

**Modelo Orientado a Objetos:** Con el modelo de datos orientado a objetos, una entidad y sus relaciones están contenidas en una única estructura o plantilla conocida como clase. Dicha clase contiene métodos y atributos que son manejados por una instancia abstracta llamada 'objeto' que puede asociarse con otras clases. El modelo orientado a objetos es una extensión del paradigma de programación orientado a objetos, ya que sigue los principios de encapsulamiento, herencia y polimorfismo.

En este modelo se debe trabajar la jerarquización de paquetes, carpetas, subcarpetas y manejo de archivos que contenga la aplicación final. La estructura de datos de estas jerarquías son de tipo árbol y todas son creadas e implementadas por el desarrollador de software.

Las fases del modelo orientado a objetos constan de:

1. Requerimientos o reglas del negocio.
2. Diagrama de Casos de Uso (en lugar del diagrama entidad-relación).
3. Diagrama UML (en lugar del diagrama lógico-relacional).
4. Desarrollo de la aplicación en un lenguaje de programación orientado a objetos.

Cuando se integran las características de una base de datos con las de un lenguaje de programación orientado a objetos, el resultado es un sistema gestor de base de datos orientada a objetos

Los sistemas de gestión de bases de datos orientados a objetos se desarrollan en gran medida como respuesta a las necesidades de ingeniería. Sus aplicaciones en este campo incluyen el diseño asistido por ordenador (CAD) y entornos de desarrollo de software integrados (IDEs) en un lenguaje de programación orientado a objetos como Java, C#, Visual Basic.NET y C++.

Un sistema gestor de base de datos orientada a objetos (ODBMS, *Object Database Management System*) ofrece un mejor control de concurrencia, optimiza la recuperación de datos y permite realizar consultas asociativas.

**Modelo de datos Objeto-Relacional:** El diseño de la base de datos objeto-relacional es una extensión del diseño de la base de datos relacional tradicional, a la cual se le proporcionan características de la programación orientada a objetos (POO).

El modelo de datos objeto-relacional se rige bajo estándares como el SQL99. Este modelo consta de relaciones anidadas en el que los dominios de dicha base de datos ya no son sólo atómicos, por lo que no se cumple la primera forma normal en las tablas, 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 el modelo se acerca más al paradigma de POO.

En el manejador de base de datos Objeto-Relacional se utilizan datos complejos como colecciones, datos estructurados e instancias de gran tamaño, además, se puede implementar la herencia de tipos y la herencia de tablas.

**Modelos NoSQL:** NoSQL (a veces llamado "no solo SQL") es una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico del SGBDR (Sistema de Gestión de Bases de Datos Relacionales), ya que no usan SQL como lenguaje principal de consultas.

Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN y no garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad). Habitualmente se diseñan y se manejan de forma renglonar (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. Los expertos se refieren a estos modelos como bases de datos con 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 las 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. 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.

### **Bibliografía**

1. Kim, Won. Introduction to Object-Oriented Databases. The MIT Press, 1990.
2. Introducción a los sistemas de Bases de Datos Relacionales. Séptima Edición. C. J. Date.
3. Sistemas de Bases de Datos; Conceptos Fundamentales. ELMASRI-NAVATHE.
4. Fundamentos de Bases de datos. Abraham Silberschatz. McGraw-Hill.
5. Tema 2. Bases de datos orientadas a objetos Diseño de Sistemas de Bases de Datos. Merche Marques. *Rescatado de:*  
<https://www3.uji.es/~mmarques/e16/teoria/cap2.pdf>
6. [essanvicente.com/colaboraciones/bdOO.pdf](http://essanvicente.com/colaboraciones/bdOO.pdf)
7. <https://publications.lib.chalmers.se/records/fulltext/123839.pdf>