

“ AÑO DE LA RECUPERACIÓN Y CONSOLIDACIÓN DE LA ECONOMÍA PERUANA”

UNIVERSIDAD PERUANA LOS ANDES

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL

SISTEMAS Y COMPUTACIÓN



Docente: Ing. Fernandez Bejarano Raul Enrique

Estudiantes: Guadalupe Carbajal Emanuel Gaddiel

Código: R01068D

Ciclo: V

Sección: B1

Huancayo - Peru - Junin
2025

Base de Datos Relacional (RDBMS)

Es un tipo de base de datos que organiza la información en **tablas** formadas por filas y columnas. Las tablas pueden relacionarse entre sí mediante **claves primarias y foráneas**, lo que permite mantener la **integridad y consistencia** de los datos.

Utiliza el **lenguaje SQL** para realizar consultas, actualizaciones o eliminaciones de información.

Ejemplo: MySQL, PostgreSQL, Oracle, SQL Server.

Base de Datos No Relacional (NoSQL)

Es un modelo de base de datos que **no usa tablas** como las bases relacionales tradicionales. Está diseñado para trabajar con **datos no estructurados o semiestructurados**, y permite **esquemas flexibles** que pueden adaptarse fácilmente a los cambios en la información.

Se orienta a mejorar el **rendimiento, la escalabilidad y la disponibilidad**, especialmente en aplicaciones web y de Big Data.

Ejemplo: MongoDB, Cassandra, Redis, Neo4j.

Base de Datos Multimodelo

Es un tipo de base de datos que **combina varios modelos de almacenamiento** dentro de un mismo sistema, como el relacional, documental, de grafos o de columnas.

Permite manejar diferentes tipos de datos de manera **integrada y eficiente**, evitando el uso de múltiples sistemas distintos. Es ideal para proyectos que necesitan **flexibilidad y variedad de estructuras** de información.

Ejemplo: ArangoDB, OrientDB, Cosmos DB.

Cuadro Comparativo

| Características | Base de Datos Relacional (RDBMS) | Base de Datos No Relacional (NoSQL) | Base de Datos Multimodelo |
|----------------------|--|--|---|
| Estructura de datos | Datos organizados en tablas con un esquema fijo. | Datos almacenados en documentos, grafos, columnas o pares clave-valor, sin esquema rígido. | Combina varios modelos de datos en un mismo sistema (tablas, documentos, grafos, etc.). |
| Lenguaje de consulta | Utiliza SQL, un lenguaje estandarizado. | Usa lenguajes específicos según el tipo (JSONPath, CQL, Gremlin, etc.). | Admite SQL y otros lenguajes dependiendo del modelo usado. |
| Escalabilidad | Principalmente vertical (aumentar | Principalmente horizontal (añadir | Puede escalar vertical y |

| | | | |
|----------------------------------|--|---|---|
| | recursos en un solo servidor). | más nodos o servidores). | horizontalmente según las necesidades. |
| Consistencia e integridad | Muy alta, cumple con las propiedades ACID. | Menor consistencia, se basa en el modelo BASE (más disponibilidad). | Puede equilibrar entre ACID y BASE. |
| Rendimiento | Alto en transacciones complejas y estructuradas. | Muy rápido en grandes volúmenes de datos distribuidos. | Rendimiento adaptable según el modelo y tipo de datos. |
| Flexibilidad del esquema | Rígido, los datos deben ajustarse a un formato definido. | Flexible, permite añadir o modificar campos sin afectar al sistema. | Muy flexible, admite diferentes estructuras de datos simultáneamente. |
| Ejemplos | MySQL, PostgreSQL, Oracle. | MongoDB, Cassandra, Neo4j, Redis. | ArangoDB, OrientDB, Cosmos DB. |
| Ventajas | Alta integridad, consistencia y soporte técnico. | Escalabilidad, flexibilidad y velocidad en lectura/escritura. | Permite trabajar con múltiples tipos de datos y modelos en un solo sistema. |
| Desventajas | Difícil de escalar horizontalmente y poco flexible. | Menor consistencia y falta de estándares universales. | Más compleja de administrar y configurar. |
| Casos de uso | Bancos, inventarios, sistemas contables y financieros. | Big Data, redes sociales, análisis de datos, IoT. | Aplicaciones que mezclan varios tipos de información (texto, imágenes, grafos, etc.). |