EJERCICIO 10. PROCESO ETL

1. Energía Renovable Producida (Excel):
https://data.worldbank.org/indicator/EG.ELC.RNEW.ZS
2. Población mundial:
https://www.kaggle.com/datasets/iamsouravbanerjee/world-population-dataset

3. Global GHG Emissions:

https://www.kaggle.com/datasets/unitednations/international-greenhouse-gas-emissions

4. PIB per cápita:
https://www.rug.nl/ggdc/historicaldevelopment/maddison/releases/maddison-project-database-2023

El objetivo de esta práctica es desarrollar y ejecutar un proceso ETL (Extracción, Transformación y Carga) combinando datos de diferentes datasets.

Instrucciones para la práctica:

0. Organización del proyecto:

- Todo el código debe estar organizado dentro de una carpeta src/.
- Los datos de entrada deben estar en la carpeta data/.
- Los resultados del proceso ETL (datos en SQLite) se guardarán en la carpeta output/. La carpeta output/ se debe crear automáticamente si no existe cuando se ejecuta el script mediante Pathlib.

- El archivo de configuración config.yaml estará en el directorio raíz del proyecto.

Estructura del proyecto:

project-root/

├── data/ # Carpeta de entrada de los datasets de Kaggle

├── output/ # Carpeta donde se almacenarán los datos procesados

├── src/ # Código fuente del proyecto

├── config.yml # Archivo de configuración YAML

├── environment.yml # Configuración del entorno con Anaconda

1. Configuración (parsers.py):

- Todas las rutas de entrada y salida, así como otros parámetros, deben gestionarse a través del archivo config.yaml y una clase para parsear los datos.

2. Extracción (extract.py):

- Crear una clase Extractor abstracto que tenga como mínimo el método extract.
- Crear extractores CSV, Excel y SQLite a partir de la clase abstracta.

3. Transformación (transform.py):

Implementar las siguientes transformaciones para cada fuente de datos:

1. **Datos de Energía Renovable**:
    - **Trasponer los datos temporales**.
    - **Filtrar desde el año 1990 al 2014**
    - **Calcular la electricidad promedio y la electricidad promedio per cápita producida por cada país.**
2. **Global Greenhouse Gas Emissions**:
    - **Calcular emisiones promedio y emisiones promedio per cápita de cada país**.
3. **PIB Mundial**:
    - **Filtrar por el mismo rango de fechas**
    - **Calcular PIB y PIB per cápita promedio de cada país**

4. Carga (load.py):

- Carga de datos en base de datos SQLite
- Carga de datos en CSV

5. ETL (etl_process.py):

- Leer
- Transformar
- Combinar los datos de las 3 tablas utilizando los códigos de los países
- Agrupar por regiones (Western Europe, East Asia, etc.) y calcular la mediana y el promedio.

- Guardar en SQLite

6. Comprobación (check.py):

- Leer los datos de la base de datos SQLite
- Guardar en CSV los datos combinados

BONUS:

- Crear pipelines de pandas para todo el proceso de transformación de cada transformador

NOTA IMPORTANTE. Creación de la carpeta output/:

- Si la carpeta output/ no existe, el código deberá crearla automáticamente utilizando la librería pathlib.