1. Descripción General

Este proyecto tiene como objetivo analizar y comprender los factores que influyen en los accidentes automovilísticos utilizando diferentes fuentes de datos (datasets). Se busca identificar patrones relacionados con condiciones climáticas, experiencia del conductor, tipo de vehículo, iluminación, y otros factores relevantes para proponer estrategias de prevención.

El sistema desarrollado permite **cargar, limpiar, procesar y visualizar datos** mediante herramientas de análisis.

---

2. **Integrantes**

Luis Gerardo López Briseño 218870894

Abel Andrés Hernández 218441993

Daniel García López 222308793

---

3. Datasets Utilizados

Dataset 1 – Accidentes viales del Reino Unido

- **Campos principales:**
  - accident_fields_reference_number, grid_ref_easting, grid_ref_northing
  - number_of_vehicles, accident_date, time_(24hr)
  - road_surface, lighting_conditions, weather_conditions
  - local_authority, casualty_class, casualty_severity, sex_of_casualty
- **Origen:** Portal público *UK Department for Transport (DfT) — Road Safety Data*.
- **Año aproximado:** 2015–2020.
- **Descripción:** Contiene registros georreferenciados de accidentes con variables ambientales, de carretera y de víctimas.

Dataset 2 – Accidentes por factores humanos y viales

- **Campos principales:**
  - time, day_of_week, age_band_of_driver, sex_of_driver
  - educational_level, driving_experience, type_of_vehicle
  - road_surface_type, light_conditions, weather_conditions
  - cause_of_accident, accident_severity
- **Origen:** *Open Data Portal de Etiopía o Kenya* (frecuente en estudios de conducción en África).
- **Descripción:** Contiene datos sobre la influencia de factores humanos (edad, educación, experiencia, etc.) en la ocurrencia de accidentes.

Dataset 3 – Accidentes en EE. UU. (Chicago Crash Data)

- **Campos principales:**
  - crash_date, traffic_control_device, weather_condition, lighting_condition
  - first_crash_type, roadway_surface_cond, damage, injuries_total
  - crash_hour, crash_day_of_week, crash_month
- **Origen:** *Chicago Data Portal – Traffic Crashes Dataset*.
- **Descripción:** Reportes oficiales de accidentes con información detallada sobre las condiciones de la vía, causas primarias y severidad de lesiones.

---

4. Tecnologías Utilizadas

Tipo	Herramienta	Descripción
Lenguaje principal	Python 3.11	Procesamiento, análisis y visualización de datos.
Framework de análisis	Pandas, NumPy	Limpieza y manipulación de grandes volúmenes de datos.
Visualización	Matplotlib, Plotly	Creación de gráficos estadísticos y mapas de calor.

Tipo	Herramienta	Descripción
Entorno de desarrollo	Jupyter Notebook / VS Code	Entorno de análisis interactivo.

---

5. Estructura de Proyecto

5.1. Carpeta raíz del proyecto

Contiene todos los archivos principales y subcarpetas. Desde aquí se ejecuta la aplicación y se almacenan los datos y scripts más importantes.

- **ProyectoAlmacenesdb/**
  Es la carpeta principal del proyecto. Dentro de ella están todos los componentes: el código fuente, los datos, las pruebas y la documentación.

5.2. Carpeta app/

Esta carpeta contiene toda la **lógica del sistema** y el **código de la aplicación FastAPI**, así como los módulos relacionados con el proceso **ETL (Extracción, Transformación y Carga)**.

Archivo main.py

- Es el **punto de entrada de la API** desarrollada con **FastAPI**.
- Se encarga de levantar el servidor web y definir los **endpoints** o rutas para que el usuario pueda:
  - Cargar archivos (por ejemplo, CSV o Excel) mediante solicitudes **POST**.
  - Consultar el estado del proceso ETL o los resultados del análisis.

En otras palabras, este archivo conecta la interfaz del usuario (API) con la lógica del procesamiento de datos.

Carpeta etl/

Aquí se implementan las tres etapas principales del pipeline ETL:
Extracción, **Transformación** y **Carga**.
Cada archivo cumple un rol definido:

- **extract.py**
  Contiene las funciones para **extraer los datos** desde diferentes fuentes (archivos CSV, Excel o JSON).
  También valida que los archivos sean válidos y tengan el formato correcto antes de procesarlos.
- **transform.py**
  Se encarga de la **limpieza y normalización** de los datos.
  Aquí se corrigen valores nulos, se eliminan duplicados, se unifican nombres de columnas y se aplican transformaciones (por ejemplo, convertir fechas o tipos numéricos).
- **load.py**
  Contiene la lógica para **cargar los datos ya transformados** hacia el **Data Warehouse** o un archivo CSV limpio.
  Es la última fase del proceso ETL y prepara los datos para su uso posterior en análisis o reportes.
- **utils.py**
  Archivo de **funciones auxiliares** reutilizables.
  Puede incluir tareas comunes como registro de logs, validaciones o cálculos estadísticos menores.

Carpeta models/

Guarda las **estructuras de datos** y modelos utilizados por la API y el ETL.
Por ejemplo:

- Definiciones de clases con Pydantic (para validar los datos que se reciben).
- Modelos de respuesta de la API.
- Plantillas para el formato de salida de los datos limpios.

5.3. Carpeta data/

Aquí se almacenan los archivos que se van a procesar. Está dividida en dos subcarpetas, para mantener un flujo claro entre los datos **sin limpiar** y los **procesados**.

- **data/raw/**
  Contiene los archivos **originales o crudos**, tal como fueron obtenidos de las fuentes (pueden tener errores, valores faltantes o formatos distintos).
  Ejemplo: dataset1_original.csv, dataset2.json, etc.
- **data/clean/**
  Almacena los archivos que ya fueron **limpiados y transformados** durante el proceso ETL.
  Estos son los que se usan en análisis posteriores o en la construcción del data warehouse.

5.4. Carpeta tests/

Incluye los **scripts de prueba** utilizados para verificar que las funciones del proyecto (especialmente las del ETL y la API) funcionen correctamente.
Por ejemplo:

- Pruebas de carga de archivos.
- Validaciones de los datos después de transformarlos.
- Pruebas de endpoints de la API con FastAPI.

Esto ayuda a asegurar la **calidad del sistema** antes de implementarlo o hacer modificaciones.

5.5. Archivo README.md

Es el documento de **presentación del proyecto**.
Contiene una descripción general, instrucciones para instalar dependencias, ejecutar la API y entender el propósito del sistema.
Es el primer archivo que se lee cuando alguien abre el repositorio en GitHub.

---

6. Manual de Usuario

Instalación

1. Instalar dependencias:
2. pip install pandas numpy matplotlib seaborn scikit-learn mysql-connector-python
3. Importar los datasets .csv o .xlsx en la carpeta /data.
4. Ejecutar el archivo principal del proyecto:
5. python main.py

Uso

1. **Inicio:** El programa carga los tres datasets y los limpia automáticamente.
2. **Menú principal:**
    - Opción 1: Mostrar resumen estadístico.
    - Opción 2: Visualizar correlaciones (por clima, hora o edad del conductor).
    - Opción 3: Guardar datos en la base SQL.
    - Opción 4: Generar reporte gráfico PDF.
3. **Salida:** El sistema genera gráficos y reportes en la carpeta /output.

---

7. Casos de Uso

Caso 1 – Identificación de factores de riesgo

Actor: Analista de tráfico.
Descripción: El usuario carga el dataset y selecciona el análisis por “condiciones climáticas”.
Resultado: Se obtiene un gráfico que muestra el incremento de accidentes en condiciones lluviosas y nocturnas.

Caso 2 – Predicción de severidad

Actor: Investigador.
Descripción: Usa el modelo de clasificación (Random Forest) para predecir la severidad del accidente según variables como velocidad, tipo de vehículo y visibilidad.
Resultado: Se genera un informe con precisión del modelo y ranking de variables más influyentes.

Caso 3 – Análisis comparativo entre países

Actor: Estudiante universitario.
Descripción: Compara los accidentes del dataset del Reino Unido y de EE. UU. según condiciones de iluminación y clima.
Resultado: Identifica diferencias culturales y estructurales en los accidentes automovilísticos.

Caso 4 – Visualización geográfica

Actor: Autoridad vial.
Descripción: Se exportan coordenadas (grid_ref_easting, grid_ref_northing) del dataset 1 a un mapa de calor con GeoPandas.
Resultado: Se observa concentración de accidentes en zonas urbanas.

---

8. Conclusiones

- La combinación de datasets internacionales permite obtener una visión amplia del fenómeno vial.
- Las condiciones de clima, iluminación y experiencia del conductor son factores determinantes.
- La base de datos relacional facilita el manejo eficiente de la información para futuras integraciones o modelos predictivos.