**UNIVERSIDAD PERUANA LOS ANDES FACULTAD DE INGENIERÍA** “ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN”



Base de datos II

**Alumno:** SORIANO TIMOTEO Joel Kevin

**Catedrático:** RAUL FERNANDEZ Bejarano

**Ciclo:** V

Huancayo - Perú - 2025

Cuadro comparativo:

|  |  |  |  |
| --- | --- | --- | --- |
| Característica | PostgreSQL (Relacional/Objet o-Relacional) | MongoDB (NoSQL - Documentos) | ArangoDB (NoSQL -  Multi-Modelo Nativo) |
| **Modelo de Datos** | 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 **nativo**: Documentos, Grafos (Graphs) y Clave/Valor. |
| **Lenguaje de Consulta** | **SQL** (Structured Query Language) y su variante **Postgres SQL**. | **MQL** (MongoDB Query Language). Soporta consultas ad-hoc, agregación y geoespaciales. | **AQL**  (ArangoDB Query Language). Un único lenguaje declarativo para los tres modelos. |
| **Escalabilida d Principal** | Principalmente **Vertical** (aumentar recursos del servidor).  También soporta replicación y fragmentación (sharding) con configuración adicional. | **Horizontal nativa** mediante **sharding** (particionamiento automático de datos en múltiples servidores). | **Horizontal** en los tres modelos de datos.  Fácilmente escalable mediante clústeres. |
| **Transaccion es / Consistencia** | Soporte **ACID completo** (Atomicidad, Consistencia, Aislamiento, Durabilidad) a nivel de fila.  **Consistencia Fuerte**. | **Transacciones multi- documento** con soporte ACID (introducido en versiones recientes).  Suele usar **Consistencia Eventual** para replicación. | Soporte **ACID** en instancia única y operaciones atómicas a nivel de documento en clúster.  **Consistencia Fuerte**. |
| Característica | PostgreSQL (Relacional/Objet o-Relacional) | MongoDB (NoSQL - Documentos) | ArangoDB (NoSQL -  Multi-Modelo Nativo) |
| **Ventajas Clave** | **Robustez, madurez, integridad** de datos, cumplimiento estricto de SQL, tipos de datos avanzados, extensibilidad (extensiones y lenguajes procedimentales). | **Flexibilidad** de esquema, desarrollo ágil, alto rendimiento en inserciones/actualizacion es masivas, **escalabilidad horizontal** sencilla. | **Consolidación** de múltiples modelos de datos, **rendimiento** al combinar modelos en una sola consulta, reduce la complejidad operativa (un solo sistema). |
| **Limitaciones** | 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. | Manejo menos eficiente de relaciones complejas (sin *JOINs* relacionales nativos). Mayor consumo de memoria (debido al modelo de documento auto- contenido). | Curva de aprendizaje de **AQL** y la lógica multi- modelo.  Comunidad y herramientas más pequeñas en comparación con PostgreSQL y MongoDB. |
| **Escenarios de Uso Típicos** | Sistemas ERP/CRM,  aplicaciones que requieren **alta integridad** de datos (banca, finanzas), análisis de datos complejos (BI), sistemas GIS (datos geoespaciales). | Catálogos de productos (e-commerce), Sistemas de Gestión de Contenidos (**CMS**), aplicaciones de **IoT** (Internet de las cosas), Big Data, prototipado rápido. | Sistemas de **recomendació n** (grafos), gestión de identidades y accesos, gestión de conocimiento (documentos y grafos), aplicaciones con requerimientos de datos muy diversos. |