Andrés Mauricio Plazas González

PREICA2502B020074

Programa Académico:

Ingeniería De Software y Datos

Actividades:

Proyecto Integrado (EA1).

Formulación de una necesidad de ingeniería de datos

Asignatura:

Proyecto Integrado V

Docente:

Andrés Felipe Callejas

Institución Universidad Digital de Antioquia

08 de noviembre del 2025

Introducción

Este proyecto tiene como objetivo principal implementar una solución técnica que aborde la necesidad de consolidación, estructuración y persistencia de datos de siniestralidad laboral. Para ello, se ha seleccionado un conjunto de datos público que detalla indicadores de gestión de las ARL. La metodología se centrará en la construcción de una base de datos local utilizando SQLite, demostrando la capacidad de migrar un archivo plano (CSV) a una estructura relacional más robusta y consultable, cumpliendo así con los estándares de un flujo de Extracción, Transformación y Carga (ETL) simplificado.

El desarrollo de este repositorio de datos se enmarca en la aplicación práctica de los conceptos de modelado de bases de datos y programación en Python, lo que permitirá un análisis inicial enfocado en correlacionar la actividad económica de las empresas con la frecuencia y severidad de los incidentes reportados, particularmente en lo referente a las muertes por accidente de trabajo (MUERTES_REPOR_AT) y los presuntos accidentes (PRESUACCIDETRASUCE). La planificación de este proceso se detalla mediante un Diagrama de Gantt, y la documentación del flujo de trabajo y los resultados iniciales se presenta bajo las Normas APA.

Resumen

El proyecto de analítica se centra en la gestión, procesamiento y documentación de datos relacionados con la siniestralidad laboral en Colombia, específicamente los reportados por las Administradoras de Riesgos Laborales (ARL).

1. **Propósito de la Actividad**

   La meta principal es demostrar la capacidad de transformar datos brutos y dispersos (un archivo CSV) en información estructurada y lista para el análisis, a través de un proceso de Extracción, Transformación y Carga (ETL) automatizado. El contexto temático es crucial, ya que busca correlacionar la actividad económica (ACTIVEC) con indicadores de riesgo como muertes por accidente de trabajo (MUERTES_REPOR_AT) y presuntos accidentes (PRESUACCIDETRASUCE).

2. **Proceso Técnico (ETL con Python)**

   Fuente de Datos: Se utiliza el archivo CSV denominado Estadisticas_Riegos_Laborales_Positiva-Sep_2025.csv, que contiene caracteres especiales propios del idioma español (tildes, eñes).

   Manejo de Datos: El script main.py (escrito en Python) es el motor del proyecto. Este script se encarga de:

   - Detectar y corregir automáticamente la codificación del archivo (pasando de UTF-8 a codificaciones compatibles con el español como latin-1).
   - Crear una base de datos relacional local llamada db/proyecto.db (utilizando SQLite).
   - Insertar los registros del CSV en la tabla reporte_arl.
   - Exportar la tabla ya limpia y estructurada a un nuevo archivo CSV de salida: db/export.csv.

3. **Entregables Requeridos**

   La actividad culmina con la presentación de la documentación y evidencias fundamentales que cubren la rúbrica del proyecto:

   - **Evidencia Técnica (GitHub):** Un repositorio que aloja el código fuente (main.py), el dataset original (data/), y los artefactos generados (db/proyecto.db y db/export.csv).
   - **Planificación:** Un excel y junto con un archivo pod que se maneja para el detalle del cronograma de trabajo, estableciendo las fases, fechas de inicio y fin, y los entregables asociados a la metodología.

Objetivo General

Diseñar, implementar y documentar un flujo completo de análisis de datos (ETL) sobre siniestralidad laboral utilizando Python y SQLite, con el propósito de consolidar reportes de riesgos laborales dispersos de la Administradora de Riesgos Laborales (ARL) en una estructura persistente y local, facilitando la extracción de indicadores clave —como la relación entre la actividad económica (ACTIVEC) y los incidentes fatales (MUERTES_REPOR_AT)— para su posterior visualización y toma de decisiones informadas.

El objetivo de este proyecto es aplicar conceptos de análisis de datos para construir una solución local y reproducible para la gestión y consulta de datos. Estamos simulando el rol de un analista que necesita consolidar información dispersa sobre seguridad laboral.

Flujo de la Actividad: El proyecto abarca las etapas clave de la analítica de datos:

CSV (Fuente) -> Python (ETL y Limpieza) -> SQLite (Almacenamiento) -> CSV (Exportación)

Objetivo Especifico

1. **Asegurar la Ingesta de Datos:** Desarrollar funciones en Python para leer el dataset fuente (Estadisticas_Riegos_Laborales_Positiva-Sep_2025.csv), implementando el manejo de codificación (latin-1/cp1252) y la detección de delimitadores para garantizar la integridad de los datos.
2. **Implementar el Modelo de Datos:** Diseñar y crear la tabla reporte_arl dentro de una base de datos SQLite, estableciendo los tipos de datos correctos para persistir las variables clave de siniestralidad.
3. **Ejecutar el Flujo ETL:** Codificar y ejecutar el script principal (main.py) que migre los datos limpios desde el CSV a la base de datos SQLite, verificando la carga completa de registros.
4. **Generar el Artefacto Analítico:** Exportar un reporte consolidado en formato CSV (db/export.csv) directamente desde la base de datos SQLite, listo para el consumo por parte de herramientas de visualización.
5. **Completar la Documentación:** Elaborar los documentos formales de entrega, incluyendo el Informe APA y el Diagrama de Gantt de planificación.

Metodología

La metodología se estructura siguiendo un modelo secuencial de gestión de datos, conocido como ETL (Extracción, Transformación y Carga), complementado con una fase inicial de planificación y una fase final de documentación.

1. **Planificación (Fase Previa)**

La planificación se llevó a cabo mediante la elaboración de un Diagrama de Gantt (Ver Anexo A) para definir el alcance, las dependencias y el cronograma del proyecto. Se establecieron las métricas de interés, enfocadas en la relación entre la Actividad Económica (ACTIVEC) y la fatalidad (MUERTES_REPOR_AT).

2. **Extracción y Normalización de Datos**

La fuente de datos es el archivo plano Estadisticas_Riegos_Laborales_Positiva-Sep_2025.csv. Esta fase se centró en la robustez del script main.py para manejar problemas de formato comunes en datasets de origen gubernamental:

- **Manejo de Codificación:** Debido a la presencia de caracteres especiales del español (ñ, tildes), se implementó un mecanismo de prueba de codificación automática (UTF-8, latin-1, cp1252), asegurando que la ingestión no fallara.
- **Lectura y Delimitación:** Se utilizó el módulo csv de Python junto con el objeto Sniffer para detectar el delimitador correcto (coma o punto y coma), garantizando que las columnas fueran leídas correctamente.

3. **Modelado y Carga (Transformación y Persistencia)**

Los datos extraídos se cargaron en una base de datos relacional SQLite, implementada localmente en el archivo db/proyecto.db.

- **Modelo de Datos:** Se creó la tabla única reporte_arl. Se asignaron tipos de datos TEXT para las variables categóricas (como ACTIVEC y DPTO) y INTEGER para las métricas cuantitativas clave (como MUERTES_REPOR_AT y RELA_DEP), asegurando la integridad numérica de los indicadores de siniestralidad.
- **Proceso de Carga:** La inserción de los registros se ejecutó mediante sentencias SQL, completando la fase de Carga del proceso ETL.

4. **Generación de Artefactos (Salida)**

La fase final consiste en generar los artefactos necesarios para la entrega y para el consumo analítico:

- **Base de Datos Persistente:** Se genera y se mantiene el archivo db/proyecto.db como la fuente de verdad local.
- **Exportación Analítica:** Se realiza una consulta SELECT * sobre la tabla reporte_arl y se exporta el resultado a un nuevo archivo CSV, db/export.csv, que se considera el producto final listo para ser consumido por herramientas externas de visualización de datos (ej., Power BI o Tableau).

Resultados

La ejecución exitosa del flujo ETL descrito en la metodología genera los siguientes artefactos de desarrollo y productos técnicos, demostrando la operatividad del sistema de ingesta y persistencia de datos:

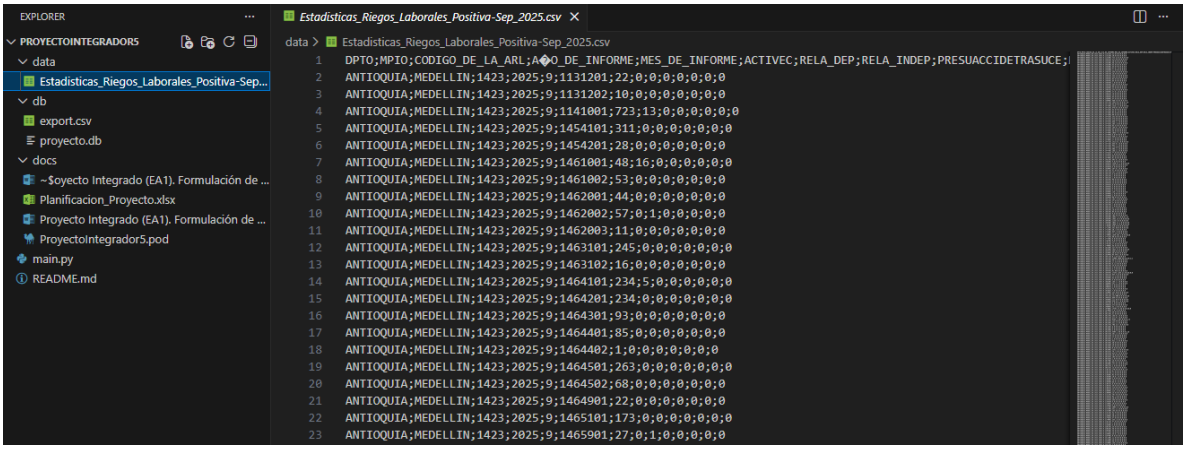
Artefactos de Desarrollo

- **Base de Datos Consolidada:** Generación del archivo db/proyecto.db. Este artefacto valida la fase de Carga, garantizando la persistencia de los datos del CSV en un formato relacional SQLite.
- **Reporte Normalizado de Salida:** Creación del archivo db/export.csv. Este archivo es el producto final del proceso ETL y demuestra la capacidad del script para exportar datos limpios y estructurados desde la base de datos.
- **Módulo de Ingesta Robusto:** El script main.py completamente funcional y parametrizado. Este código fuente valida la fase de Extracción y Transformación, ya que incluye la lógica necesaria para:
    - ✓ *Manejar y corregir automáticamente problemas de codificación (latin-1 / cp1252) del dataset fuente.*
    - ✓ *Crear la estructura de la tabla reporte_arl con tipos de datos definidos.*
    - ✓ *Migrar todos los registros del CSV a la base de datos de manera reproducible.*

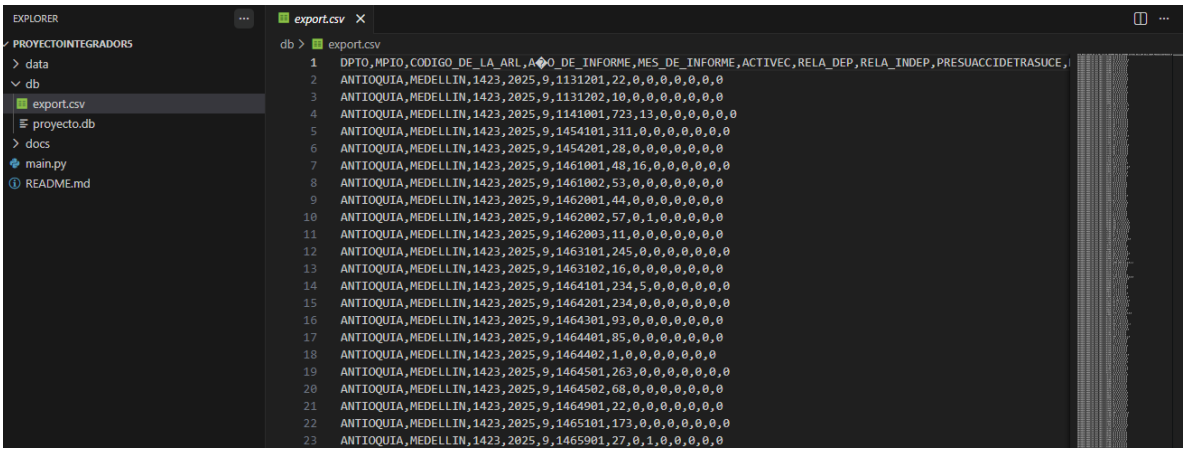
Función del Campo	Nombre de la Columna (ID Interno)	Descripción	Tipo de Dato (SQLite)
Identificación	id	Clave Primaria Auto-incremental	INTEGER (PK)
Identificación	codigo_de_la_arl	Código de la administradora de riesgos laborales	TEXT
Temporal	a_o_de_informe	Año del informe de la data publicada	TEXT
Temporal	mes_de_informe	Mes del informe de la información	TEXT
Geografía	dpto	Departamento de Colombia	TEXT
Geografía	mpio	Municipio de Colombia	TEXT
Segmentación	activec	Actividad económica del afiliado	TEXT
Población	rela_dep	Relación trabajadores dependientes	INTEGER
Población	rela_indep	Relaciones laborales trabajadores independientes	INTEGER
Métrica (Frecuencia)	presuaccidetrasuce	Presunto accidente de trabajo sucedido	INTEGER
Métrica (Severidad)	muertes_repor_at	Muertes de trabajadores reportadas por accidente de trabajo	INTEGER
Métrica (Consecuencia)	nuevapensioinva_r_at	Nueva pensión Accidentes de trabajo	INTEGER
Métrica (Consecuencia)	nuevapensioinva_r_el	Nueva pensión Enfermedad laboral	INTEGER
Métrica (Consecuencia)	incapermaparciar_at	Incapacidad permanente parcial reportada por accidente de trabajo	INTEGER
Métrica (Consecuencia)	incapermaparciar_el	Incapacidad permanente parcial por enfermedad laboral	INTEGER

Evidencia de la actividad

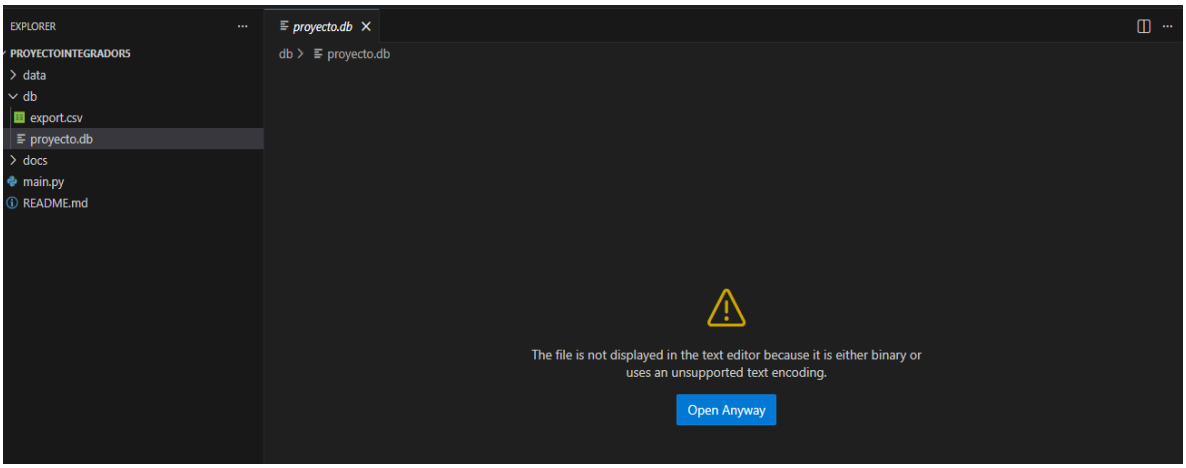
- ProyectoIntegrador5>data>Estadisticas_Riegos_Laborales_Positiva-Sep_2025.csv



- ProyectoIntegrador5>db>export.csv

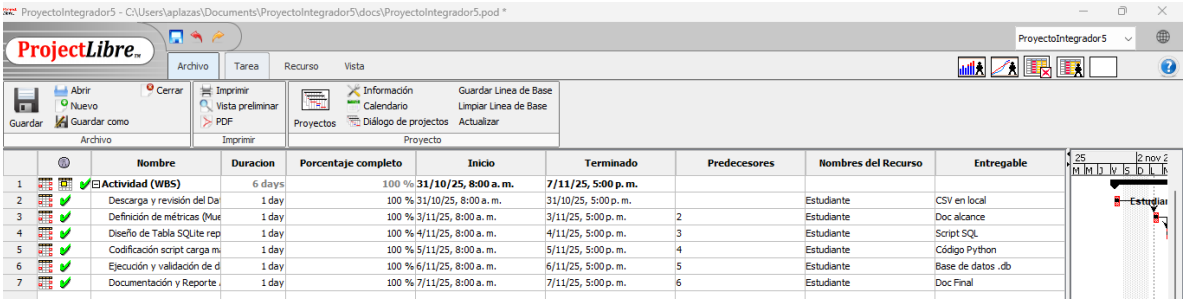


- ProyectoIntegrador5>db>proyecto.db

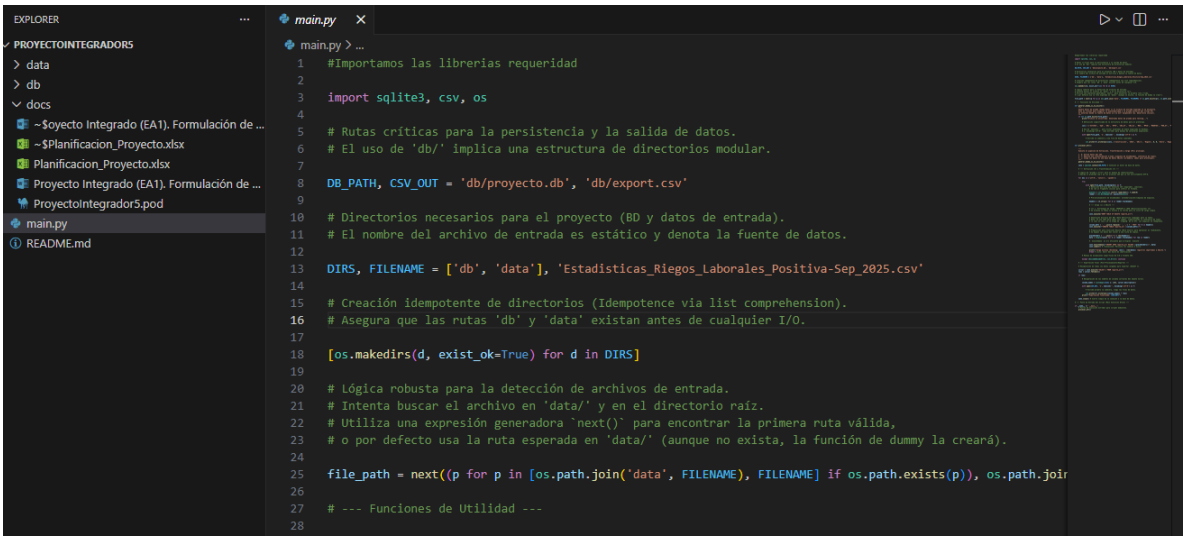


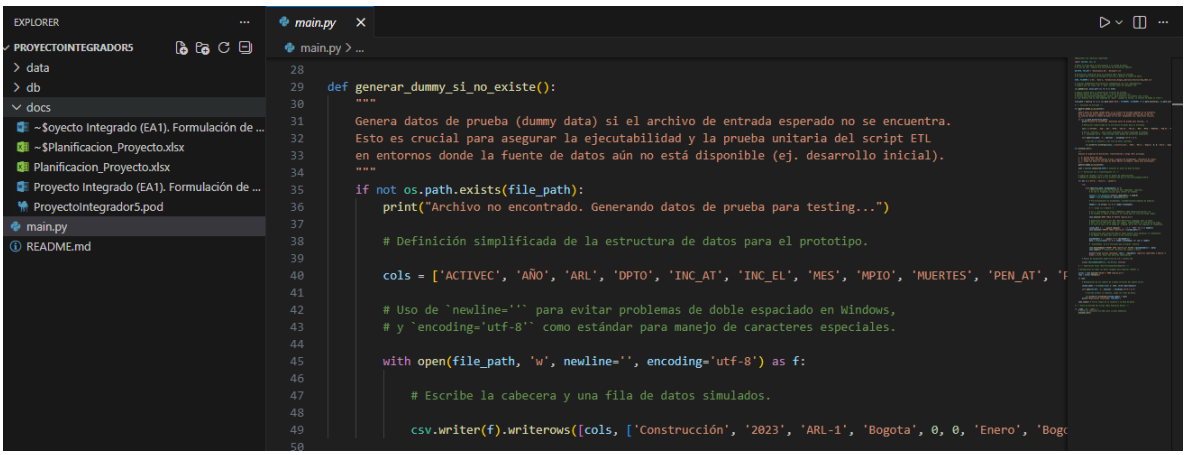
- ProyectoIntegrador5>docs>cronogramas

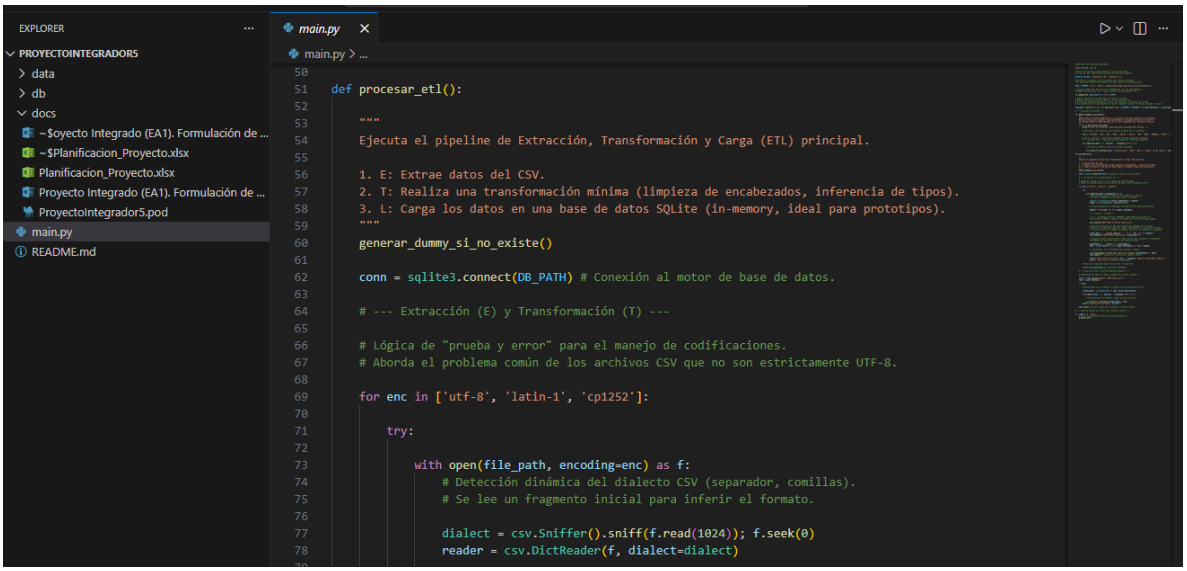
Nombre	Duración	Porcentaje Completo	Inicio	Fin	Predecesores	Nombre responsable	Entregable
Actividad (WBS)	**6 days**	**1.0**	**31/10/25, 8:00 a. m.**	**7/11/25, 5:00 p. m.**			
Descarga y revisión del Dataset ARL	1 day	1.0	31/10/25, 8:00 a. m.	31/10/25, 5:00 p. m.		Estudiante	CSV en local
Definición de métricas (Muertes vs Actividad)	1 day	1.0	3/11/25, 8:00 a. m.	3/11/25, 5:00 p. m.	2	Estudiante	Doc alcance
Diseño de Tabla SQLite reporte_arl	1 day	1.0	4/11/25, 8:00 a. m.	4/11/25, 5:00 p. m.	3	Estudiante	Script SQL
Codificación script carga main.py	1 day	1.0	5/11/25, 8:00 a. m.	5/11/25, 5:00 p. m.	4	Estudiante	Código Python
Ejecución y validación de datos	1 day	1.0	6/11/25, 8:00 a. m.	6/11/25, 5:00 p. m.	5	Estudiante	Base de datos .db
Documentación y Reporte APA	1 day	1.0	7/11/25, 8:00 a. m.	7/11/25, 5:00 p. m.	6	Estudiante	Doc Final

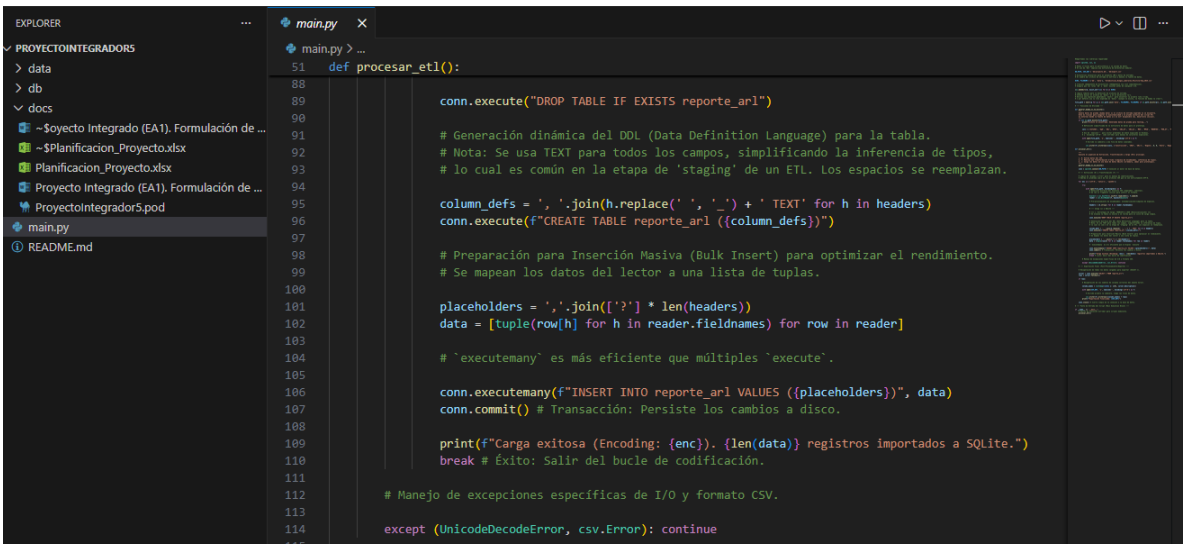


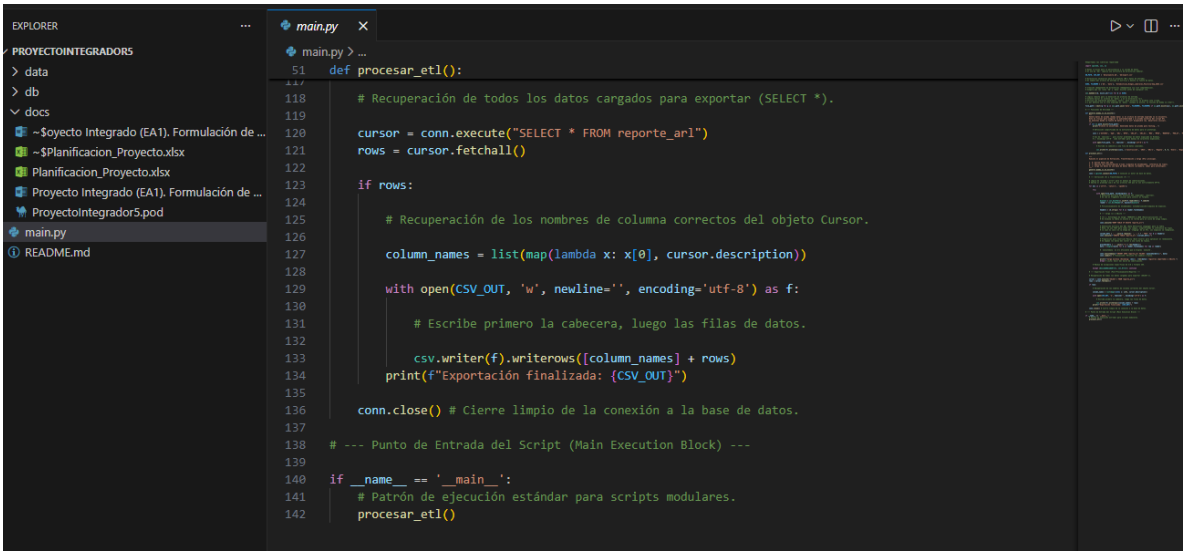
- ProyectoIntegrador5>main.py



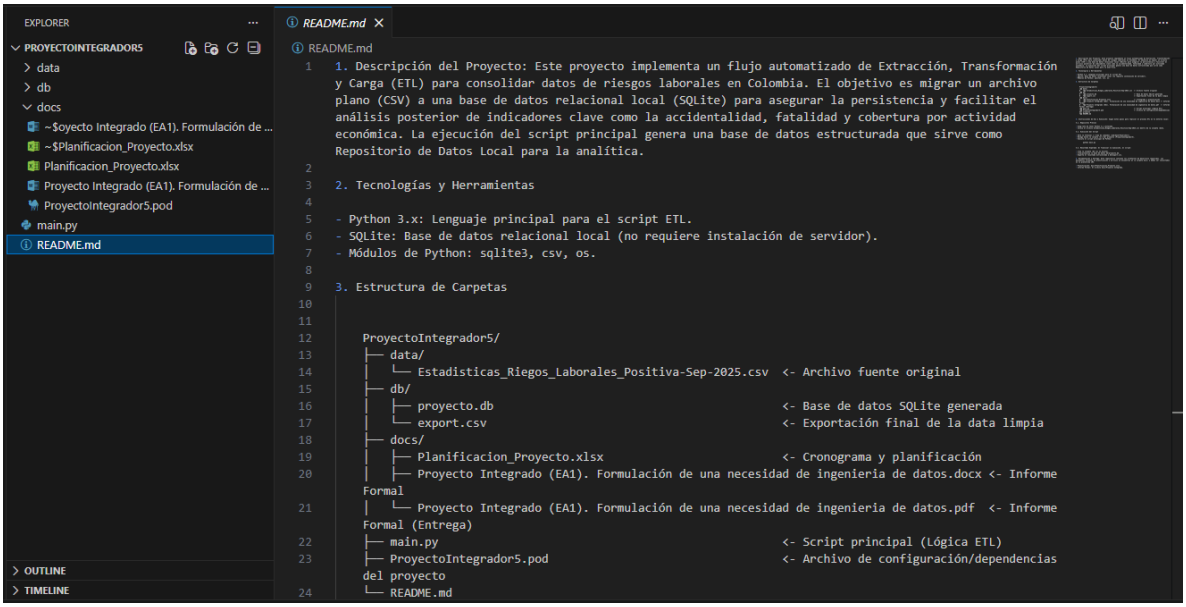


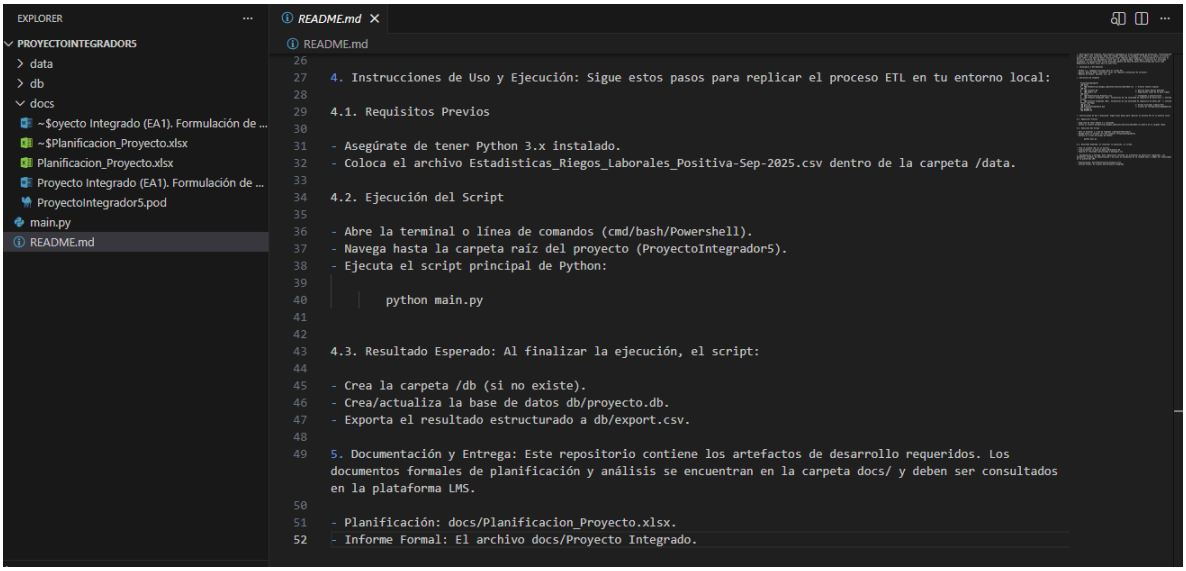

EXPLORER
PROYECTOINTEGRADOR5
data
db
docs
~$oyecto Integrado (EA1). Formulación de ...
~$Planificacion_Proyecto.xlsx
Planificacion_Proyecto.xlsx
Proyecto Integrado (EA1). Formulación de ...
ProyectoIntegrador5.pod
main.py
README.md
main.py
main.py > ...
def procesar_etl():
"""
Ejecuta el pipeline de Extracción, Transformación y Carga (ETL) principal.
1. E: Extrae datos del CSV.
2. T: Realiza una transformación mínima (limpieza de encabezados, inferencia de tipos).
3. L: Carga los datos en una base de datos SQLite (in-memory, ideal para prototipos).
"""
generar_dummy_si_no_existe()
conn = sqlite3.connect(DB_PATH) # Conexión al motor de base de datos.
# --- Extracción (E) y Transformación (T) ---
# Lógica de "prueba y error" para el manejo de codificaciones.
# Aborda el problema común de los archivos CSV que no son estrictamente UTF-8.
for enc in ['utf-8', 'latin-1', 'cp1252']:
try:
with open(file_path, encoding=enc) as f:
# Detección dinámica del dialecto CSV (separador, comillas).
# Se lee un fragmento inicial para inferir el formato.
dialect = csv.Sniffer().sniff(f.read(1024)); f.seek(0)
reader = csv.DictReader(f, dialect=dialect)


EXPLORER
PROYECTOINTEGRADOR5
data
db
docs
~$oyecto Integrado (EA1). Formulación de ...
~$Planificacion_Proyecto.xlsx
Planificacion_Proyecto.xlsx
Proyecto Integrado (EA1). Formulación de ...
ProyectoIntegrador5.pod
main.py
README.md
main.py
main.py > ...
def procesar_etl():
conn.execute("DROP TABLE IF EXISTS reporte_arl")
# Generación dinámica del DDL (Data Definition Language) para la tabla.
# Nota: Se usa TEXT para todos los campos, simplificando la inferencia de tipos,
# lo cual es común en la etapa de 'staging' de un ETL. Los espacios se reemplazan.
column_defs = ', '.join(h.replace(' ', '_') + ' TEXT' for h in headers)
conn.execute(f"CREATE TABLE reporte_arl ({column_defs})")
# Preparación para Inserción Masiva (Bulk Insert) para optimizar el rendimiento.
# Se mapean los datos del lector a una lista de tuplas.
placeholders = ','.join(['?'] * len(headers))
data = [tuple(row[h] for h in reader.fieldnames) for row in reader]
# `executemany` es más eficiente que múltiples `execute`.
conn.executemany(f"INSERT INTO reporte_arl VALUES ({placeholders})", data)
conn.commit() # Transacción: Persiste los cambios a disco.
print(f"Carga exitosa (Encoding: {enc}). {len(data)} registros importados a SQLite.")
break # Éxito: Salir del bucle de codificación.
Manejo de excepciones específicas de I/O y formato CSV.
except (UnicodeDecodeError, csv.Error): continue


EXPLORER
PROYECTOINTEGRADOR5
data
db
docs
~$oyecto Integrado (EA1). Formulación de ...
~$Planificacion_Proyecto.xlsx
Planificacion_Proyecto.xlsx
Proyecto Integrado (EA1). Formulación de ...
ProyectoIntegrador5.pod
main.py
README.md
main.py
main.py > ...
def procesar_etl():
# Recuperación de todos los datos cargados para exportar (SELECT *).
cursor = conn.execute("SELECT * FROM reporte_arl")
rows = cursor.fetchall()
if rows:
# Recuperación de los nombres de columna correctos del objeto Cursor.
column_names = list(map(lambda x: x[0], cursor.description))
with open(CSV_OUT, 'w', newline='', encoding='utf-8') as f:
# Escribe primero la cabecera, luego las filas de datos.
csv.writer(f).writerows([column_names] + rows)
print(f"Exportación finalizada: {CSV_OUT}")
conn.close() # Cierre limpio de la conexión a la base de datos.
# --- Punto de Entrada del Script (Main Execution Block) ---
if __name__ == '__main__':
# Patrón de ejecución estándar para scripts modulares.
procesar_etl()

- ProyectoIntegrador5>README.md





Conclusión

El presente proyecto ha logrado cumplir con éxito el Objetivo General de implementar un flujo de trabajo ETL robusto y reproducible para la consolidación de datos de siniestralidad laboral de las ARL. Mediante el uso de Python y SQLite, se ha demostrado la capacidad de migrar eficientemente datos desde un archivo plano con desafíos de formato (codificación) hacia un modelo relacional estructurado. La metodología empleada resultó en la generación de tres artefactos esenciales: el script ETL (main.py), la base de datos consolidada (db/proyecto.db), y el reporte normalizado de salida (db/export.csv), asegurando la integridad y la persistencia de la información.

Este repositorio de datos local se establece como un punto de partida fundamental para cualquier análisis descriptivo o predictivo futuro. La información ahora estructurada permite la inmediata identificación de patrones de riesgo críticos, particularmente en la correlación entre la actividad económica y la fatalidad laboral. En retrospectiva, la robustez en la fase de Extracción, al abordar la codificación y delimitadores, es un factor clave de éxito que garantiza la confiabilidad del proceso completo, preparando la data para la fase más crítica del proyecto: la visualización y la toma de decisiones informadas.

Referencias

Estadísticas riesgos laborales positiva 2025 | Datos abiertos Colombia. (2025, 22 octubre). https://www.datos.gov.co/Salud-y-Protecci-n-Social/Estad-sticas-Riesgos-Laborales-Positiva-2025/kwqa-xugj/about_data

AndresPlazas. (s. f.). *AndresPlazas19931504/proyectointegrador5*. GitHub. https://github.com/AndresPlazas19931504/proyectointegrador5

csv — Lectura y escritura de archivos CSV — documentación de Python - 3.9.24. (s. f.). https://docs.python.org/es/3.9/library/csv.html