Maestría en Inteligencia Artificial y Ciencia de Datos

García Muñoz, Marko D; Herrera Rivera, Jorge A; Torres Polanco, Cristhian A

Informe Técnico – Proyecto ETL y Análisis Exploratorio de Datos de Dengue

Repositorio GitHub: https://github.com/cristhianalextorres/etl_project.git

Descripción del Proyecto

El proyecto desarrolla un proceso ETL (Extract, Transform, Load) orientado a la gestión y análisis de información epidemiológica sobre casos de dengue en Colombia. El flujo se implementa en Python utilizando SQLite como motor de base de datos local, garantizando trazabilidad, modularidad y reproducibilidad.

El propósito general es automatizar la integración, limpieza y almacenamiento de datos provenientes de fuentes estructuradas, para posteriormente habilitar su análisis exploratorio y visualización en etapas analíticas posteriores.

La arquitectura del proyecto incluye los siguientes componentes principales:

- **Módulo de extracción (Extract):** lectura de los archivos de origen, validación de estructura y conteo de registros.
- **Módulo de transformación (Transform):** estandarización de campos, normalización de tipos de datos y depuración de registros inconsistentes o nulos.
- **Módulo de carga (Load):** inserción controlada de los datos validados en la base dbProject.db, dentro de la tabla datos_dengue.
- **Módulos complementarios:** manejo de logging, monitoreo de ejecución y verificación de métricas de calidad del proceso.

En conjunto, el flujo ETL asegura que los datos almacenados sean limpios, consistentes y auditables, permitiendo su posterior uso en análisis descriptivos y modelos de aprendizaje automático.

1. **Estructura del Proyecto**
   a. **consultas/**
      Contiene los scripts SQL necesarios para la creación de tablas y estructuras de la base de datos:
      datos_dengue.sql → define la estructura de la tabla principal con los campos del evento dengue.

```
CREATE TABLE IF NOT EXISTS datos_dengue (
    consecutive             INTEGER,
    cod_eve                 INTEGER,
    fec_not                 TEXT,
    semana                  INTEGER,
    ano                     INTEGER,
```

etl_monitor.sql → define la tabla encargada de registrar las métricas y monitoreo del proceso ETL.

```sql
CREATE TABLE IF NOT EXISTS etl_monitor (
                    id INTEGER PRIMARY KEY AUTOINCREMENT,
                    fecha TEXT,
                    registros_leidos INTEGER,
                    registros_validos INTEGER,
                    registros_descartados INTEGER,
                    duracion_segundos REAL,
                    error TEXT
);
```

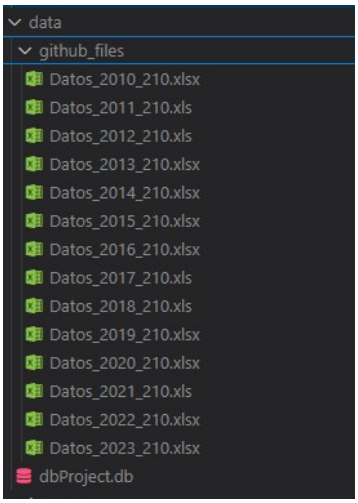
b. **data/**

La carpeta data la encuentra en el enlace porque es muy pesado. Este directorio donde se almacenan los archivos de datos y la base de datos generada:

https://drive.google.com/drive/folders/1yXFmbg-YAtsJxkfk29I1_0KQDHAcgtmZ?usp=sharing

Datos_Dengue.csv → archivo fuente con la información bruta del evento dengue.

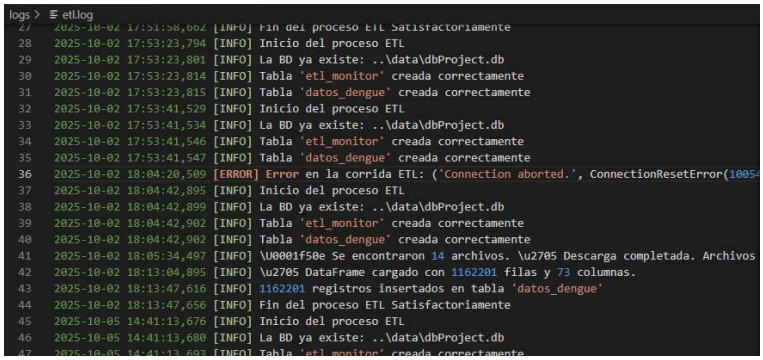
dbProject.db → base de datos SQLite resultante del proceso ETL, donde se cargan las tablas procesadas.



c. **logs/**

Carpeta destinada a registrar la trazabilidad del proceso:

etl.log → archivo de registro donde se almacenan eventos, errores y resultados de cada ejecución del ETL.

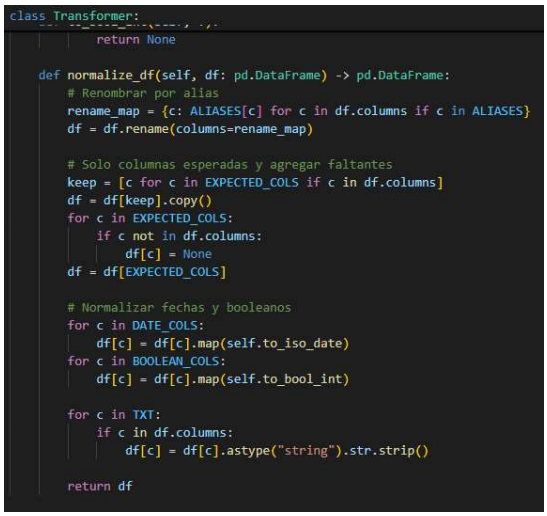


d. src/

Contiene el código fuente modular del proyecto. Cada archivo implementa una parte del flujo ETL:

extract.py → lee los datos desde el archivo CSV y valida su estructura.

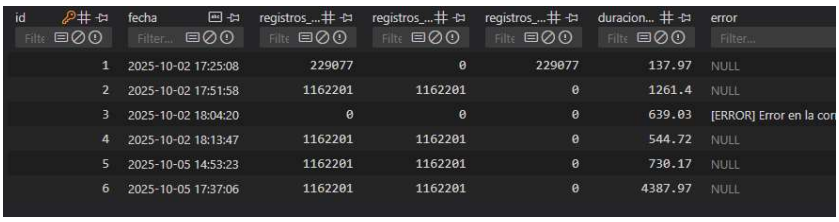
transform.py → limpia, estandariza y transforma los datos para su correcta carga.



load.py → inserta los datos transformados en la base de datos SQLite.

logger.py → gestiona el registro de eventos y errores durante la ejecución.

monitor.py → almacenas métricas de rendimiento y resultados en la tabla etl_monitor.



id	fecha	registros_...	registros_...	registros_...	duracion...	error
1	2025-10-02 17:25:08	229077	0	229077	137.97	NULL
2	2025-10-02 17:51:58	1162201	1162201	0	1261.4	NULL
3	2025-10-02 18:04:20	0	0	0	639.03	[ERROR] Error en la cor
4	2025-10-02 18:13:47	1162201	1162201	0	544.72	NULL
5	2025-10-05 14:53:23	1162201	1162201	0	730.17	NULL
6	2025-10-05 17:37:06	1162201	1162201	0	4387.97	NULL

schema.py → define la estructura de las tablas en la base de datos.

```python
schema.py > ...
EXPECTED_COLS = [
  "consecutive","cod_eve","fec_not","semana","ano","cod_pre","cod_sub","edad","uni_med",
  "nacionalidad","nombre_nacionalidad","sexo","cod_pais_o","cod_dpto_o","cod_mun_o","area",
  "ocupacion","tip_ss","cod_ase","per_etn","gru_pob","nom_grupo","estrato","gp_discapa",
  "gp_desplaz","gp_migrant","gp_carcela","gp_gestan","sem_ges","gp_indigen","gp_pobicfb",
  "gp_mad_com","gp_desmovi","gp_psiquia","gp_vic_vio","gp_otros","fuente","cod_pais_r",
  "cod_dpto_r","cod_mun_r","cod_dpto_n","cod_mun_n","fec_con","ini_sin","tip_cas","pac_hos",
  "fec_hos","con_fin","fec_def","ajuste","fecha_nto","cer_def","cbmte","fec_arc_xl","fec_aju",
  "fm_fuerza","fm_unidad","fm_grado","confirmados","consecutive_origen","va_sispro",
  "estado_final_de_caso","nom_est_f_caso","nom_upgd","pais_ocurrencia","nombre_evento",
  "departamento_ocurrencia","municipio_ocurrencia","pais_residencia","departamento_residencia",
  "municipio_residencia","departamento_notificacion","municipio_notificacion"
]

DATE_COLS = {
    "fec_not","fec_con","ini_sin","fec_hos","fec_def",
    "fecha_nto","fec_arc_xl","fec_aju"
}
```

main.py → orquesta la ejecución completa del proceso ETL (extract, transform, load).

	consecuti...	cod_eve	fec_not	semana	ano	cod_pre	cod_sub	edad
1	3076019	210	2010-05-27	21	2010	7689504656	1	
2	3076135	210	2010-06-16	24	2010	6827602278	0	
3	3076147	210	2010-12-14	49	2010	800103117	2	
4	3076156	210	2010-01-20	1	2010	6800101666	16	
5	3076182	210	2010-12-03	48	2010	7600103956	1	
6	3076186	210	2010-05-07	18	2010	507904806	7	
7	3076189	210	2010-09-07	35	2010	500102120	1	
8	3067261	210	2010-03-12	10	2010	8541000080	1	
9	3076331	210	2010-03-29	13	2010	500102178	7	
10	3067315	210	2010-03-29	12	2010	5400100603	1	
11	3067336	210	2010-07-23	29	2010	6800100431	1	
12	3067367	210	2010-04-14	15	2010	1343000056	0	
13	3067391	210	2010-02-10	6	2010	5400100603	1	
14	3067439	210	2010-03-28	12	2010	7300100956	1	
15	3067495	210	2010-06-24	25	2010	842100144	1	
	3067516	210	2010-02-19	7	2010	5000100635	1	

test.py → contiene pruebas básicas para verificar el correcto funcionamiento de los módulos.

2. EDA_Dengue.ipynb

Notebook de análisis exploratorio de datos (EDA). Permite revisar y visualizar los resultados del proceso ETL mediante gráficos y análisis descriptivos.

```python
plt.style.use("seaborn-v0_8")
sns.set_palette("pastel")

# ======================
# 1. Casos por Año
# ======================
conteo_ano = df["ano"].value_counts().sort_index()
ax = conteo_ano.plot(kind="bar", figsize=(8,4), edgecolor="black", color="skyblue")
plt.title("Casos de Dengue por Año")
plt.xlabel("Año")
plt.ylabel("Número de casos")
for i, v in enumerate(conteo_ano):
    ax.text(i, v + 50, str(v), ha="center", va="bottom", fontsize=9)
plt.show()
```

3. **.env**

   Archivo que almacena variables de entorno, como rutas o configuraciones de conexión, para mantener la seguridad y portabilidad del proyecto.

4. **.gitignore**

   Lista de archivos y carpetas que no deben ser rastreados por Git (por ejemplo, entornos, logs o bases de datos temporales).