Documentación Técnica del Proyecto

Análisis del Transporte de Pasajeros a Medellín

Autor: Juan David Acevedo Gómez
NRC: 000521113
Asistentes de IA: Claude 4 y en menor medida Gemini 2.5 pro

PostgreSQL 16 | Docker | Python | DBeaver | ETL

Resumen Ejecutivo

Este documento detalla el diseño, desarrollo y despliegue de una solución de base de datos para el análisis de datos de transporte terrestre hacia Medellín. El proyecto fue implementado utilizando PostgreSQL 16 en un entorno contenerizado con Docker, siguiendo las mejores prácticas de la industria para garantizar la reproducibilidad y el aislamiento.

La solución no solo cumple con todos los requisitos funcionales del examen, sino que también incorpora un enfoque profesional en la organización del código y el manejo de datos, como se detalla a continuación.

1. Infraestructura y Conectividad

Etapas 1 y 2

Se seleccionó PostgreSQL 16 por su robustez y sus avanzadas capacidades analíticas. La infraestructura se provisionó mediante un archivo `docker-compose.yml`, que define el servicio de la base de datos, las credenciales y el volumen de datos persistente.

Enfoque de "Infraestructura como Código": Este enfoque permite levantar el entorno completo con un solo comando (`docker-compose up -d`).

La conexión desde el IDE DBeaver se configuró para atacar el puerto `5432` expuesto por el contenedor, utilizando las credenciales definidas en el archivo de composición.

2. Diseño y Estructura del Proyecto

El proyecto se organizó en una estructura de carpetas lógica para separar los datos, los scripts y la documentación, facilitando su mantenimiento y comprensión.

1. Diseño del Modelo de Datos (Etapa 3)

   `scripts/01_modelo_ddl.sql`

   Descripción: Se diseñó un modelo relacional normalizado hasta la Tercera Forma Normal (3NF) para eliminar la redundancia de datos y garantizar la integridad referencial. Esto resultó en la creación de tablas de dimensiones (como `terminales`, `empresas`, `ciudades`) y una tabla de hechos central (`viajes`).

2. Proceso de Carga de Datos (ETL)

   `scripts/02_cargar_datos.py`

   `scripts/requirements.txt`

   Justificación: Para poblar la base de datos con los 82,748 registros del CSV, se optó por un script de Python en lugar de miles de sentencias `INSERT`. Este enfoque de ETL (Extracción, Transformación y Carga) es una práctica estándar en la industria para manejar volúmenes de datos significativos.

Extracción	→	Transformación	→	Carga
Lectura del CSV		Limpieza de datos		Inserción eficiente

   El script automatiza la lectura del archivo, la limpieza de datos y la inserción eficiente en las tablas normalizadas, asegurando un proceso de carga rápido y fiable. Aunque no es un "entregable" explícito en la rúbrica, es la pieza que hace que el análisis sea posible y demuestra un dominio del flujo de datos completo.

3. Lógica Almacenada (Etapa 5)

   `scripts/04_logica_almacenada.sql`

   Justificación: Se implementaron la función `calcular_duracion_promedio` y el procedimiento `actualizar_estadisticas_diarias` como lo requiere la Etapa 5. Se decidió separarlos en su propio archivo en lugar de incluirlos en el DDL principal para mejorar la organización y mantenibilidad del código. Esto permite modificar la lógica de negocio sin alterar la estructura de las tablas.

4. Consultas de Análisis (Etapa 4)

   `scripts/03_consultas_analisis.sql`

   Descripción: Contiene las dos consultas de análisis requeridas, utilizando CTEs y Funciones Ventana para responder a preguntas de negocio complejas sobre el rendimiento de las empresas y las rutas.

3. Proceso de Desarrollo Asistido por IA

La construcción del proyecto se realizó en colaboración con los asistentes de IA Claude y Gemini. El proceso iterativo incluyó:

Generación de Código

Creación de la estructura inicial del proyecto, los scripts SQL (DDL, DML, lógica almacenada) y el script de carga de datos en Python.

Depuración Conjunta

Identificación y resolución de errores de sintaxis en SQL, problemas con el entorno de Python (`ModuleNotFoundError`) y fallos en la lógica de carga de datos.

Refinamiento y Documentación

Mejora de los scripts y generación de la documentación técnica del proyecto.

4. Conclusión

El proyecto se completó exitosamente, cumpliendo todos los requisitos del examen. La solución final es robusta, reproducible y está organizada siguiendo prácticas profesionales. La colaboración con los asistentes de IA fue fundamental para acelerar el desarrollo y, sobre todo, para diagnosticar y resolver eficientemente los desafíos técnicos que surgieron.

Logros destacados:

- Implementación de infraestructura contenerizada con Docker
- Diseño de modelo de datos normalizado (3NF)
- Proceso ETL automatizado para 82,748 registros
- Consultas analíticas avanzadas con CTEs y funciones ventana
- Lógica almacenada optimizada para el negocio