



# **UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

## **FACULTAD DE INGENIERÍA**

Ingeniería en Computación

### **TAREA 1.**

Velarde Valencia Josue

**Grupo: 01**

**Asignatura: Bases de Datos**

**Fecha de entrega: 15 de Agosto de 2025**

# **TAREA 1**

## **1. Modelo Orientado a Objetos**

### **Descripción:**

La programación orientada a objetos es un enfoque que utiliza conjuntos de objetos en lugar de métodos lógicos. Almacena datos y operaciones relacionadas en forma de tipos de datos abstractos. Estas clases de datos se utilizan varias veces para generar nuevos objetos con las mismas funciones. Esta programación es útil para aplicaciones que permiten la interacción entre humanos y máquinas.

### **Ventajas:**

- La programación orientada a objetos (OOP) ayuda a la reutilización y el reciclaje del código
- Elimina la necesidad de duplicación de código y promueve la colaboración y el intercambio de conocimientos.
- Puede utilizar el código principal de la clase maestra y solo cambiar las características requeridas para su código secundario
- Mejora la eficiencia y la productividad en el desarrollo de software.
- Puedes comenzar un código usando un código base y trabajar modificándolo y mejorándolo en lugar de escribirlo desde cero.
- Además, te da pequeños fragmentos de problemas para resolver y reduce la complejidad de la tarea.
- Facilita la resolución rápida de problemas al señalar dónde se encuentra el error. Así, no es necesario revisar el código de principio a fin.
- Filtra los datos y limita el acceso, mejorando así la seguridad del código y del software.

### **Desventajas:**

- La programación orientada a objetos aumenta la longitud de los programas, que llegan a ser más largos que los creados utilizando un enfoque procedimental.
- El tiempo de ejecución del programa toma mucho tiempo, lo que retrasa otras tareas.
- No se puede aplicar a todas las aplicaciones. Requiere un proceso de pensamiento distinto al de la programación tradicional.
- Se requiere un equilibrio entre habilidades de programación y diseño para que la POO funcione perfectamente

### **Casos de uso:**

- La programación orientada a objetos mejora la flexibilidad al facilitar modificaciones, adiciones y actualizaciones sin alterar el resto del código base.
- Ayuda a crear conjuntos manejables al dividir sistemas con conjuntos de datos complejos y grandes en clases de objetos. Esto permite al equipo trabajar en un desarrollo más estructurado.
- Los objetos listos se pueden emplear en múltiples códigos y reutilizar, ahorrando así tiempo al crear fragmentos de datos.

## 2. Modelo Clave-Valor

### **Descripción:**

Una base de datos clave valor (Key-Value) es un tipo de base de datos NoSQL que funciona con un modelo simple de claves y valores. Esto se refiere al hecho de que la base de datos almacena en pares clave/valor. La clave puede ser sintética o autogenerada, así como también puede poseer diversos formatos. La condición obligatoria que ha de tener dicha clave, es que debe ser única. Los valores por otro lado, se construyen en una estructura simple, que acepta diferentes tipos de formatos.

### **Ventajas:**

Las bases de datos clave valor almacenan los datos en diccionarios. Esto nos garantiza tener la información, sin importar su peso, en una clasificación funcional que podemos recuperar y usar en cualquier momento. La velocidad y escalabilidad que ofrecen son de los puntos más fuertes de estas bases.

### **Desventajas:**

Estas bases de datos son muy importantes y funcionales, pero tienen sus puntos flojos. Comparte con el resto de las NoSQL la debilidad de no tener estándares en el manejo de datos. En ocasiones los lenguajes de consultas son básicos, impidiendo realizar trabajos de profundidad. Algunos gestores carecen de herramientas avanzadas para su uso y configuración personalizada. Adicionalmente las consultas y el ordenamiento de la información pueden ser realizados solo partiendo de la clave primaria.

### **Casos de uso:**

El uso principal de esta base de datos se centra en las operaciones básicas get (obtener datos vinculados a una clave), put (vincular un valor determinado a una clave) y delete (borrar una entrada con una clave).

Las características simples pero ventajosas de las bases de datos clave valor permiten vincularla con diversos tipos de proyectos. Podemos disfrutar de máxima funcionalidad con un diseño sencillo para proyectos en los que la capacidad de almacenar y recuperar datos sea importante.

Podemos considerarla en casos en los que los datos no sean estructurados o cuando no es importante establecer relaciones entre los elementos a vincular. Esta base de datos tiene una gran potencialidad para ser implementada en el desarrollo rápido de prototipos.

## 3. Modelo Documental

### **Descripción:**

Tienen el mismo formato de modelo de documento que los desarrolladores utilizan en el código de sus aplicaciones. Almacenan los datos como objetos JSON que son flexibles, semiestructurados y de naturaleza jerárquica. La naturaleza flexible, semiestructurada y

jerárquica de los documentos y las bases de datos de documentos permite que evolucionen según las necesidades de las aplicaciones.

**Ventajas:**

- Permite almacenar datos con estructuras variables y semiestructuradas, adaptándose a cambios sin las complejas migraciones de tablas relacionales.
- Las bases de datos documentales pueden escalar añadiendo más servidores (nodos) a un clúster, lo que es ideal para manejar grandes volúmenes de datos de manera eficiente.
- Están optimizadas para patrones de datos específicos, ofreciendo gran velocidad en la lectura y escritura de grandes cantidades de datos.
- Los desarrolladores pueden interactuar directamente con los datos en formatos como JSON, simplificando la gestión de datos y reduciendo la dependencia de ORMs (Object-Relational Mappers).
- Son especialmente efectivas para manejar datos procedentes de redes sociales, logs, y datos de IoT, donde la estructura puede variar considerablemente.

**Desventajas:**

- A menudo, sacrifican la consistencia fuerte de los datos (garantías ACID) por mayor disponibilidad y escalabilidad.
- No hay un lenguaje de consulta único y estandarizado como SQL. Cada base de datos documental puede tener su propio lenguaje, lo que dificulta la portabilidad y la estandarización.
- Realizar consultas complejas y uniones entre diferentes conjuntos de datos puede ser más lento y complicado que en bases de datos relacionales.
- A pesar de su crecimiento, algunas tecnologías NoSQL tienen menos herramientas y un soporte comunitario más limitado que las bases de datos relacionales más establecidas.
- Los índices utilizados en las bases de datos documentales pueden consumir una cantidad significativa de memoria RAM, especialmente con grandes volúmenes de datos.

**Casos de uso:**

El modelo de base de datos de documentos funciona bien con catálogos, perfiles de usuario y sistemas de administración de contenido en los que cada documento es único y evoluciona con el tiempo.

## 4. Modelo de Grafos

**Descripción:**

Las bases de datos orientadas a grafos son excelentes herramientas para trabajar datos complejos. Estas BBDD nos permite darle representar los datos en estructuras de grafos. Esto es de gran utilidad cuando los datos que vamos a procesar tienen altos niveles de interrelación.

**Ventajas:**

- Las bases de datos en grafos cuentan con gran cantidad de aspectos positivos. Es importante comprenderlos y analizarlos a profundidad para descubrir sus potencialidades, veamos:
- La experiencia reciente lo ha demostrado. El peso de los datos crece de forma acelerada y exponencial. En la misma medida o velocidad crecerán las relaciones en vuestros datos. Ante tal dinámica las Bases en grafos nos ofrece una amplia capacidad de responder a las exigencias estructurales de estos volúmenes de datos.
- Las conexiones complejas de las BBDD en grafos nos proveen de diversos métodos analíticos. Esto nos permite solucionar los diversos retos que plantea el mundo digital hoy en día.
- La velocidad de procesamiento que nos ofrece este tipo de Base de Datos puede significar una ventaja. En un entorno de alta exigencia, donde las decisiones deben ser tomadas a toda velocidad, esto es fundamental.

**Desventajas:**

Existen pocas desventajas en estas bases de datos, pero pueden ser significativas. Como la mayoría de las BBDD de la familia NoSQL tienen dificultades con respecto a la atomicidad y sus patrones de estandarización. En algunos casos cuando se realizan particiones en función de propiedades locales pueden presentarse dificultades similares a las de una base de datos relacional.

**Casos de uso:**

Estas versátiles bases de datos, nos permite ejecutar consultas y almacenamiento de datos de cualquier característica sin siquiera preocuparnos por el volumen de los datos.

## 5. Modelo Columnar

**Descripción:**

Una base de datos columnar almacena los datos en columnas en lugar de en filas. Estas bases de datos modernas también se denominan a veces almacenes «orientados a columnas» o «de columna ancha». A medida que las empresas aumentan la cantidad de datos almacenados, pueden alcanzar terabytes (o más) de almacenamiento de datos que deben recuperarse. Las bases de datos columnas aceleran el procesamiento de las consultas y a menudo se utilizan para macrodatos o consultas para analíticas de aprendizaje automático.

**Ventajas:**

- Son ideales para tareas que implican la lectura y evaluación de grandes volúmenes de datos, como el procesamiento analítico en línea (OLAP).
- Los datos dentro de una misma columna son del mismo tipo, lo que permite una compresión de datos más eficiente y, por lo tanto, un acceso más rápido a los mismos.
- Permiten una mayor flexibilidad en comparación con las bases de datos relacionales, ya que no es necesario definir un esquema fijo antes de almacenar los datos.

- El diseño de la base de datos almacena los datos por columnas, lo que hace que las consultas que solo necesitan una o pocas columnas sean muy rápidas.

#### **Desventajas:**

- No son adecuadas para el procesamiento transaccional en línea (OLTP), ya que no son eficientes para operaciones que requieren escribir o actualizar datos dispersos en múltiples columnas o filas.
- Las actualizaciones de datos, especialmente si afectan a múltiples atributos de una entidad, pueden requerir más bloques y ser más lentas que en un modelo relacional.
- No existe un lenguaje de consulta unificado como SQL, lo que puede dificultar la migración entre diferentes bases de datos NoSQL.
- En comparación con las bases de datos SQL, el ecosistema de NoSQL es menos maduro, con menos herramientas, menos experiencia disponible para los desarrolladores y un soporte comunitario que puede ser limitado para proyectos a gran escala.

#### **Casos de uso:**

- Analíticas empresariales: Para muchas métricas empresariales, necesita unas pocas columnas para resumir el éxito. Una base de datos columnar puede mostrar mejor las predicciones de análisis y aprendizaje automático basadas en estas pocas columnas. Por ejemplo, los análisis basados en las ventas totales de un producto pueden ser adecuados para el almacenamiento de bases de datos columnares.
- Supervisión de la seguridad o las aplicaciones: Los datos recopilados a partir de los eventos de la aplicación (por ejemplo, errores de autenticación o tiempos de respuesta) pueden almacenarse en una base de datos columnar y usarse en analíticas para mejorar el rendimiento y detener cualquier ciberataque en curso.
- IoT: los sensores de IoT para la maquinaria de almacén o la supervisión sanitaria recogen datos y los almacenan en columnas específicas, que luego pueden usarse para detectar anomalías en la maquinaria o la bioactividad humana.

## **Referencias**

[1] K. Kelley, "What is Object-Oriented Programming? Definition, Pros, Cons, and Applications," \*CaltechBootcampBlog\*, updated Sep. 24, 2024. <https://pg-p.ctme.caltech.edu/blog/coding/what-is-object-oriented-programming>.

[2] GraphEverywhere, "Bases de Datos NoSQL | Qué son, marcas, tipos y ventajas," \*GraphEverywhereBlog\*, <https://www.grapheverywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/>.

[3] GraphEverywhere, "Bases de datos NoSQL | Bases de datos de Grafos," \*GraphEverywhere Blog\*, <https://www.grapheverywhere.com/nosql-de-grafos/>.

[4] Pure Storage, "¿Qué es una base de datos Columnar?", \*Pure Storage Knowledge\*, Abr. 2024. <https://www.purestorage.com/es/knowledge/what-is-a-columnar-database.html>.