

Taller 3 bases de datos

Tatiana Casallas Marin

Actividad 1

| | Cassandra | MongoDB | Firestore | Neo4j | Elasticsearch | BigQuery |
|-----------------|--|---|---|--|--|--|
| Características | Base de datos distribuida, descentralizada, tolerante a fallos | Base de datos orientada a documentos, esquema flexible | Base de datos en la nube, tiempo real, sin servidor | Base de datos de grafos nativa, ACID | Motor de búsqueda y análisis distribuido | Data warehouse totalmente gestionado, análisis a gran escala |
| Tipos de datos | UUID, text, int, bigint, float, double, boolean, timestamp, list, set, map | BSON (JSON binario), string, number, boolean, map, array, timestamp, buffer, boolean, array, object | string, number, boolean, map, array, timestamp, null | nodos, relaciones, propiedades, paths | text, keyword, number, date, boolean, geo_point | STRING, INT64, FLOAT64, BOOL, BYTES, DATE, TIMESTAMP |
| Ventajas | Alta disponibilidad, escalabilidad lineal, sin punto único de fallo | Escalabilidad horizontal, consultas ricas, desarrollo ágil | Sincronización en tiempo real, escalado automático, seguridad integrada | Consultas complejas de relaciones, rendimiento en datos conectados | Búsqueda full-text, análisis en tiempo real, escalabilidad | Procesamiento paralelo masivo, sin infraestructura, SQL estándar |
| Desventajas | Consistencia eventual, aprendizaje complejo | Uso elevado de memoria, transacciones limitadas | Costo por uso, dependencia del proveedor | Escalabilidad horizontal limitada, uso específico | No transaccional, complejidad operacional | Latencia en consultas, costo por escaneo de datos |
| Aplicaciones | Sistemas de mensajería, IoT, | CMS, e-commerce, aplicación | Aplicaciones móviles, juegos en | Redes sociales, recomendaciones, | Búsqueda empresarial, análisis de logs, | Business Intelligence, análisis de |

| | | | | | | |
|--------|--------------------------------------|----------------------|--------------------------------------|-----------------------------|-------------------------------|-------------------------------------|
| | análisis de datos en tiempo real | es móviles, big data | tiempo real, colaboración | detección de fraudes | seguridad | big data, machine learning |
| Fuente | Apache Cassandra Documentation, 2023 | MongoDB Manual, 2024 | Google Firestore Documentation, 2024 | Neo4j Developer Guide, 2024 | Elasticsearch Reference, 2024 | Google BigQuery Documentation, 2024 |

Actividad 2

| Base de Datos | Modelo de Datos | Escalabilidad | Tolerancia a Fallos | Rendimiento por Consulta |
|----------------------|-----------------|-----------------------|---------------------|---|
| Cassandra | Columnar | Horizontal (linear) | Alta (réplicas) | Alto en escrituras, consultas por clave |
| MongoDB | Documentos | Horizontal (sharding) | Alta con réplicas | Alto en lecturas, consultas flexibles |
| Firestore | Documentos | Automática (Google) | Alta (global) | Tiempo real, latencia baja |
| Neo4j | Grafos | Vertical/Cluster | Media-Alta | Excelente en relaciones complejas |
| Elasticsearch | Documentos | Horizontal | Alta (shards) | Excelente en búsquedas y análisis |
| BigQuery | Columnar | Automática (Google) | Alta | Excelente en análisis batch |

Actividad 3

- El poliglot persistence es generalmente mejor para aplicaciones complejas. Según Fowler (2011), diferentes tipos de datos y patrones de acceso se benefician de diferentes modelos de bases de datos. Por ejemplo, usar PostgreSQL para datos

transaccionales, Redis para caché, y Elasticsearch para búsquedas. Sin embargo, esto aumenta la complejidad operacional.

Fuente: Fowler, M. (2011). NoSQL: Poliglot Persistence.

b. Desventajas principales:

- Falta de estandarización en consultas
- Consistencia eventual en lugar de ACID inmediato
- Menor madurez en herramientas y ecosistema
- Complejidad en joins y relaciones complejas
- Curva de aprendizaje más pronunciada

Fuente: Stonebraker, M. (2010). SQL Databases v. NoSQL Databases.

c. Riesgos:

- Vendor lock-in (dependencia del proveedor)
- Costos impredecibles según el uso
- Limitaciones en personalización y control
- Posibles cambios en pricing o características
- Dependencia de la conectividad a internet

Fuente: Google Cloud Architecture Center (2023). Managing Vendor Lock-in.