



# **README**

## **Guía general del proyecto**

**Proyecto Integrador - README**

**Autor: Facundo Acosta**

**Fecha: 10/15/2025**



## INDICE

FleetLogix - Sistema de Gestión de Flotas y Data Warehouse .....	3
Descripción del Proyecto .....	3
Objetivos del Proyecto.....	3
Arquitectura del Sistema .....	3
Componentes Principales .....	3
Métricas y Resultados .....	4
Tecnologías Implementadas .....	5
Estructura del Proyecto .....	6
Características Destacadas.....	6
Lecciones Aprendidas .....	7
Próximos Pasos Recomendados .....	7
Autor.....	8



# FleetLogix - Sistema de Gestión de Flotas y Data Warehouse

## Descripción del Proyecto

FleetLogix es un sistema integral de gestión de flotas vehiculares y análisis de datos diseñado específicamente para operaciones logísticas en Argentina. El proyecto abarca desde la base de datos transaccional hasta un Data Warehouse analítico en Snowflake, implementando mejores prácticas de ingeniería de datos y arquitectura cloud.

## Objetivos del Proyecto

- Migrar de un sistema transaccional PostgreSQL a un Data Warehouse analítico en Snowflake
- Implementar un pipeline ETL robusto y automatizado
- Garantizar la calidad e integridad de los datos mediante validaciones exhaustivas
- Optimizar el performance de consultas para reporting ejecutivo
- Diseñar una arquitectura escalable en AWS para operaciones en tiempo real

## Arquitectura del Sistema

### Componentes Principales

#### **Base de Datos Transaccional (PostgreSQL)**

- 7 tablas principales diseñadas para operaciones logísticas
- 506,650 registros sintéticos generados
- Modelo híbrido transaccional-dimENSIONAL
- SCD Type 2 para tracking histórico en vehículos y conductores



### **Data Warehouse (Snowflake)**

- Modelo estrella con FACT\_DELIVERIES como tabla de hechos
- Dimensiones: fecha, vehículo, conductor, ruta, cliente
- Pipeline ETL automatizado en Python
- Estrategia UPSERT dual para prevenir duplicados

### **Arquitectura AWS (Propuesta)**

- RDS PostgreSQL para datos históricos
- DynamoDB para operaciones en tiempo real
- Lambda Functions para procesamiento serverless
- S3 para almacenamiento y backups

## **Métricas y Resultados**

### **Volúmenes de Datos**

- Vehículos: 200 registros
- Conductores: 400 registros
- Viajes: 100,000 registros
- Entregas: 400,000 registros
- Clientes: 1,000 registros

### **Performance ETL**

- Tiempo ejecución: 48-49 segundos (proceso completo)
- Registros procesados: 1,898 sin duplicados
- Eficiencia UPSERT: 100% de integridad referencial



## **Métricas de Negocio**

- Entregas a tiempo: 32.9%
- Eficiencia combustible: 2.7 km/L
- Rentabilidad analizada: \$289,075.88

## **Tecnologías Implementadas**

### **Backend y Base de Datos**

- PostgreSQL 15.4 - Base de datos transaccional
- Snowflake - Data Warehouse analítico
- Python 3.8+ - Pipeline ETL y generación de datos
- SQLAlchemy - ORM y conexiones a bases de datos

### **Librerías Python Principales**

- Pandas - Manipulación y transformación de datos
- Faker - Generación de datos sintéticos realistas
- NumPy - Distribuciones probabilísticas avanzadas
- Boto3 - Integración con AWS (simulada)

### **Arquitectura Data**

- Modelo Dimensional - Esquema estrella optimizado para análisis
- ETL Automatizado - Pipeline en Python con logging completo
- Validación Multi-nivel - Calidad de datos en generación y carga



## Estructura del Proyecto

```
ProyectoM2_FacundoAcosta/
|
├── Scripts/
|   ├── 01_data_generation/           # Script generación 50k+ registros (Avance 1)
|   ├── 02_y_03_queries_optimization/ # 12 queries + optimización de índices (Avance 2)
|   ├── 04_dimensional_model/         # DDL del Data Warehouse (Avance 3)
|   ├── 05_etl_pipeline/              # Pipeline ETL completo + script de consultas (Avance 3)
|   ├── 06_aws_setup/                 # Pipeline AWS setup + Simulador (Avance 4)
|   └── lambda_functions.py            # Funciones Lambda (Avance 4)
|
├── Documentación/
|   ├── FA_README.pdf                 # Guía general del proyecto
|   ├── FA_FleetLogix_ER_Diagram.JPG  # Documento de entrega (Avance 1)
|   ├── FA_Análisis_del_modelo_proporcionado.pdf # Documento de entrega (Avance 1)
|   ├── FA_Manual_Consultas_SQL.pdf   # Documento de entrega (Avance 2)
|   ├── FA_Análisis_Snowflake_ETL.pdf # Documento de entrega (Avance 3)
|   ├── FA_aws_architecture_diagram.png # Documento de entrega (Avance 4)
|   └── FA_AWS_Análisis_Arquitectura.pdf # Documento de entrega (Avance 4)
```

## Características Destacadas

### Generación de Datos Sintéticos Avanzada

- Distribuciones probabilísticas realistas
- Coherencia temporal garantizada
- Integridad referencial 100%
- Patrones operativos específicos para logística argentina

### Pipeline ETL Robusto

- Estrategia UPSERT dual: MERGE nativo + fallback DELETE/INSERT
- Procesamiento por lotes para optimización de memoria
- Manejo explícito de tipos de datos entre sistemas
- Logging detallado para monitoreo y debugging

### Herramientas de Verificación



- FA\_snowflake\_verify\_3: Interfaz interactiva para consultas
- Detección automática de columnas disponibles
- 10 opciones predefinidas para análisis comunes
- Modo SQL interactivo para consultas personalizadas

### **Optimización de Performance**

- 12 índices estratégicos implementados y evaluados
- Análisis EXPLAIN ANALYZE para todas las queries
- Mejora selectiva en queries complejas (hasta 63.6%)
- Recomendaciones específicas basadas en métricas reales

## **Lecciones Aprendidas**

### **Técnicas**

- "Más índices ≠ mejor performance" - Evaluar relación costo-beneficio por query
- MERGE de Snowflake es la solución más eficiente para UPSERT
- Importancia del fallback en ingeniería de datos
- Manejo explícito de tipos de datos entre Pandas/Python/SQL

### **Arquitectónicas**

- SCD Type 2 esencial para análisis histórico
- Arquitectura híbrida optimiza costos y performance
- Validación multi-fase garantiza calidad de datos
- Simulación local permite desarrollo sin dependencia de cloud

## **Próximos Pasos Recomendados**

### **Inmediatos**



- Implementar materialized views para métricas pre-calculadas
- Establecer dashboard ejecutivo en Tableau/Power BI
- Automatizar monitoreo de calidad de datos
- Implementar alertas proactivas para anomalies

### **Estratégicos**

- Migración gradual a arquitectura AWS propuesta
- Implementación de machine learning para predictivos
- Expansión a otras regiones de Latinoamérica
- Integración con sistemas de telemetría vehicular

### **Autor**

#### **Facundo Acosta**

Data Scientist

Email: [facundoacostast@gmail.com](mailto:facundoacostast@gmail.com)

LinkedIn: [linkedin.com/in/facundo-acosta-marketing/](https://www.linkedin.com/in/facundo-acosta-marketing/)