Especificación de Proyecto Final

Dashboard Inteligente con Streamlit & LLM (Groq)

Curso: Fundamentos Ciencia de Datos
Docente: Jorge Iván Padilla-Buriticá
Universidad EAFIT - Periodo 2026-1

Objetivo Académico

Integrar el ciclo completo de la Ciencia de Datos (ETL, EDA, Despliegue) con la Inteligencia Artificial Generativa. El estudiante deberá construir una aplicación web capaz de ingerir datos crudos, procesarlos y, mediante un modelo LLM, ofrecer recomendaciones estratégicas de negocio.

1 Definición del Reto y Datos

El estudiante actuará como un consultor de datos senior. Deberá seleccionar un conjunto de datos real (Open Data, Kaggle, Datos Gubernamentales) que cumpla estrictamente con las siguientes condiciones para garantizar complejidad estadística:

1.1 Requisitos del Dataset

- **Volumen:** Mínimo **1000 registros** (filas).
- **Dimensionalidad:** Mínimo **10 columnas/variables**, distribuidas así:
  - Variables Numéricas (Continuas/Discretas).
  - Variables Categóricas (Nominales/Ordinales).
  - Variables Booleanas y/o Temporales (Fechas).
- **Estado:** El dataset debe contener imperfecciones (nulos, outliers o formatos inconsistentes) para justificar el módulo de limpieza.

1.2 Preguntas de Negocio

El Dashboard no es solo visualización; debe responder a **tres preguntas estratégicas** del dominio elegido. Ejemplos:

- *¿Qué factores correlacionan más con la deserción de clientes (Churn)?*
- *¿Existe estacionalidad en las ventas y cómo afecta el inventario?*
- *¿Cómo impacta la variable X en la rentabilidad final?*

2 Arquitectura Técnica del Dashboard

La solución debe desplegarse utilizando **Streamlit** y debe contar con una barra lateral (`st.sidebar`) para navegación y configuración.

2.1 Módulo 1: Ingesta y Procesamiento (ETL)

La aplicación debe permitir la carga dinámica de datos:

- **Fuentes:** Soporte para archivos CSV (Upload), JSON o lectura directa desde una URL.
- **Limpieza Interactiva:**
  - Checkbox para eliminar duplicados.
  - Selectbox para elegir método de imputación (Media, Mediana, Cero) en variables numéricas.
  - Detección y tratamiento de valores atípicos (Outliers).
- **Feature Engineering:** Crear al menos una nueva columna calculada a partir de las existentes (ej: `Ticket Promedio = Ventas / Cantidad`).

2.2 Módulo 2: Visualización Dinámica (EDA)

El usuario debe tener control sobre los gráficos (no imágenes estáticas).

- **Filtros Globales:** Rango de fechas, Categorías, Slider de valores numéricos.
- **Gráficos Requeridos:**
  - Distribuciones (Histogramas/Boxplots) usando Plotly.
  - Correlaciones (Heatmap).
  - Evolución temporal (Line/Area Charts).
- **Interactividad:** Uso de pestañas (`st.tabs`) para organizar Análisis Univariado, Bivariado y Reporte.

3 Inteligencia Artificial: Integración con Groq

El Factor Diferencial: AI-Driven Insights

La aplicación debe conectarse a la API de **Groq** (modelos Llama-3 o Mixtral) para actuar como un analista virtual.

Funcionalidad Requerida: Incluir un botón *Generar Insights con IA* que:

1. Tome el resumen estadístico del DataFrame filtrado (`df.describe()`).
2. Envíe estos datos al LLM mediante un prompt estructurado.
3. Muestre en pantalla una interpretación en lenguaje natural sobre tendencias, riesgos y oportunidades detectadas en los datos.

4 Buenas Prácticas y Entregables

4.1 Estructura del Repositorio (GitHub)

El proyecto debe alojarse en un repositorio público con la siguiente estructura limpia:

```
nombre-proyecto-final/
|-- data/                    # Dataset de muestra (si aplica)
|-- .streamlit/              # Configuración de tema (colores EAFIT)
|-- app.py                   # Código principal de la aplicación
|-- requirements.txt         # Dependencias (streamlit, pandas, groq,
   plotly)
|-- README.md                # Documentación completa
|-- manual_usuario.pdf       # Guía PDF para el usuario final
```

4.2 Contenido del README.md

No es solo un título. Debe incluir:

- **Descripción del Problema:** Contexto de negocio.
- **Instalación:** Pasos para clonar y ejecutar localmente.
- **Link al Despliegue:** URL de Streamlit Cloud funcionando.
- **Créditos:** Autor y fuentes de datos.

4.3 Diseño y UX

- Use `st.columns` para evitar el scrolling infinito.
- Use `st.expander` para ocultar tablas de datos crudos extensas.
- Maneje errores (`try-except`) para que la app no colapse si el usuario sube un archivo incorrecto.

5 Cronograma de Entrega

Hito	Fecha Límite
Repositorio Funcional	10 de Febrero de 2026, 23:59 hrs
Despliegue en Nube	10 de Febrero de 2026, 23:59 hrs
Sustentación	Envíar repositorio por correo para agendar vía TEAMS

La tecnología por sí sola no genera valor; es la capacidad de usarla para responder las preguntas correctas lo que define a un Científico de Datos.