**Análisis de Caso: Bases de Datos No Relacionales**

**1. Resumen Ejecutivo**

La empresa de servicios de streaming ha experimentado un rápido crecimiento en usuarios y contenido multimedia. La base de datos relacional actual está generando problemas de rendimiento y escalabilidad. Se propone migrar a una solución **NoSQL**, recomendando un modelo **Document-Oriented** como **MongoDB**, para mejorar la flexibilidad del modelo de datos, optimizar consultas y escalar horizontalmente de forma rentable.

**2. Análisis del Problema**

* **Problemas actuales:**
  + Consultas lentas debido a la gran cantidad de registros relacionados (usuarios, historiales, recomendaciones).
  + Escalabilidad horizontal compleja y costosa con la base relacional.
  + Modelo rígido para manejar datos semi o no estructurados como recomendaciones personalizadas y preferencias.
* **Necesidad del cambio:**
  + Requerimiento de un sistema que soporte lectura/escritura intensiva y estructuras de datos dinámicas.
  + Reducción de costos de infraestructura mediante particionamiento (sharding) y escalabilidad horizontal.

**3. Comparación de Tecnologías**

| **Tipo de NoSQL** | **Ejemplos** | **Ventajas** | **Desventajas** | **Casos de uso** |
| --- | --- | --- | --- | --- |
| **Key-Value** | Redis, DynamoDB | Velocidad extremadamente alta, almacenamiento en caché | Modelo simple, consultas limitadas | Sesiones de usuario, tokens |
| **Document-Oriented** | MongoDB, CouchDB | Flexible, ideal para datos semiestructurados, fácil de escalar | Consistencia eventual, manejo de transacciones limitado | Perfiles de usuario, historiales, recomendaciones |
| **Column-Oriented** | Cassandra, HBase | Escrituras masivas y alto rendimiento distribuido | Curva de aprendizaje, consistencia eventual | Logs, métricas de visualización |
| **Graph-Oriented** | Neo4j, ArangoDB | Excelente para relaciones complejas entre nodos | Escalabilidad más compleja, menor rendimiento para consultas simples | Redes sociales, recomendaciones basadas en grafos |

**4. Propuesta de Solución**

**Solución recomendada:**  
Adoptar una base de datos **Document-Oriented**, específicamente **MongoDB**, por su flexibilidad para manejar documentos JSON con estructuras variables, lo que permite representar fácilmente historiales de usuario, catálogos de contenido y recomendaciones dinámicas.

**Justificación:**

* **Escalabilidad:** MongoDB permite escalabilidad horizontal nativa mediante particionamiento (sharding).
* **Rendimiento:** Consulta eficiente para grandes volúmenes de datos.
* **Flexibilidad:** Modelo de datos dinámico adaptable a cambios en la estructura de contenido y preferencias de usuario.
* **Integración:** Amplio soporte de drivers y compatibilidad con la arquitectura de microservicios.

**Ventajas:**

* Rendimiento mejorado para consultas y escrituras simultáneas.
* Ahorro de costos operativos al escalar horizontalmente.
* Reducción de tiempos de desarrollo al no requerir esquemas rígidos.

**Desafíos:**

* Migración de datos y sincronización con la base relacional actual.
* Capacitación del equipo para administrar y optimizar la nueva tecnología.

**5. Conclusiones y Recomendaciones Finales**

Para soportar el crecimiento continuo y optimizar la experiencia de usuario, la empresa debe adoptar una solución **NoSQL Document-Oriented**. Se recomienda iniciar una prueba piloto con **MongoDB**, evaluar su desempeño con datos reales y planificar una migración gradual. Además, se sugiere mantener un esquema híbrido en caso de que ciertos procesos críticos requieran operaciones ACID estrictas, usando la base relacional solo donde sea imprescindible.