

# Análisis de Caso: Bases de Datos No Relacionales

## 1. Resumen Ejecutivo

La empresa de streaming enfrenta problemas de rendimiento y escalabilidad con su base de datos relacional actual. Dado el crecimiento exponencial de usuarios y contenido, se propone migrar a una base de datos NoSQL. Tras un análisis de alternativas, se recomienda utilizar MongoDB, una base de datos orientada a documentos, por su flexibilidad, escalabilidad horizontal y facilidad de integración con los sistemas actuales.

## 2. Análisis del Problema

- Demoras en las consultas: El volumen de datos ha superado la capacidad óptima de la base de datos relacional.
- Escalado vertical limitado y costoso.
- Estructura rígida que dificulta la evolución del modelo de datos.

## 3. Comparación de Tecnologías NoSQL

Tipo	Ejemplos	Ventajas	Desventajas	Aplicabilidad
Key-Value	Redis, DynamoDB	Alta velocidad, ideal para cachés o sesiones	Modelo de datos muy simple	Útil para sesiones de usuario y tokens
Document-Oriented	MongoDB, CouchDB	Flexible, esquemas dinámicos, fácil de escalar	No es ideal para relaciones complejas	Ideal para perfiles de usuario, historial, recomendaciones
Column-Oriented	Cassandra, HBase	Alta eficiencia para consultas en columnas específicas	Complejo de administrar	Bueno para análisis masivo y eventos
Graph-Oriented	Neo4j, ArangoDB	Excelente para relaciones complejas	No es óptimo para datos independientes	Útil en sistemas de recomendación muy complejos o redes sociales

## 4. Propuesta de Solución

**Base de datos recomendada:** MongoDB

### Justificación:

- Modelo flexible: Documentos JSON permiten guardar perfiles de usuario, historial y recomendaciones sin necesidad de esquemas rígidos.
- Escalabilidad horizontal: Sharding automático permite crecer fácilmente en múltiples servidores.
- Rendimiento: Optimizada para consultas frecuentes y variadas.
- Compatibilidad: Fácil integración con aplicaciones web modernas (Node.js, Python, etc.).
- Amplio soporte en la comunidad y documentación robusta.

### Posibles desafíos:

- Requiere un diseño adecuado de índices (para evitar cuellos de botella).
- Migración desde SQL requiere planificación cuidadosa (para mantener la integridad de datos).

## 5. Conclusiones y Recomendaciones Finales

MongoDB se presenta como la mejor opción para la empresa por su equilibrio entre rendimiento, flexibilidad y facilidad de integración.

Se recomienda:

- Iniciar una fase piloto con un subconjunto de datos.
- Capacitar al equipo en modelado documental.
- Planificar una estrategia de migración escalonada.
- Diseñar índices y colecciones optimizadas desde el inicio.