Proyecto Integrador – FleetLogix

Índice

1. Introducción

FleetLogix es una empresa de transporte y logística que opera una flota de 200 vehículos realizando entregas de última milla en cinco ciudades principales de Colombia. El objetivo del proyecto es migrar de sistemas *legacy* hacia una infraestructura moderna de datos, basada en *PostgreSQL*, que permita análisis operativos y decisiones en tiempo real.

2. Objetivos

- Poblar una base de datos PostgreSQL con más de *500.000 registros sintéticos realistas*.
- Garantizar *integridad referencial*, *consistencia temporal* y *calidad de datos*.
- Documentar el modelo relacional y generar un *diagrama ER profesional*.
- Diseñar un *modelo dimensional tipo estrella* para análisis OLAP.
- Implementar *queries operativas* y *KPIs logísticos*.
- Proponer una *arquitectura cloud escalable* para análisis en tiempo real.

3. Modelo Relacional (ERD)

3.1 Tablas maestras *vehicles*: define la flota con atributos técnicos. *drivers*: registra conductores con licencias válidas. *routes*: conecta ciudades con distancia y duración estimada.

3.2 Tablas transaccionales *trips*: viajes realizados, vinculando vehículo, conductor y ruta. *deliveries*: entregas asociadas a cada viaje. *maintenance*: registros de mantenimiento por vehículo.

3.3 Relaciones y restricciones trips.vehicle_id → vehicles.vehicle_id trips.driver_id → drivers.driver_id trips.route_id → routes.route_id deliveries.trip_id → trips.trip_id maintenance.vehicle_id → vehicles.vehicle_id

El diagrama ERD completo se incluye en *assets/Diagrama_Fleetlogix.png*.

4. Generación de Datos Sintéticos

4.1 Herramientas utilizadas Python con librerías *Faker*, *pandas*, *numpy*, *psycopg2*.

4.2 Volumen de datos 200 vehículos 400 conductores 50 rutas 100.000 viajes 400.001 entregas 5.000 mantenimientos

4.3 Métodos clave *generate_trips()*: simula viajes con distribución horaria, consumo de combustible y carga coherente y realista. *get_hourly_distribution()*: concentra actividad entre 6:00–20:00, con picos en 8–12 y 14–18. Este método utiliza intervalos de tiempo y frecuencia de ocurrencia del evento dentro de ese intervalo predeterminada por la probabilidad. El funcionamiento está explicado en el archivo *Explicación_generate_trips()gethourly_distribution().pdf*.

5. Validaciones de Calidad

5.1 Integridad referencial Todas las claves foráneas válidas.

5.2 Consistencia temporal arrival_datetime > departure_datetime.

5.3 Reglas de negocio Peso ≤ capacidad del vehículo, tracking numbers únicos, entregas por viaje entre 2 y 6.

5.4 Nota sobre entregas generadas El diseño planteaba 400.000 entregas, pero se generaron *400.001* debido a la distribución probabilística. Este comportamiento es esperado y refleja la naturaleza aleatoria controlada del generador.

Todas las validaciones fueron exitosas.

6. KPIs Operativos

6.1 Métricas calculadas % de entregas a tiempo vs retrasadas Consumo promedio de combustible por tipo de vehículo Utilización de capacidad por viaje Mantenimientos por cada 1.000 km Promedio de entregas por viaje

7. Arquitectura Cloud Propuesta

Ingesta: *PostgreSQL + CDC (Debezium)* Streaming: *Apache Kafka* Procesamiento: *Apache Flink / Spark Streaming* Almacenamiento: *Data Lake + Data Warehouse* Visualización: *Power BI / Looker* Gobernanza: *Data Catalog + Lineage*

8. Ejecución paso a paso

Crear la base *fleetlogix* en PostgreSQL. Ejecutar *fleetlogix_db_schema.sql*. Configurar entorno Python e instalar dependencias. Ejecutar *fleetlogix_data_generator.py*. Validar registros en DBeaver. Revisar logs y resumen. Ejecutar queries de validación y KPIs. Documentar resultados.

11. Conclusiones

El proyecto *FleetLogix* demuestra la capacidad de migrar de sistemas legacy hacia una infraestructura moderna de datos, garantizando calidad, integridad y consistencia. El modelo relacional soporta operaciones logísticas a gran escala, mientras que el modelo dimensional habilita análisis OLAP y dashboards. La arquitectura cloud propuesta asegura escalabilidad y decisiones en tiempo real.

11. Autoría

Elaborado por *Federico Ceballos Torres* como parte del proyecto integrador de la carrera de *Data Science* en *SoyHenry*.