**1. Resumen Ejecutivo**

* **El Problema:** La empresa está experimentando una ralentización significativa en su plataforma debido a que la base de datos relacional actual no puede soportar el creciente volumen de clientes y transacciones. Esto se traduce en una mala experiencia de usuario y un aumento en los costos operativos, frenando nuestro potencial de crecimiento.
* **Análisis y Solución Propuesta:** Tras un análisis comparativo de tecnologías líderes como PostgreSQL, MongoDB y Amazon DynamoDB, se concluye que un modelo NoSQL gestionado es la solución óptima. Se propone la migración de la infraestructura de datos a **Amazon DynamoDB**.
* **Impacto y Recomendación:** Esta transición eliminará los cuellos de botella de rendimiento, reducirá drásticamente la carga de mantenimiento y alineará los costos de infraestructura con el uso real.

**2. Análisis del Problema**

* **Problemas de Rendimiento:** La base de datos relacional sufre con el aumento de datos. Los JOIN entre diferentes tablas como clientes, productos y transacciones se vuelven cada vez más lentas. La indexación tiene un rendimiento decreciente a gran escala.
* **Problemas de Escalabilidad:** El sistema actual escala verticalmente lo que significa comprar servidores más potentes y caros. Esta estrategia es costosa y tiene un límite físico.
* **Costos de Mantenimiento Crecientes:** Los costos no son solo de hardware. Incluyen el tiempo que el equipo de ingeniería dedica a optimizar consultas, aplicar parches de seguridad, gestionar backups, planificar la capacidad y resolver fallos, en lugar de desarrollar nuevas funcionalidades para el negocio.

**3. Comparación de Tecnologías**

* **Tabla Comparativa de Opciones:**

|  |  |  |  |
| --- | --- | --- | --- |
| Característica | Opción 1: Optimizar PostgreSQL (Relacional Mejorado) | Opción 2: Migrar a MongoDB (NoSQL - Documentos) | Opción 3: Migrar a Amazon DynamoDB (NoSQL - Serverless) |
| Escalabilidad | Moderada. Mejor rendimiento que bases de datos más básicas, pero la escalabilidad horizontal sigue siendo un desafío. | Alta. Diseñada para escalar horizontalmente de forma nativa. Buena para un crecimiento general. | Masiva y Automática. Escalabilidad prácticamente ilimitada. Se gestiona de forma transparente por AWS. |
| Rendimiento | Consultas complejas con datos normalizados. Se degrada bajo alta concurrencia y volumen extremo. | Alto y Flexible. Excelente para leer y escribir objetos complejos (JSON). El rendimiento se mantiene bien al escalar. | Predecible y Constante. Latencia de milisegundos, sin importar la escala. Optimizado para consultas por clave. |
| Modelo de Datos | Relacional. Esquema estricto, ideal para integridad transaccional (ACID). | Documentos. Esquema flexible, ideal para desarrollo ágil y datos semi-estructurados. | Clave-Valor y Documentos. Esquema flexible. Requiere conocer los patrones de acceso de antemano para un diseño óptimo. |
| Costos y Mantenimiento | Altos. Requiere administración activa de servidores, backups, parches. | Moderados. Menos mantenimiento que SQL, pero aún requiere gestión del clúster si es autogestionado. | Bajos. Cero administración de servidores (serverless). El modelo de pago por uso alinea el costo directamente con el tráfico, eliminando costos fijos altos. |

**4. Propuesta de Solución**

* **Solución Recomendada:** **Amazon DynamoDB**
* **Justificación de la Elección:**
  1. **Enfoque en el Negocio, no en la Infraestructura:** Para una "empresa de tecnología en crecimiento", el recurso más valioso es el tiempo de sus ingenieros. DynamoDB, al ser un servicio **cloud**, elimina casi por completo la carga de gestión de la base de datos.
  2. **Escalabilidad para el Futuro:** DynamoDB está diseñado para manejar una escala masiva. Al elegir DynamoDB, la empresa no tendrá que volver a preocuparse por los problemas de escalabilidad, sin importar cuánto crezca.
  3. **Costos Optimizados (Pay-Per-Use):** El modelo de capacidad On Demand de DynamoDB es ideal para una empresa en crecimiento. Durante los períodos de bajo tráfico, los costos son mínimos. Durante los peak de venta, la base de datos escala automáticamente y solo se paga por las transacciones realizadas, lo que ofrece una eficiencia de costos superior al mantenimiento de servidores 24/7.
  4. **Rendimiento Garantizado:** DynamoDB ofrece un rendimiento predecible, lo que se traduce directamente en una experiencia de usuario rápida y fluida, crucial para la retención de clientes.

**5. Conclusiones y Recomendaciones Finales**

* **Conclusión:** La base de datos relacional actual ha cumplido su ciclo y ahora representa un freno para el crecimiento. La migración a una arquitectura cloud con Amazon DynamoDB no es solo una mejora técnica, sino una decisión estratégica que dota a la empresa de la agilidad, escalabilidad y eficiencia de costos necesarias para liderar en el mercado.
* **Recomendaciones Clave:**
  1. **Realizar una Proof of Concept (PoC):** Migrar un servicio específico a DynamoDB para validar en un entorno real los beneficios de rendimiento y costos.
  2. **Capacitar al Equipo:** Invertir en la formación del equipo de desarrollo en los patrones de diseño de datos de DynamoDB, ya que difieren significativamente del modelado relacional.
  3. **Diseñar un Plan de Migración por Fases:** Crear una hoja de ruta para migrar los servicios de forma gradual, comenzando por los que causan más problemas de rendimiento, para minimizar los riesgos.
  4. **Adoptar una Arquitectura Orientada a Eventos:** Aprovechar la integración nativa de DynamoDB con otros servicios de AWS (como Lambda y SQS) para modernizar la arquitectura general de la aplicación, haciéndola más modular y resiliente.