

## TAREA 2

INVESTIGAR LOS SIGUIENTES MODELOS EN BASES DE DATOS: ORIENTADO A OBJETOS, OBJETO/RELACIONAL, NoSQL.

Los administradores de base de datos evolucionan con el afán de satisfacer nuevos requerimientos tecnológicos y de información. Aunque los DBMS relacionales (RDBMS) son actualmente líderes del mercado y brindan las soluciones necesarias a las empresas comerciales, existen aplicaciones que necesitan funciones con las que no cuentan.

En esta ocasión me enfocare en tres tipos de modelos los cuales se consideran los mas importantes, empezare explicando el modelo orientado a objetos.

MODELO ORIENTADO A OBJETOS.



En el caso de los sistemas de gestión de base de datos orientada a objetos (OODBMS por sus siglas en inglés), la persistencia implica almacenar los valores de atributos de un objeto con la transparencia necesaria para que el desarrollador de aplicaciones no tenga que implementar ningún mecanismo distinto al mismo lenguaje de programación orientado a objetos.

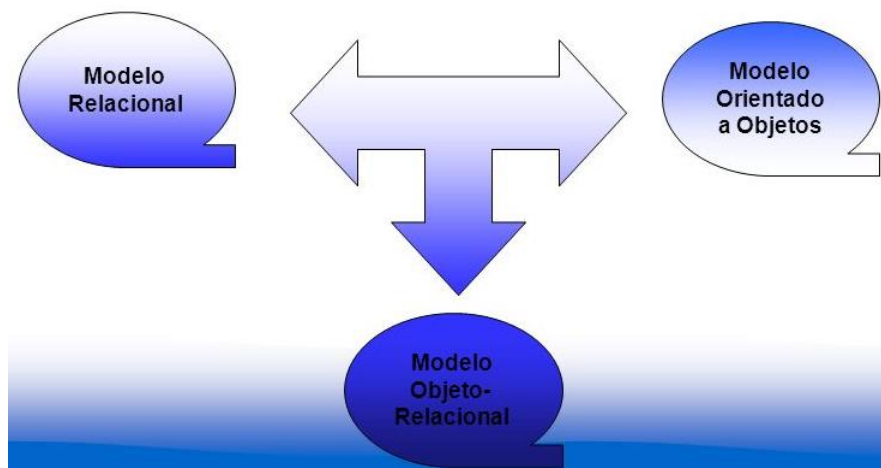
Lo anterior traería como ventaja que no sería necesario el uso de dos lenguajes de programación para construir una aplicación; es decir, actualmente, el desarrollo de aplicaciones se hace con lenguajes de programación orientada a objetos almacenando datos en bases relacionales, por lo que el desarrollador debe utilizar un lenguaje para la aplicación (Java, PHP, C++) y otro para la base de datos (SQL).

## 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.

Nacen como una extensión del modelo relacional, en el que los dominios de dicha base de datos ya no son sólo atómicos, por lo que no se cumple la 1FN, debido a que las tuplas también pueden ser una relación, que llevará a la creación de una relación de relaciones. 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 POO.



## MODELO noSQL

Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala.

Las bases de datos NoSQL se adaptan perfectamente a muchas aplicaciones modernas, como dispositivos móviles, web y juegos, que requieren bases de datos flexibles, escalables, de alto rendimiento y altamente funcionales para proporcionar excelentes experiencias de usuario. Las principales ventajas son:

- **Flexibilidad:** las bases de datos NoSQL generalmente ofrecen esquemas flexibles que permiten un desarrollo más rápido e iterativo. El modelo de datos flexible hace que las bases de datos NoSQL sean ideales para datos semiestructurados y no estructurados.
- **Escalabilidad:** las bases de datos NoSQL generalmente están diseñadas para escalar usando clústeres distribuidos de hardware en lugar de escalar añadiendo servidores caros y sólidos. Algunos proveedores de la nube manejan estas operaciones en segundo plano, como un servicio completamente administrado.
- **Alto rendimiento:** la base de datos NoSQL está optimizada para modelos de datos específicos y patrones de acceso que permiten un mayor rendimiento que el intento de lograr una funcionalidad similar con bases de datos relacionales.
- **Altamente funcional:** las bases de datos NoSQL proporcionan API altamente funcionales y tipos de datos que están diseñados específicamente para cada uno de sus respectivos modelos de datos.

