MODULO 05 - EJERCICIO 01

ALEXIS YURI M.

Diseña una arquitectura de datos mínima para una app. de delivery que incluya fuentes, almacenamiento, procesamiento, acceso y seguridad.

1. Fuentes de Datos Principales

Usuarios: Nombre, RUT, Dirección, Historial de compras, Formas de pago.

Pedidos: Productos, Cantidad, Estados (pendiente, en ruta, entregado).

Restaurantes: Productos, Stock, Horario de atención, Dirección, Calificaciones de clientes.

Repartidores: Ubicación en tiempo real, Disponibilidad, Calificaciones de clientes.

Logs del sistema: Logs de usuarios, Logs de repartidores, Indicadores(métricas).

2. Tecnologías de Almacenamiento

Se propone un modelo híbrido con base de datos SQL + NoSQL:

PostgreSQL / MySQL (SQL):

Para datos estructurados y transaccionales (usuarios, pedidos, restaurantes).

Justificación: ACID, entrega integridad referencial y manejo de queries complejas.

MongoDB (NoSQL):

Para datos semiestructurados o altamente dinámicos (logs, ubicacion repartidores en tiempo real, historial de compras).

Justificación: alta disponibilidad y flexibilidad.

Amazon S3 / Google Cloud Storage (Data Lake):

Para almacenar datos históricos, archivos, logs.

Justificación: barato, escalable, ideal para posterior análisis de indicadores.

3. Procesamiento de Datos

Se propone realizar procesamiento de datos de la siguiente forma:

Batch ETL (Talend, Apache Nifi):

Consolidación diaria de pedidos y usuarios para reportes del negocio.

Procesamiento en tiempo real (Apache Kafka + Spark Streaming):

Para monitorear Ubicación en tiempo real de repartidores y Estado de los pedidos.

4. Herramientas de Acceso y Visualización.

Se propone usar las siguientes herramientas

APIs REST / GraphQL:

Para exponer funcionalidades a la app móvil y al frontend web.

Power BI / Google Data Studio:

Dashboards para operaciones internas (pedidos por zona, desempeño de repartidores).

Panel administrativo personalizado:

Acceso seguro por roles (usuarios, restaurantes, soporte).

5. Esquema de la Solución de la Arquitectura de Datos y Buenas Prácticas.

Se propone el siguiente diagrama:

[Usuarios/App] [Restaurantes] [Repartidores]

------------> API Gateway <---------------

+---------+----------+

[PostgreSQL/MySQL] [MongoDB]

+---------+----------+

[ETL / Kafka Streams]

+-------------+-------------+

[Data Lake (S3/GCS)] [Dashboards / BI]

[Control de Acceso RBAC]

[Cifrado - Seguridad]

Buenas prácticas propuestas:

- Gobernanza de datos: se definen roles, reglas de acceso (RBAC) y trazabilidad.

- Escalabilidad: bases distribuidas, procesamiento en tiempo real y batch.

- Flexibilidad: uso de NoSQL y APIs desacopladas para fácil evolución.

- Seguridad: cifrado en tránsito/reposo, autenticación y control de acceso.

- Interoperabilidad: uso de estándares (REST, JSON), integración cloud.