



**UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA**



**Bases de datos**

**Profesor: Arreola Franco Fernando**

**Grupo: 1**

**Tarea 1: Modelos de datos**

**Alumno: Romero Rivera Geovanni**

**Fecha de entrega: 08 de septiembre de 2021**

**Semestre: 2022-1**

## Modelos de datos

### Modelo orientado a objetos:

Es un modelo de administración de datos. Las bases de datos orientadas a objetos en lugar de incorporar tablas como lo hacen las relacionales, utilizan objetos. Los objetos de una base de datos tienen las mismas características conocidas de los objetos de los lenguajes orientados a objetos (herencia, polimorfismo, encapsulación, otros). Este modelo admite relaciones uno a varios, uno a uno, y varios a varios.



En el método tradicional se realizan diagramas de entradas y salidas, tablas y estructuras de datos y diagramas de acción. En el modelo orientado a objetos el manejo de los objetos está determinado por las implementaciones de los métodos detectados en el análisis. Esto quiere decir que, en el método tradicional, el análisis de diseño suele tener más fronteras bien definidas, mientras que en el modelo de objetos la frontera no está tan bien definida.

Al intentar trabajar con sistemas de información geográfica o sistemas multimedia, los modelos (jerárquicos, de red o relacional) no se sienten demasiado cómodos e incluso algunos ni siquiera pueden soportarlos por, entre otros, los motivos que se enumeran a continuación:

- La estructura de los objetos es más compleja
- Las transacciones son de larga duración
- Se necesitan nuevos tipos de datos para almacenar imágenes y textos
- Hace falta definir operaciones no estándar, específicas para cada aplicación.

## Modelo Objeto/relacional:

El modelo de base de datos objeto-relacional integra los conceptos de la tradicional base de datos relacional y los conceptos de paradigma de objetos que se utiliza en la programación orientada a objetos (POO).

El objetivo de este concepto es poder aplicar la tecnología madura de bases de datos relacionales sobre la organización de los datos complejos es decir datos de texto e imagen, mapas, datos en el rango de audio etc. Las bases de datos Objeto-relacional son compatibles con estos objetos de datos y las operaciones de mayor complejidad.

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- **Encapsulación** - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- **Herencia** - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- **Polimorfismo** - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En una base de datos objeto relacional los dominios de dicha base de datos ya no son sólo atómicos por esta razón no cumplen la 1FN debido a que las tuplas también pueden ser una relación, que llevará a la creación de una relación de relaciones es así como no se puede aplicar el concepto de normalización. Esto porque ni siquiera se puede aplicar la primera forma normal y como consecuencia ni la segunda, ni tercera etc.

De este modo, se genera la posibilidad de guardar objetos más complejos en una sola tabla con referencias a otras relaciones, con lo que se acerca más al paradigma de programación orientada a objetos.

Veamos más de cerca las características de estos datos complejos:

**Colecciones:** También conocidos como conjuntos, este tipo de datos clasifican los arrays y los conjuntos en que los elementos pueden aparecer varias veces.

**Tipos estructurados:** Los tipos estructurados permiten representación directa de los atributos compuestos en los diagramas entidad-relación.

**Objetos de gran tamaño:** Desde ya hace varios años que se necesita almacenar datos con atributos muy grandes (Varios Mbytes), como libros, canciones, etc. E incluso aún más grandes; como mapas de alta resolución, video, etc. que puede llegar fácilmente a los Gbytes.

Los objetos en la base de datos constan de dos partes:

1. **Atributos:** Datos, metadatos e información de localización.
2. **Métodos:** Son funciones cuyos parámetros son los atributos del objeto.

### **Modelo NoSQL:**

Es una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico de SGBDR (Sistema de Gestión de Bases de Datos Relacionales) en aspectos importantes, siendo el más destacado que no usan SQL como lenguaje principal de consultas. Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN, ni garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad) y habitualmente escalan bien horizontalmente. Los sistemas NoSQL se denominan a veces "no solo SQL" para subrayar el hecho de que también pueden soportar lenguajes de consulta de tipo SQL.

Por lo general, los investigadores académicos se refieren a este tipo de bases de datos como almacenamiento estructurado, término que abarca también las bases de datos relacionales clásicas. A menudo, las bases de datos NoSQL se clasifican según su forma de almacenar los datos, y comprenden categorías como clave-valor, las implementaciones de BigTable, bases de datos documentales, y bases de datos orientadas a grafos.

Los sistemas de bases de datos NoSQL crecieron con las principales redes sociales, como Google, Amazon, Twitter y Facebook. Estas tenían que enfrentarse a desafíos con el tratamiento de datos que las tradicionales SGBDR no solucionaban {ver artículo}. Con el crecimiento de la web en tiempo real existía una necesidad de proporcionar información procesada a partir de grandes volúmenes de datos que tenían unas estructuras horizontales más o menos similares. Estas compañías se dieron cuenta de que el rendimiento y sus propiedades de tiempo real eran más importantes que la coherencia, en la que las bases de datos relacionales tradicionales dedicaban una gran cantidad de tiempo de proceso.

En ese sentido, a menudo, las bases de datos NoSQL están altamente optimizadas para las operaciones recuperar y agregar, y normalmente no ofrecen mucho más que la funcionalidad de almacenar los registros (p.ej. almacenamiento clave-valor). La pérdida de flexibilidad en tiempo de ejecución, comparado con los sistemas SQL clásicos, se ve compensada por ganancias significativas en escalabilidad y rendimiento cuando se trata con ciertos modelos de datos.