

Cuadro Comparativo de Bases de Datos Relacionales, No Relacionales y Multimodelo

Universidad Peruana Los Andes

Facultad de Ingeniería — Escuela Profesional de Ingeniería de Sistemas y Computación

- Asignatura: Base de Datos II
- Docente: Mg. Raúl Fernández Bejarano
- Estudiante: Luis Enrique Espejo Quispe
- Ciclo: V — Sección A1 — HYO-2025



El Ecosistema de Almacenamiento de Datos

En la ingeniería de sistemas y computación, la base de datos es el corazón de cualquier aplicación, asegurando la persistencia y la integridad de la información. La elección de la arquitectura correcta es crucial para el éxito, el rendimiento y la escalabilidad de un proyecto.



Bases de Datos (BBDD)

Sistemas que permiten almacenar, organizar, gestionar y recuperar grandes volúmenes de datos de manera eficiente y segura.



Enfoque Relacional (SQL)

Modelo clásico basado en la teoría de conjuntos y lógica de primer orden, con un énfasis en la integridad transaccional.



Enfoque No Relacional (NoSQL)

Surgió para manejar el volumen y la velocidad del Big Data, priorizando la escalabilidad y la flexibilidad de esquema.



Enfoque Multimodelo

Representa la evolución que integra múltiples modelos de datos bajo una única plataforma unificada.

1. Bases de Datos Relacionales (SQL)

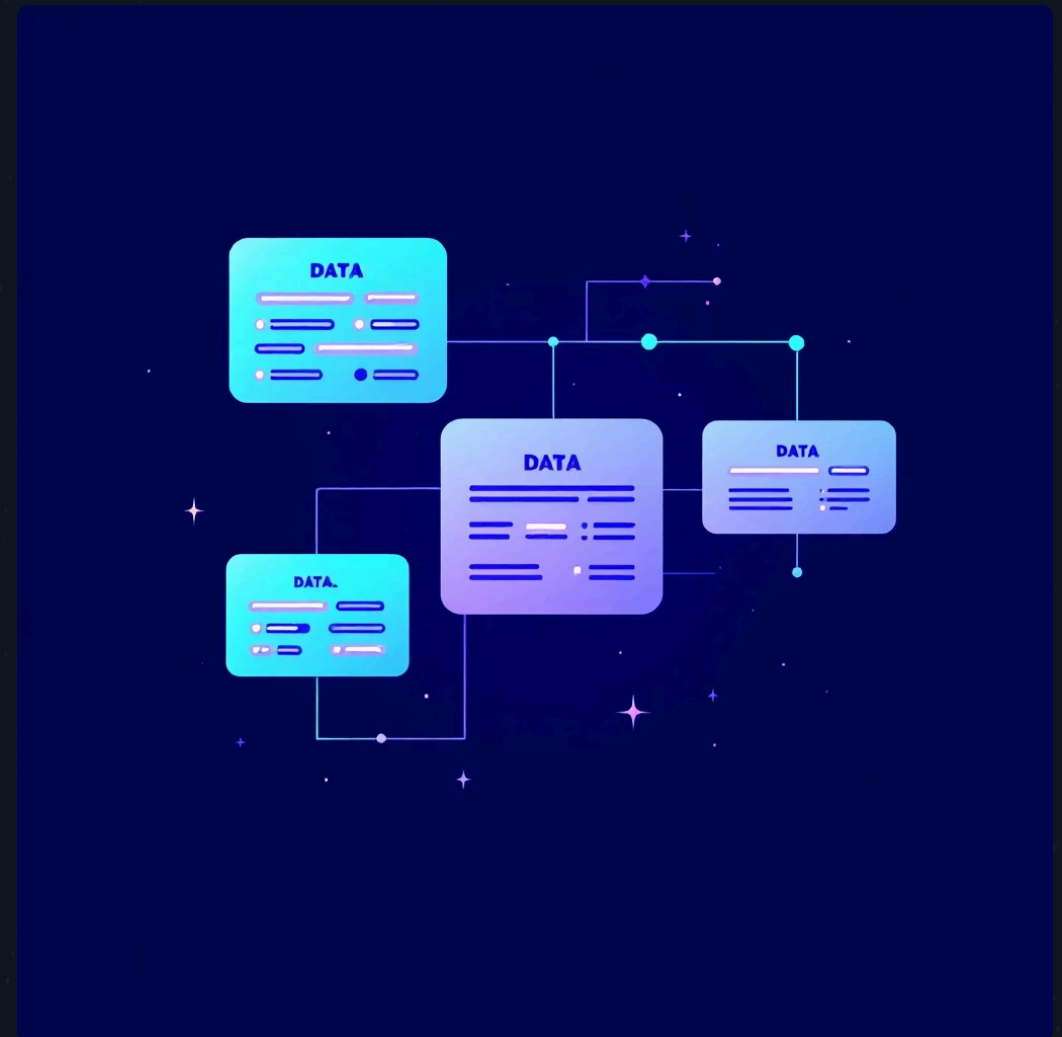
El modelo relacional, conceptualizado por E.F. Codd en 1970, organiza la información de forma estructurada y predefinida. Este modelo garantiza la máxima fiabilidad de los datos, siendo esencial para aplicaciones donde la coherencia es crítica.

Estructura y Consistencia

- Organización mediante tablas bidimensionales (entidades), compuestas por filas (registros) y columnas (atributos).
- Las relaciones se establecen rigurosamente a través de claves primarias (identificador único) y claves foráneas (referencia a otra tabla).
- Ejemplo clave: **PostgreSQL** (un potente sistema de gestión relacional de objetos).

Propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)

Son el pilar de las BBDD relacionales, asegurando que las transacciones se procesen de manera confiable, especialmente en escenarios de alta concurrencia.



Ventajas

Alta integridad, consistencia garantizada y transacciones seguras (ACID).

Limitaciones

Esquema rígido (poca flexibilidad) y escalabilidad principalmente vertical (aumentar capacidad del servidor).

2. Bases de Datos No Relacionales (NoSQL)

Las BBDD NoSQL surgieron para abordar las limitaciones de escalabilidad y flexibilidad de SQL, especialmente con el auge de Big Data y las aplicaciones web masivas. Se centran en la velocidad, la disponibilidad y la capacidad de manejar estructuras de datos variables.



Documento (e.g., MongoDB)

Almacenan datos semiestructurados en formatos similares a JSON/BSON. Ideales para catálogos o perfiles de usuario.



Clave-Valor (e.g., Redis)

Almacenamiento simple de datos asociados a una clave única. Extremadamente rápidas para caché y sesiones.



Grafo (e.g., Neo4j)

Enfocadas en las relaciones entre datos (nodos y aristas). Perfectas para análisis de redes y estructuras complejas.

A diferencia de ACID, NoSQL a menudo sigue el **Principio BASE** (Disponibilidad Básica, Estado Suave, Consistencia Eventual), priorizando la disponibilidad y la partición sobre la consistencia inmediata.



90%

Escalabilidad Horizontal

Permite distribuir la carga de trabajo añadiendo más servidores (sharding) fácilmente.



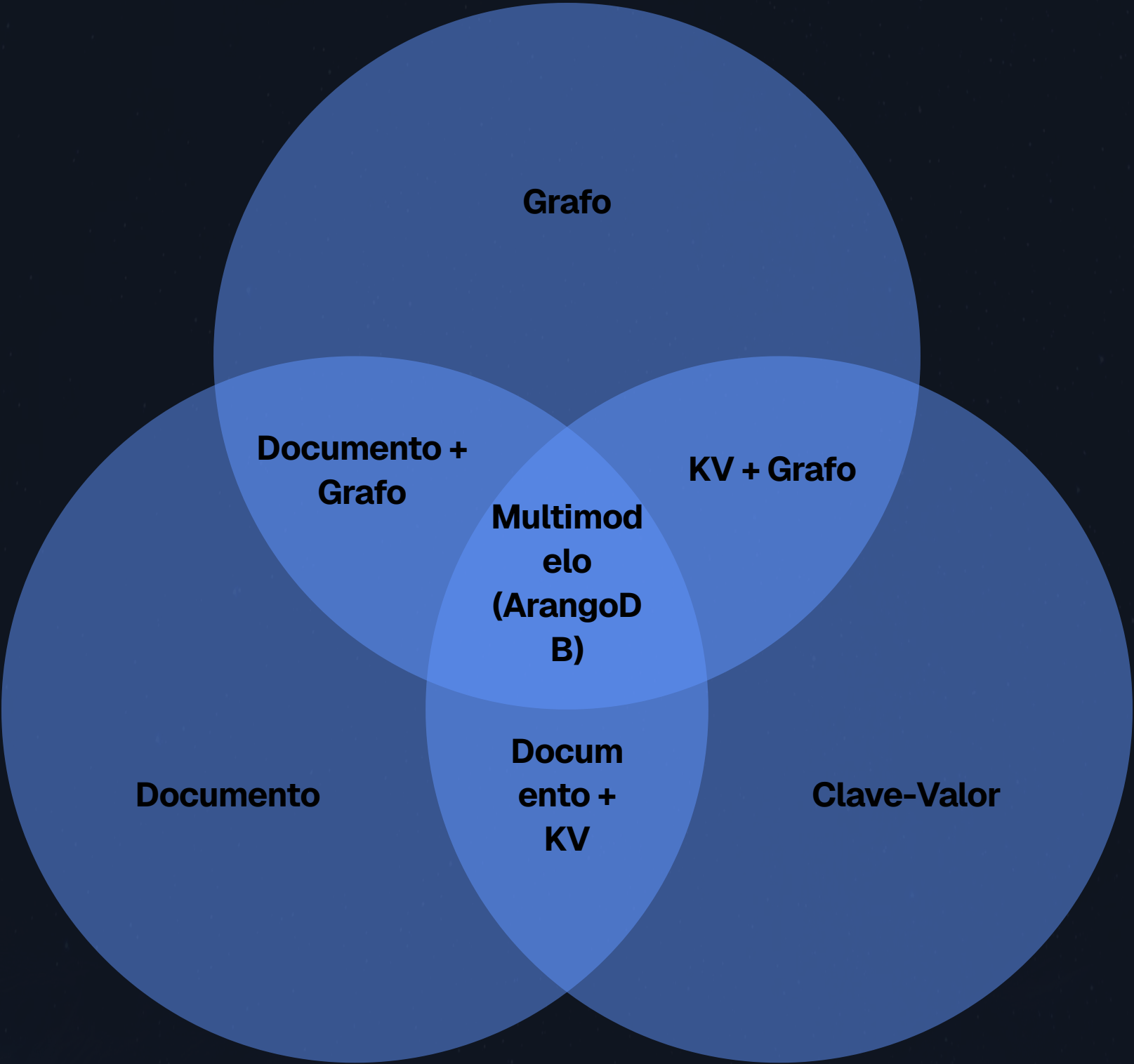
85%

Flexibilidad de Esquema

Permite modificar la estructura de los datos sin afectar la operación del sistema.

3. Bases de Datos Multimodelo: La Convergencia

Las bases de datos multimodelo representan una solución moderna que reconoce la necesidad de manejar diversos tipos de datos dentro de un único sistema. En lugar de forzar todos los datos a un solo formato (relacional, documento o grafo), estas BBDD ofrecen un soporte nativo para múltiples modelos.



Ventajas de la Unificación

- **Consultas Híbridas**

Capacidad de ejecutar consultas que combinan datos de diferentes modelos simultáneamente (ejemplo: consultas de documentos y grafos). ArangoDB utiliza el **AQL (ArangoDB Query Language)** unificado.
- **Simplicidad Operacional**

Se reduce la complejidad de la infraestructura al no necesitar múltiples sistemas especializados (una base de datos para cada tipo de dato).
- **Modelado Versátil**

Permite modelar problemas complejos de la vida real con mayor precisión.
Ejemplo: ArangoDB.

Escenarios de Aplicación

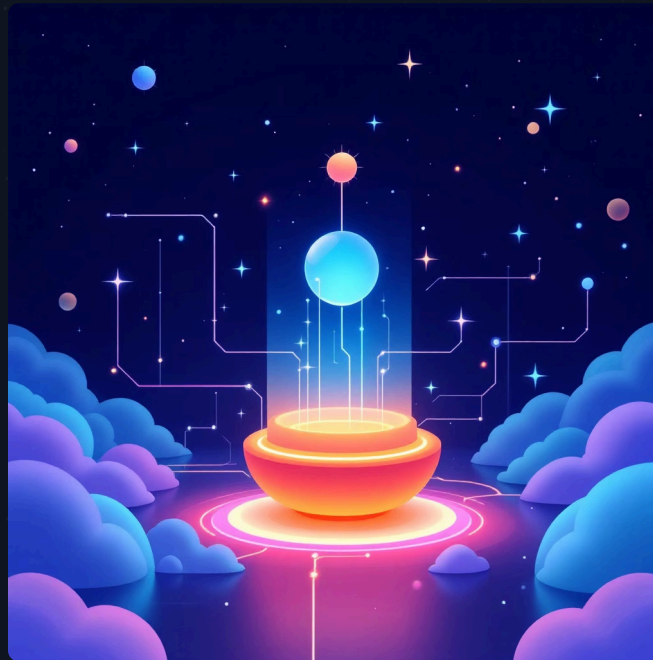
La elección del motor de base de datos debe estar impulsada por los requisitos funcionales del proyecto. Analicemos dónde sobresale cada arquitectura.

SQL (Relacional)



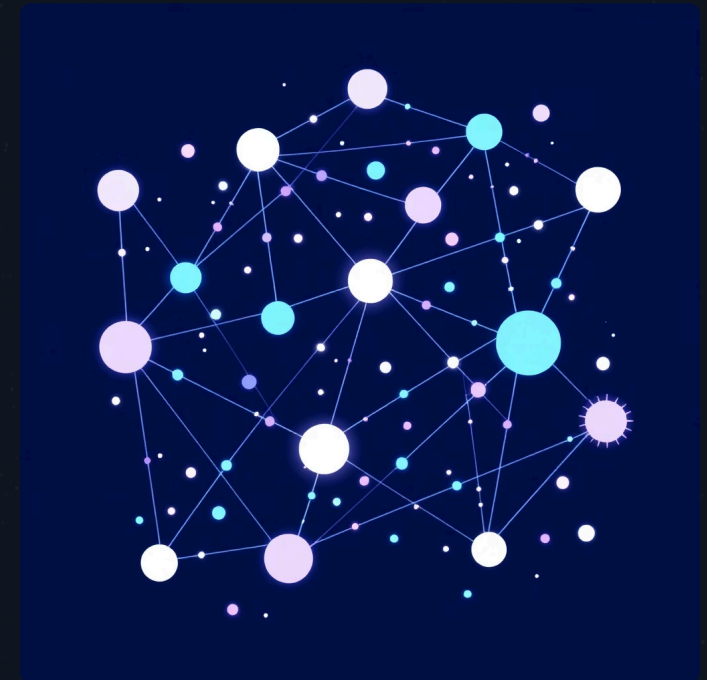
- Sistemas Bancarios y Financieros.
- Sistemas de Planificación de Recursos Empresariales (ERP).
- Plataformas de Comercio Electrónico con alta exigencia transaccional.
- Cualquier sistema que requiera consistencia estricta (ACID).

NoSQL (No Relacional)



- Internet de las Cosas (IoT) y flujos de datos rápidos.
- Plataformas de Contenido Dinámico (blogs, CMS).
- Catálogos de productos con atributos variables.
- Manejo de grandes volúmenes de datos con esquema cambiante.

Multimodelo



- Motores de Recomendación y Personalización (combinando usuarios y productos).
- Análisis de Fraude y Detección de Patrones.
- Redes Sociales y gestión de datos interconectados.
- Sistemas que necesitan flexibilidad y alta capacidad de consulta.

Cuadro Comparativo Detallado

Aspecto	Relacional (SQL)	No Relacional (NoSQL)	Multimodelo
Modelo de Datos	Tabular (Esquema Fijo)	Flexible (Documento, Clave-Valor, Grafo)	Combinación Nativa de Múltiples Modelos
Lenguaje	SQL (Structured Query Language)	API's de consulta, lenguajes específicos (ej. MQL)	Lenguaje Unificado (ej. AQL)
Consistencia	Fuerte (ACID)	Eventual (BASE)	Configurable (ACID/BASE según modelo)
Ventajas	Integridad de datos, Transacciones fiables	Escalabilidad horizontal, Flexibilidad de esquema	Versatilidad, Simplicidad de infraestructura
Limitaciones	Baja flexibilidad, Escalabilidad costosa	Relaciones complejas difíciles, Consistencia eventual	Menor madurez, Mayor complejidad de modelado
Usos Típicos	Banca, Contabilidad, ERP	Big Data, IoT, Caching, Contenido dinámico	Recomendaciones, Análisis de Grafos, Aplicaciones Híbridas

Este cuadro resalta cómo la evolución del manejo de datos busca superar la dicotomía tradicional, ofreciendo herramientas que se adaptan mejor a la realidad de los sistemas modernos.

Factores Clave para la Decisión Arquitectónica

La elección entre SQL, NoSQL y Multimodelo nunca es arbitraria. Depende de las necesidades específicas del proyecto. Los ingenieros de sistemas deben evaluar la arquitectura basándose en estos criterios:



Requerimiento de Consistencia

Si la integridad transaccional (ACID) es crítica (ej. dinero), se prioriza SQL. Si se acepta una consistencia eventual (ej. redes sociales), se opta por NoSQL.



Volumen y Velocidad de Datos

Para manejar Terabytes o Petabytes y flujos de datos rápidos (Big Data), la escalabilidad horizontal de NoSQL es indispensable.



Naturaleza del Esquema

Si la estructura de los datos es fija y conocida, SQL es ideal. Si el esquema evoluciona constantemente (datos variables), NoSQL o Multimodelo ofrecen mayor agilidad.



Complejidad de Relaciones

Si las relaciones son profundas y complejas (ej. redes de contactos), una base de datos de grafo (o Multimodelo) es la opción más eficiente.

Conclusiones y Tendencias Futuras

Hemos analizado los tres paradigmas principales de bases de datos. La conclusión fundamental es que **no existe una solución única**; cada tipo se adapta mejor a un conjunto particular de requisitos técnicos y de negocio.



Estrategia Contextual

La elección debe ser siempre una estrategia contextual. Los sistemas empresariales modernos a menudo emplean una arquitectura políglota de persistencia, usando varias bases de datos especializadas a la vez.



Ascenso Híbrido

La tendencia se dirige hacia la flexibilidad y la simplificación. Las bases de datos **Multimodelo** están ganando terreno al ofrecer una plataforma unificada que reduce la complejidad operacional y facilita la adopción de arquitecturas híbridas.



Rol del Ingeniero

El éxito de un proyecto de software recae en la capacidad del ingeniero de sistemas para discernir la arquitectura de base de datos más apropiada para optimizar el rendimiento, la escalabilidad y la confiabilidad.

Gracias por su Atención

La base de datos es el pilar de la información. Elegir bien define el futuro del sistema.

Facultad de Ingeniería – Escuela Profesional de Ingeniería de Sistemas y Computación

Docente: Mg. Raúl Fernández Bejarano

Estudiante: Luis Enrique Espejo Quispe | Ciclo V - 2025