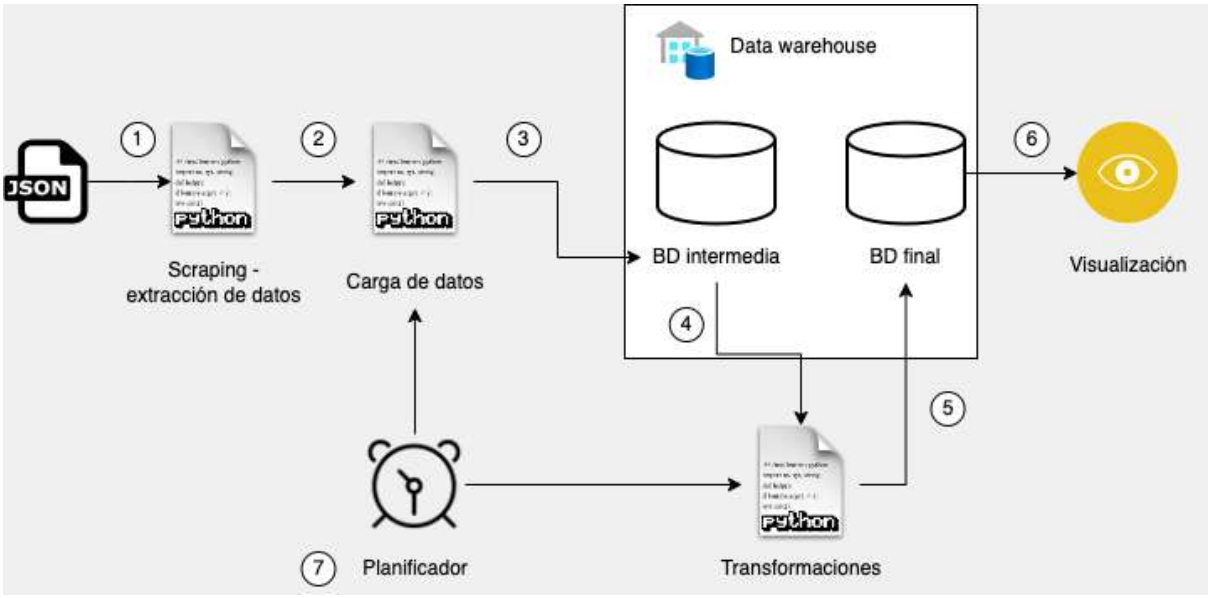
Traslatio - Data Pipeline



Descripción general:

Hacer una tubería de datos (data pipeline) sencilla en 2 versiones:

1. Versión simple con pocas herramientas y en local
2. Versión con herramientas avanzadas (docker o en la nube)

Objetivo:

Poner en práctica conceptos aprendidos del libro Fundamentals of Data Engineering. El objetivo es crear un data pipeline para hacer un proceso ETL. La idea es hacerlo de una forma sencilla y sin herramientas y luego en una segunda fase es elegir y usar herramientas de ingeniería de datos para comparar sus ventajas.

Objetivos específicos:

1. Extraer datos de un archivo JSON o CSV
2. Cargar datos del archivo a una base de datos (BD) intermedia.
3. Transformar los datos para que tengan sentido en una BD final.
4. Aprender a escoger herramientas para extracción, carga y transformación de los datos.
5. Aprender a escoger herramientas de visualización de datos
6. Aprender a escoger herramientas para la planificación de tareas repetitivas en el tiempo.

Conocimientos previos:

- Conocimiento básico en programación con Python.
- Opcional:
    - Conocimientos básico en SQL
    - Conocimiento básico en Docker
    - Conocimiento básico en Linux

Dificultad del proyecto:

El proyecto empieza con una dificultad principiante para luego pasar a una dificultad intermedia al incluir tecnologías de ingenieros de datos y tal vez tecnologías cloud.

Referencia:

- Proyecto Garantias.io

Versión 1

1. Crear un ambiente virtual con tu manejador de paquetes preferido.
2. Instalar paquetes de requirements.txt
3. Usaremos sqlite para las tablas de la base de datos
4. Para el ORM usaremos SQLAlchemy

Sesión de control:
Cada 15 días

Extracción:

- Extraer la cantidad de visitas de cada sitio y agregarlo al archivo JSON con el campo "**view_count**"
    - Recomendaciones de paquetes de Python:
        - requests
        - beautifulsoup
        - lxml (soporta XPATH)

Carga de datos:

- Solo carga de datos en bruto a la base de datos intermedia

Transformaciones

- Añadir deltas para saber cuánto tiempo ha pasado por cada noticia
    - Cambiar el nombre de la columna "date" por "**created date**"
    - Crear una nueva columna "**current date**"
    - *Crear una nueva columna "**delta date**" con la diferencia en formato epoch (columna tipo BIGINTEGER en la base de datos)

- Limpiar títulos
    - Quitar los saltos de línea
    - Quitar espacios

  * Verificar en la práctica

Análisis exploratorio de los datos

1. ¿Hay correlación entre el tiempo delta y el conteo de vistas?
    a. Esto para saber si entre más antigua la noticia, había tenido más visualizaciones.
2. ¿Cuáles son los tipos de datos de las columnas?
3. ¿Qué tipos de distribución de las columnas?
4. ¿Existen datos nulos o None?
5. ¿Existen datos duplicados?

Información a presentar

- Ranking de noticias más populares
- Usaremos **Matplotlib** por simplicidad