos de grafos están diseñadas para representar y analizar relaciones entre datos. Utilizan nodos para representar entidades y aristas para representar las conexiones entre ellas.

Este tipo de base de datos es ideal para redes sociales, sistemas de recomendación y análisis de relaciones complejas, ya que permite visualizar y recorrer fácilmente las conexiones entre los datos.

#### II-D1. Ventajas:

Las bases de datos orientadas a grafos ofrecen múltiples ventajas, especialmente cuando se trabaja con información altamente relacionada y en constante crecimiento.

1. **Rendimiento** Las bases de datos en grafos están diseñadas para manejar grandes volúmenes de datos y, sobre todo, un número elevado de relaciones entre ellos. A medida que los datos crecen, estas bases mantienen un buen desempeño, lo que las hace adecuadas para entornos donde la información aumenta de forma rápida y continua.
2. **Flexibilidad** Este tipo de bases de datos permite modelar relaciones complejas de manera natural. Gracias a su estructura basada en nodos y conexiones, es posible realizar distintos tipos de análisis y adaptarse fácilmente a los retos que presentan las aplicaciones digitales actuales.
3. **Rapidez** Las bases de datos en grafos ofrecen tiempos de respuesta muy rápidos, ya que las relaciones están almacenadas directamente. Esto es una gran ventaja en sistemas donde se requieren decisiones inmediatas, como motores de recomendación o redes sociales.

#### II-D2. Desventajas :

Aunque las bases de datos en grafos presentan múltiples beneficios, también cuentan con algunas limitaciones que deben considerarse.

1. **Limitaciones en la atomicidad** Al igual que otras bases de datos NoSQL, las bases de datos en grafos pueden presentar dificultades para garantizar completamente la atomicidad de las operaciones, lo que puede afectar la consistencia de los datos en ciertos escenarios.
2. **Falta de estandarización** No existe un estándar único para el manejo y consulta de bases de datos en grafos, lo que puede generar una curva de aprendizaje mayor y dependencia de herramientas específicas.

3. **Problemas en la partición de datos** Cuando se realizan particiones basadas en propiedades locales, pueden surgir dificultades similares a las de las bases de datos relacionales, afectando el rendimiento o la gestión de las relaciones.

## *II-E. Casos de uso de las bases de datos orientadas a grafos*

Las bases de datos en grafos resultan especialmente útiles en proyectos donde las relaciones entre los datos son tan importantes como los datos mismos. Gracias a su estructura basada en nodos y conexiones, permiten analizar información compleja de manera eficiente y clara.

### **1. Detección de fraude**

En el ámbito financiero y bancario, las bases de datos en grafos se utilizan para identificar patrones sospechosos entre usuarios, transacciones y cuentas. Este modelo permite detectar relaciones irregulares que serían difíciles de identificar con otros tipos de bases de datos, ayudando a prevenir fraudes de forma más precisa.

### **2. Recomendaciones en tiempo real en redes sociales**

Las bases de datos en grafos facilitan el análisis de las relaciones entre personas, intereses y comportamientos. Esto permite a las redes sociales ofrecer recomendaciones personalizadas en tiempo real, como sugerencias de contactos o contenido relevante, además de ayudar a las empresas a mejorar sus productos y servicios según los intereses de los usuarios.

### **3. Gestión de grandes volúmenes de datos**

El crecimiento constante de datos generados por usuarios y dispositivos puede saturar los sistemas tradicionales. Las bases de datos en grafos permiten gestionar y optimizar redes físicas y virtuales de manera eficiente, facilitando el monitoreo y control de grandes volúmenes de información sin afectar el rendimiento del sistema.

## **III. VENTAJAS Y DESVENTAJAS DE NOSQL**

### *III-A. Ventajas*

Las bases de datos NoSQL ofrecen un modelo de datos flexible, ya que no requieren un esquema fijo y permiten almacenar distintos tipos de información. Esto facilita el desarrollo ágil y permite realizar cambios rápidos sin modificar toda la estructura de la base de datos. Otra ventaja importante es la escalabilidad horizontal, que permite aumentar la capacidad del sistema agregando más servidores sin interrumpir el servicio. Además, NoSQL proporciona alta disponibilidad, ya que los datos se replican en varios servidores, reduciendo el riesgo de fallos.

Finalmente, estas bases de datos permiten realizar consultas más rápidas, ya que evitan uniones complejas entre tablas, lo que las hace ideales para aplicaciones web, móviles, big data e Internet de las Cosas.

### *III-B. Desventajas*

Una de las principales desventajas de NoSQL es que no cuenta con un lenguaje estándar como SQL, ya que cada sistema utiliza su propio método de consulta. Además, algunas bases de datos NoSQL no

ofrecen el mismo nivel de consistencia e integridad de datos que las bases de datos relacionales. Asimismo, NoSQL no es la mejor opción para aplicaciones que requieren consultas complejas o muchas relaciones entre datos, ya que prioriza la velocidad y la disponibilidad sobre una coherencia estricta.

#### IV. CASOS DE USO DE LAS BASES DE DATOS NoSQL

Las bases de datos NoSQL se utilizan principalmente en aplicaciones que requieren alta velocidad, escalabilidad y la capacidad de manejar grandes volúmenes de datos. El tipo de base de datos NoSQL que se elija depende del problema que se desea resolver y del tipo de información que se va a gestionar. Cuando se necesita analizar relaciones complejas entre datos, como ocurre en redes sociales, motores de recomendación o sistemas de detección de fraude, se emplean bases de datos NoSQL orientadas a grafos. Este tipo de bases de datos permite representar de forma clara las conexiones entre personas, objetos o eventos. Casos de uso de SQL:

Transacciones financieras  
Análisis de datos de atención médica  
Información del cliente y de las transacciones  
En aplicaciones que requieren respuestas casi inmediatas, como videojuegos, plataformas de publicidad digital o aplicaciones en tiempo real, se utilizan bases de datos NoSQL en memoria. Estas bases de datos ofrecen baja latencia y pueden manejar un gran número de usuarios simultáneamente sin generar retrasos. Para escenarios donde existen grandes volúmenes de datos y picos elevados de tráfico, como en el comercio electrónico o el Internet de las Cosas (IoT), las bases de datos NoSQL de tipo clave-valor son una buena opción. Su estructura simple facilita el escalado rápido cuando aumenta la demanda de usuarios.

#### REFERENCIAS

- [1] Universidad Nacional Autónoma de México, “Modelo orientado a objetos,” UAPA / Repositorio UNAM, s.f., consultado: 5 de febrero de 2026. [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)
- [2] OWOX, “Object-oriented data models,” OWOX, 2023, consultado: 5 de febrero de 2026. [Online]. Available: <https://www.owox.com/blog/articles/object-oriented-data-models>
- [3] IBM, “¿qué es una base de datos nosql?” IBM, 2024, consultado: 5 de febrero de 2026. [Online]. Available: <https://www.ibm.com/mx-es/think/topics/nosql-databases>
- [4] Google Cloud, “¿qué es nosql?” Google Cloud, 2024, consultado: 5 de febrero de 2026. [Online]. Available: <https://cloud.google.com/discover/what-is-nosql?hl=es-419>
- [5] Couchbase, “Key-value database | conceptos de bases de datos clave-valor,” Couchbase, 2026, consultado: 8 de febrero de 2026. [Online]. Available: <https://www.couchbase.com/es/resources/concepts/key-value-database/>
- [6] MongoDB, “Document database - bases de datos orientadas a documentos,” MongoDB, 2026, consultado: 8 de febrero de 2026. [Online]. Available: <https://www.mongodb.com/es/resources/basics/databases/document-databases>
- [7] Ayuda Ley Protección Datos, “Bases de datos documentales: Qué son, tipos y ejemplos,” Ayuda Ley Protección Datos, 2026, consultado: 8 de febrero de 2026. [Online]. Available: <https://ayudaleyprotecciondatos.es/bases-de-datos/documentales/>
- [8] GraphEverywhere, “Bases de datos nosql | bases de datos de grafos,” GraphEverywhere, 2026, consultado: 8 de febrero de 2026. [Online]. Available: <https://www.grapheverywhere.com/nosql-de-grafos/>
- [9] Pure Storage, “¿qué es una base de datos columnar?” Pure Storage, 2026, consultado: 8 de febrero de 2026. [Online]. Available: <https://www.purestorage.com/es/knowledge/what-is-a-columnar-database.html>
- [10] IONOS, “Base de datos columnar (guía técnica),” IONOS, 2026, consultado: 8 de febrero de 2026. [Online]. Available: <https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/base-de-datos-columnar/>