



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Semestre 2026-2

BASES DE DATOS

Grupo 01

Ing. Fernando Arreola Franco

TAREA 1

FECHA DE ENTREGA: 8 de febrero de 2026

Parra Vera Héctor Jesús

CALIFICACIÓN: _____

1. Modelo de Base de Datos Orientado a Objetos (OODBMS)

Este modelo surgió en la década de 1980 impulsado por la necesidad de gestionar estructuras de datos complejas que los sistemas relacionales no manejaban eficientemente, coincidiendo con el auge de la programación orientada a objetos (POO) [1].

- **Descripción:** La información se almacena como objetos persistentes que encapsulan tanto datos como comportamiento (métodos). A diferencia del modelo relacional, que obliga a "descomponer" los objetos en tablas planas, este modelo busca eliminar la **incompatibilidad de impedancia** (*impedance mismatch*). Esto permite que objetos complejos de lenguajes como C++ o Java se guarden directamente en disco manteniendo su identidad, herencia y polimorfismo sin necesidad de mapeo (ORM) [2].
- **Ventajas:**
 - *Persistencia Directa:* Elimina el costo computacional de traducir objetos a tablas y viceversa.
 - *Manejo de Datos Complejos:* Superior para estructuras jerárquicas profundas o anidadas que requerirían múltiples *JOINS* en SQL.
 - *Identidad Única:* Cada objeto tiene un identificador inmutable (OID) independiente de sus valores [1].
- **Desventajas:**
 - *Falta de Estándares:* A pesar de los esfuerzos del ODMG (*Object Data Management Group*), la falta de un estándar universal (como lo es SQL) fragmentó el mercado.
 - *Complejidad y Mercado:* Su curva de aprendizaje es alta y su cuota de mercado se ha mantenido baja, siendo desplazada en casos generales por los ORMs modernos y las bases NoSQL [3].
- **Casos de Uso:** Sistemas CAD/CAM (diseño de piezas complejas), Sistemas de Información Geográfica (GIS), y aplicaciones científicas (como el genoma humano).

2. Modelos NoSQL

El movimiento NoSQL surgió con fuerza en la década de 2010 debido al auge del *Big Data* y la Web 2.0. Estos sistemas están diseñados para escalar horizontalmente, sacrificando a menudo las propiedades **ACID** (Atomicidad, Consistencia, Aislamiento, Durabilidad) del modelo relacional en favor del modelo **BASE** (Disponibilidad Básica, Estado Blando, Consistencia Eventual) [3, 4].

2.1. Modelos Documentales (Document Stores)

- **Descripción:** Almacenan datos en documentos semi-estructurados y autodescriptivos (JSON, BSON, XML). No requieren un esquema rígido (*schema-less*).
- **Tecnología Representativa:** MongoDB.
- **Ventajas:** Flexibilidad extrema para evolución del software (añadir campos sin tirar la base de datos) y mapeo natural a objetos JSON en aplicaciones web [5].
- **Desventajas:** No soportan transacciones complejas (ACID) que involucren múltiples documentos y las uniones (*joins*) son costosas o inexistentes.

2.2. Modelos de Grafos (Graph Databases)

- **Descripción:** Optimizados para gestionar relaciones interconectadas. Utilizan **nodos** y **aristas** que almacenan punteros físicos directos (*index-free adjacency*).
- **Tecnología Representativa:** Neo4j (Lenguaje Cypher).
- **Ventajas:** Rendimiento constante ($O(1)$) al navegar relaciones profundas, ideal para descubrir patrones ocultos.
- **Desventajas:** Dificil escalabilidad horizontal (*sharding*) debido a la interconexión de los datos; curva de aprendizaje de lenguajes específicos distintos a SQL [6].
- **Casos de Uso:** Redes sociales, detección de fraudes bancarios y motores de recomendación.

2.3. Modelos Columnares (Wide-Column Stores)

- **Descripción:** Almacenan datos agrupados por familias de columnas. Permite leer solo los atributos necesarios de millones de registros sin cargar la fila completa [3].
- **Tecnología Representativa:** Apache Cassandra.
- **Ventajas:**
 - *Compresión:* Al guardar datos homogéneos juntos, se logra una compresión de disco eficiente.
 - *Escalabilidad:* Diseñadas para escritura masiva y distribución en cientos de servidores [5].
- **Desventajas:** Diseño de esquema rígido y orientado a la consulta (debes saber qué preguntarás antes de crear la tabla); las actualizaciones de filas completas son lentas.

2.4. Modelos Clave-Valor (Key-Value)

- **Descripción:** Funciona como una tabla hash distribuida donde un valor opaco (BLOB) se accede mediante una clave única [4].
- **Tecnología Representativa:** Redis, Amazon DynamoDB.
- **Ventajas:** Velocidad extrema de acceso ($O(1)$) y simplicidad.
- **Desventajas:** No permite consultas basadas en el contenido del valor (caja negra); no existen relaciones nativas entre datos.
- **Casos de Uso:** Caché de sesiones, carritos de compra y contadores en tiempo real.

Referencias

- [1] R. Elmasri and S. B. Navathe, *Fundamentos de Sistemas de Bases de Datos*. Madrid: Pearson Educación S.A., 5a ed., 2007.
- [2] C. J. Date, *An Introduction to Database Systems*. Boston: Pearson Addison Wesley, 8a ed., 2004.
- [3] Varios Autores, “Comprehensive analytical compendium of database systems: Theoretical foundations, design methodologies, and implementation architectures,” tech. rep., s.f. Reporte técnico basado en obras de C.J. Date, Elmasri, Navathe, y otros.
- [4] Ángel Pisco Gómez *et al.*, *Fundamentos sobre la Gestión de Base de Datos*. Alcoy: 3Ciencias, 2017.

- [5] Universitat de València, “Course guide 34695 next generation information systems,” 2024. Accedido: 2025-02-05.
- [6] Universitat Politècnica de València, “Course description: Data management technology,” 2024. Accedido: 2025-02-05.