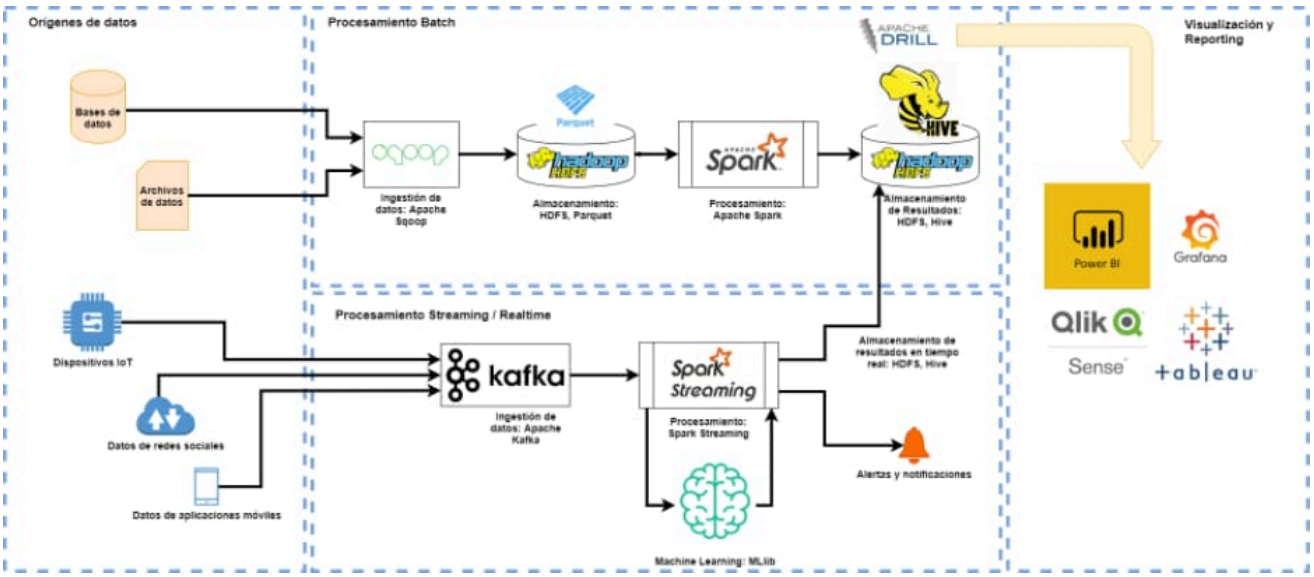
Diseño y Comparación de Arquitecturas de Almacenamiento en Big Data



Adrián Yared Armas de la Nuez

Contenido

1. Objetivo

Desarrollar la capacidad de analizar y diseñar diferentes arquitecturas de almacenamiento en Big Data según los requisitos específicos de una organización.

2. Descripción

En esta actividad, trabajarás como consultor de arquitectura de datos. Deberás analizar sus necesidades y proponer la arquitectura más adecuada para el caso de uso.

3. Caso de estudio

3.1 Datos

Cadena de Supermercados "FreshMart"
- Necesita analizar patrones de compra históricos
- Maneja datos estructurados de ventas, inventario y clientes
- Requiere generar informes diarios de rendimiento
- Tiene 500 tiendas en todo el país
- Procesa 1 millón de transacciones diarias

3.2 Tareas

3.2.1 Tarea 1

3.2.1.1 Enunciado

Proponer una arquitectura de almacenamiento específica (Data Warehouse, Data Lake, Data Mesh o Federación de Datos)
- Justificar la elección basándose en las características y necesidades de la empresa
- Describir las herramientas tecnológicas que se utilizarían
- Identificar posibles desafíos y proponer soluciones

3.2.1.1 Resolución

Como arquitectura de almacenamiento he escogido un Data Warehouse, debido a la necesidad de análisis de patrones históricos, generación de informes diarios y manejo de datos estructurados (ventas, inventario, clientes). Además garantiza accesos rápidos para reportes y análisis centralizados.
Como herramientas que podría utilizar encontramos las siguientes; Amazon Redshift, ya que compone bases de datos analíticas escalables e ideal para gran volumen transaccional como es el caso de FreshMart; Para ETL Apache Airflow podría ser una buena opción debido a que permite procesar y transformar datos desde los sistemas transaccionales al Data Warehouse; para la visualización de datos Power Bi, ya que permite crear informes diarios personalizados e interactivos para cada tienda; finalmente para la gestión de transacciones podría usar Kafka, ya que es una excelente herramienta de manejo del flujo de datos en tiempo real.

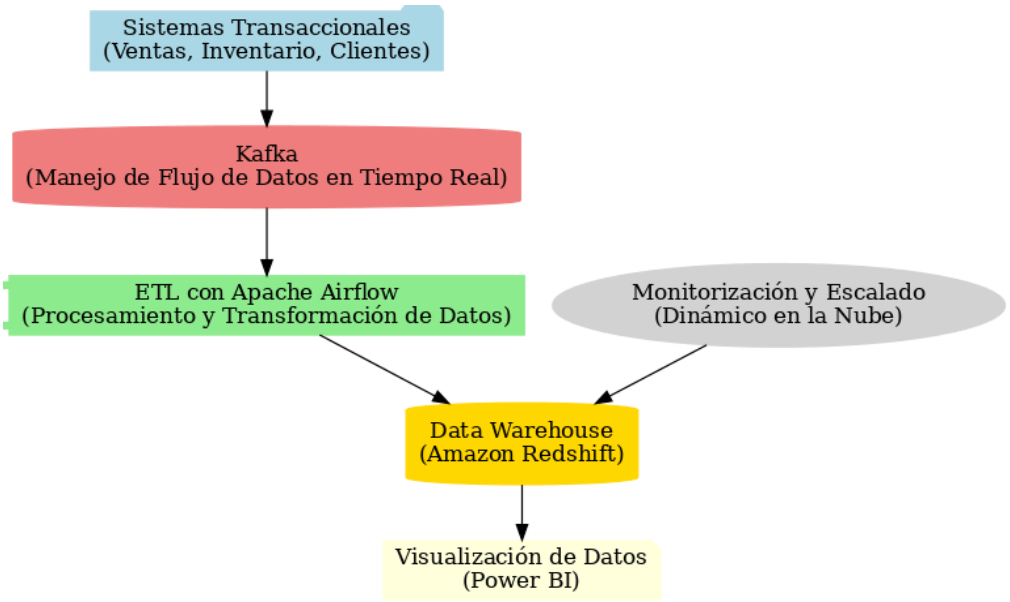
En cuanto a los principales desafíos de la arquitectura propuesta incluyen el manejo del volumen y la velocidad de los datos, la integración de sistemas diversos y los costos operativos. Para optimizar las consultas en el Data Warehouse frente al alto flujo de datos, se implementará un sistema de particionamiento. La integración de fuentes heterogéneas se abordará mediante el uso de APIs estándar y ETL, asegurando la unificación eficiente de los datos, para controlar los costos operativos, se empleará un escalado dinámico en servicios en la nube, complementado con una monitorización constante del uso y su interpretación en herramientas gráficas. Todo esto garantiza rendimiento, escalabilidad y dificultad de fallas, como denotan las necesidades de "FreshMart".

3.2.2 Tarea 2 (Opcional)

3.2.2.1 Enunciado

Crear un diagrama simple de la arquitectura propuesta para cada caso
(puede ser un boceto a mano o utilizando herramientas de diagramación)

3.2.2.1 Resolución



3.2.3 Tarea 3

3.2.3.1 Enunciado

Destaca en una tabla los siguientes ítems de la arquitectura elegida:
- Ventajas y desventajas
- Costos relativos
- Complejidad de implementación
- Escalabilidad
- Mantenimiento

3.2.3.1 Resolución

Ventajas	Centralización de datos estructurados para análisis.	Alto rendimiento en consultas analíticas.	Escalabilidad horizontal y vertical.	Soporte para grandes volúmenes de datos.
Desventajas	Requiere procesos ETL complejos para la integración de datos.	Costos significativos en infraestructura y licencias de herramientas como Redshift o Power BI.	No optimizado para datos en tiempo real.	

Costos relativos	Amazon Redshift: Modelo de costos por hora y almacenamiento, generalmente medio-alto dependiendo del volumen de datos.	Apache Airflow y Kafka: Costos reducidos al ser open-source, pero con inversión inicial en configuración y mantenimiento	Power BI: Costo por usuario mensual o licencia Premium, dependiendo de la escala.	
Complejidad de implementación	Moderada a alta.	Configuración de ETL en Apache Airflow.	Creación de esquemas y particionamiento en el Data Warehouse.	Integración con múltiples fuentes mediante APIs y conectores.
Escalabilidad	Alta escalabilidad.	Amazon Redshift permite agregar nodos fácilmente para manejar más datos.	Los sistemas ETL y Kafka soportan flujos de datos crecientes sin afectar el rendimiento.	
Mantenimiento	Requiere monitoreo continuo para optimizar consultas y recursos.	Necesidad de ajustes regulares en pipelines ETL conforme cambian los formatos de datos de origen.	Dependencia de personal técnico capacitado para gestión y resolución de problemas.	

4. Entregables

1. Documento con el análisis y justificación de la arquitectura
2. Diagrama de la arquitectura elegida. (Actividad opcional)
3. Explicación de la arquitectura propuestas

5. Recursos Recomendados

● Documentación de herramientas: Hadoop, MongoDB, Cassandra, Amazon S3
● Herramientas de diagramación: draw.io, Lucidchart (actividad opcional)
● Ejemplos de casos reales de implementación