EXÁMEN FINAL

CASE 1

Carlos Ivan Gómez Cardenas 1151600

Marianella Herrera Rondón 1151495

Carlos Daniel Roman Barrera 1151625

Yuber Eduardo Lobo Quintero 1151857

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER
INGENIERÍA DE SISTEMAS
ARQUITECTURA DE SOFTWARE
JULIO 2024

1. DECISIONES DE ARQUITECTURA

ADR1: Base de Datos MySQL

Razones o Justificación: Expensive Stuff For Sale (ESFS) necesita almacenar y gestionar datos relacionados con contratos, solicitudes de propuestas (RFPs) y detalles de empresas contratistas para soportar la operación del clasificador de redes neuronales profundas. MySQL es una solución popular y bien conocida por el equipo de ESFS. Ofrece un modelo relacional adecuado para estructurar datos tabulares con relaciones definidas, lo cual es esencial para manejar los datos estructurados requeridos por el sistema de garantía de transacciones de ESFS.

Consecuencias: MySQL es eficiente para manejar grandes volúmenes de datos y proporciona un rendimiento sólido.
A largo plazo, podría requerir técnicas de particionamiento y optimización para mantener el rendimiento con el crecimiento de la base de datos.

ADR 2: Caché Redis

Razones o Justificación: ESFS busca mejorar el rendimiento al evitar el recálculo repetido de vectores de características mediante el uso de almacenamiento en caché.
Redis es conocido por su velocidad y eficiencia en operaciones de lectura y escritura, lo cual es crucial para acelerar el acceso a datos en memoria y mejorar la respuesta del sistema de gestión de transacciones de ESFS.

Consecuencias: Redis facilita operaciones más rápidas debido a su estructura en memoria. Introduce complejidad adicional que requiere configuración y mantenimiento específico para optimizar su uso.

ADR 3: Implementar el patrón Batch Processing para el cálculo de vectores de características

Razones y justificación: ESFS enfrenta el desafío de realizar cálculos costosos de vectores de características sin afectar el rendimiento del sistema principal en tiempo real.
Separar el cálculo de vectores de características en un proceso por lotes permite escalar horizontalmente y optimizar el rendimiento sin interferir con las operaciones críticas del sistema en tiempo real.

Consecuencias: Mejora la eficiencia y escalabilidad del sistema al ejecutar procesos de

cálculo de manera periódica y desacoplada.
Requiere monitoreo adicional para gestionar eficazmente el flujo de trabajo por lotes.

ADR 4: Arquitectura por Capas

Razones y justificación: ESFS necesita una arquitectura que permita gestionar de manera modular diferentes aspectos funcionales y técnicos del sistema.
Una arquitectura por capas facilita la gestión, mantenimiento y escalabilidad del sistema al proporcionar una estructura modular que permite cambios y actualizaciones sin afectar otras partes del sistema.

Consecuencias: Permite desarrollar, escalar y optimizar cada capa de manera independiente según sea necesario.
Requiere flexibilidad en el diseño inicial para adaptarse a cambios futuros en los requisitos del negocio y tecnológicos.

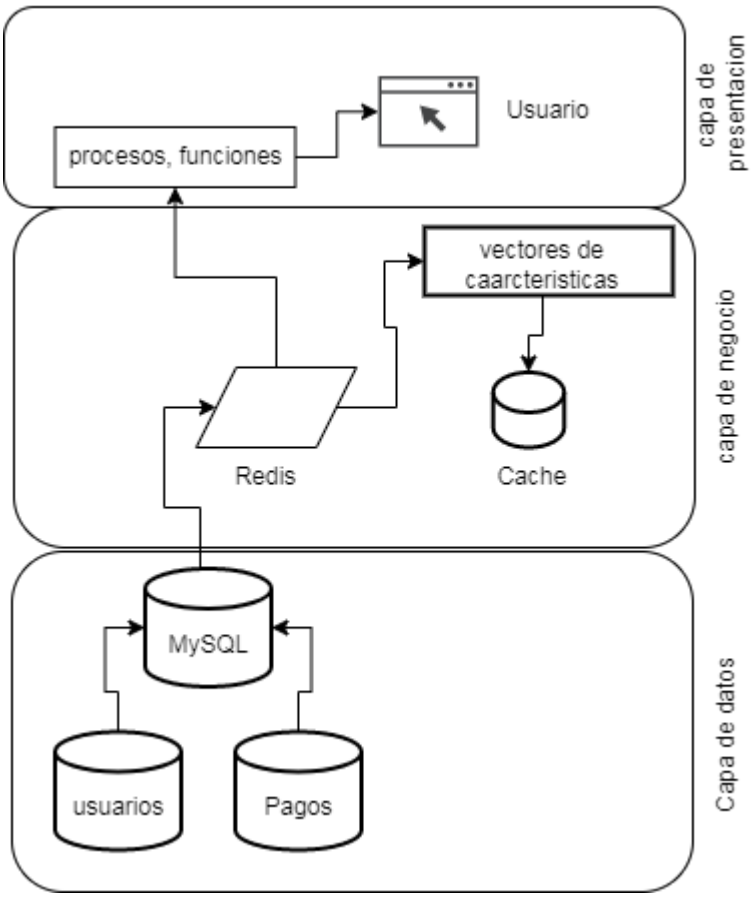
ADR 5: Despliegue en Spring Boot

Razones y justificación: ESFS necesita integrar un clasificador de redes neuronales profundas para evaluar transacciones en tiempo real, manteniendo la capacidad de revertir a evaluaciones manuales si surgen problemas de rendimiento.

Spring Boot proporciona una configuración simplificada y estructuras fáciles de usar que son ideales para integrar aplicaciones Java de manera eficiente. Esto aprovecha la familiaridad del equipo con Java y el ecosistema de Spring.

Consecuencias: Simplifica el desarrollo inicial y la integración del clasificador en el sistema existente.
Requiere monitoreo continuo del rendimiento para optimizar el consumo de memoria, especialmente en operaciones en tiempo real.

Modelo De arquitectura por capas


capa de presentacion
Usuario
procesos, funciones
capa de negocio
vectores de caarcteristicas
Redis
Cache
Capa de datos
MySQL
usuarios
Pagos