



“Año De La Recuperación Y  
Consolidación De La Economía Peruana”



# **UNIVERSIDAD PERUANA LOS ANDES**

“FACULTAD DE INGENIERÍA”

ESCUELA PROFESIONAL “SISTEMAS Y  
COMPUTACIÓN”

## **Cuadro Comparativo**

**CÁTEDRA:** Base de Datos II

**CATEDRÁTICO:** Ing. Fernandez Bejarano Raul Enrique

**ESTUDIANTE:** Quispe Segama Franklin Noe

**CICLO:** V

**SECCIÓN:** B1

**HUANCAYO PERÚ**

**2025**

## **Cuadro Comparativo**

### **1. Bases de Datos Relacionales (SQL)**

La base de las bases de datos relacionales (RDBMS), ejemplificada por PostgreSQL, es el Modelo Relacional desarrollado por E.F. Codd en los años 70, que se basa en la teoría de conjuntos y la lógica de predicados. El principio fundamental es la organización de los datos en tablas bidimensionales, donde cada fila es una tupla (registro) y las columnas representan atributos con tipos de datos estrictos. La fortaleza de este modelo reside en la capacidad de definir relaciones claras entre tablas usando claves primarias y foráneas, lo que permite realizar JOINS complejos. La piedra angular de su fiabilidad son las propiedades ACID (Atomicidad, Consistencia, Islamiento, Durabilidad), que garantizan que las transacciones sean procesadas de manera segura y que la integridad de los datos se mantenga en todo momento, haciéndolas ideales para sistemas transaccionales críticos.

### **2. Bases de Datos No Relacionales (NoSQL)**

Las bases de datos NoSQL, como MongoDB, surgieron para abordar las limitaciones de escalabilidad y la rigidez de los esquemas relacionales, especialmente con el auge de Big Data y las aplicaciones web modernas. A diferencia de las SQL, no se limitan a un único modelo tabular, sino que utilizan estructuras diversas como documentos (JSON/BSON), clave-valor, o grafos. Su diseño favorece el escalamiento horizontal (distribuyendo la carga en múltiples servidores) y la flexibilidad de esquema, permitiendo a los desarrolladores iterar rápidamente sin necesidad de un plan de esquema rígido. Teóricamente, a menudo se adhieren al modelo BASE (Básicamente Disponible, Estado Suave, Consistencia Eventual), priorizando la disponibilidad y el rendimiento sobre la consistencia inmediata, una compensación vital para aplicaciones de alta concurrencia.

### 3. Bases de Datos Multimodelo (Híbridas)

El enfoque multimodelo, representado por bases de datos como ArangoDB, es una evolución que busca la eficiencia y la simplicidad operacional al unificar varios modelos de datos distintos (Documento, Grafo, Clave-Valor) dentro de un solo núcleo de motor de base de datos. El principio central es que una aplicación moderna a menudo necesita diferentes modelos de datos para diferentes tareas (ej. documentos para perfiles de usuario y grafos para relaciones entre ellos). El multimodelo permite gestionar todo esto con una única base de datos y un lenguaje de consulta unificado (como AQL), evitando la complejidad y la sobrecarga de latencia de tener que integrar y sincronizar múltiples bases de datos de un solo modelo (persistencia políglota). Esto resulta en un desarrollo más ágil y en la capacidad de ejecutar consultas que combinan las fortalezas de cada modelo.

#### • Cuadro Comparativo de Bases de Datos

Característica	PostgreSQL (Relacional/Objeto-Relacional)	MongoDB (NoSQL - Documentos)	ArangoDB (NoSQL - Multi-Modelo Nativo)
<b>Modelo de Datos</b>	Relacional (tablas con filas y columnas) y Objeto-Relacional. Soporta datos estructurados y semi-estructurados (JSON/JSONB).	Documentos flexibles (BSON, similar a JSON), organizados en colecciones. Esquema dinámico.	Multi-Modelo <b>nativo</b> : Documentos, Grafos (Graphs) y Clave/Valor.

<b>Lenguaje de Consulta</b>	<b>SQL</b> (Structured Query Language) y su variante <b>Postgres SQL</b> .	<b>MQL</b> (MongoDB Query Language). Soporta consultas ad-hoc, agregación y geoespaciales.	<b>AQL</b> (ArangoDB Query Language). Un único lenguaje declarativo para los tres modelos.
<b>Escalabilidad Principal</b>	Principalmente <b>Vertical</b> (aumentar recursos del servidor). También soporta replicación y fragmentación (sharding) con configuración adicional.	<b>Horizontal nativa</b> mediante <b>sharding</b> (particionamiento automático de datos en múltiples servidores).	<b>Horizontal</b> en los tres modelos de datos. Fácilmente escalable mediante clústeres.
<b>Transacciones / Consistencia</b>	Soporte <b>ACID completo</b> (Atomicidad, Consistencia, Aislamiento, Durabilidad) a nivel de fila. <b>Consistencia Fuerte</b> .	<b>Transacciones multi-documento</b> con soporte ACID (introducido en versiones recientes). Suele usar <b>Consistencia Eventual</b> para replicación.	Soporte <b>ACID</b> en instancia única y operaciones atómicas a nivel de documento en clúster. <b>Consistencia Fuerte</b> .

Característica	PostgreSQL (Relacional/Objeto-Relacional)	MongoDB (NoSQL - Documentos)	ArangoDB (NoSQL - Multi-Modelo Nativo)
<b>Ventajas Clave</b>	<b>Robustez, madurez, integridad</b> de datos, cumplimiento estricto de SQL, tipos de datos avanzados, extensibilidad (extensiones y lenguajes procedimentales).	<b>Flexibilidad</b> de esquema, desarrollo ágil, alto rendimiento en inserciones/actualizaciones masivas, <b>escalabilidad horizontal</b> sencilla.	<b>Consolidación</b> de múltiples modelos de datos, <b>rendimiento</b> al combinar modelos en una sola consulta, reduce la complejidad operativa (un solo sistema).

<b>Limitaciones</b>	<p>Curva de aprendizaje más alta que MySQL. Puede ser más lento que NoSQL en operaciones de lectura/escritura simples de alta frecuencia. Escalabilidad horizontal más compleja.</p>	<p>Manejo menos eficiente de relaciones complejas (sin <i>JOINS</i> relacionales nativos). Mayor consumo de memoria (debido al modelo de documento auto-contenido).</p>	<p>Curva de aprendizaje de <b>AQL</b> y la lógica multi-modelo. Comunidad y herramientas más pequeñas en comparación con PostgreSQL y MongoDB.</p>
<b>Escenarios de Uso Típicos</b>	<p>Sistemas ERP/CRM, aplicaciones que requieren <b>alta integridad</b> de datos (banca, finanzas), análisis de datos complejos (BI), sistemas GIS (datos geoespaciales).</p>	<p>Catálogos de productos (e-commerce), Sistemas de Gestión de Contenidos (<b>CMS</b>), aplicaciones de <b>IoT</b> (Internet de las cosas), Big Data, prototipado rápido.</p>	<p>Sistemas de <b>recomendación</b> (grafos), gestión de identidades y accesos, gestión de conocimiento (documentos y grafos), aplicaciones con requerimientos de datos muy diversos.</p>



