PIPELINE OPENWEATHERMAP API ISRAEL FUENTE SAAVEDRA

El Data Pipeline es un flujo ETL que tiene como propósito obtener datos meteorológicos de una API, transformar (limpieza) y cargarlos en una base de datos SQLite

1. Extracción de Datos

* Descripción: La extracción consiste en extraer la información de tres distintas ciudades (Nueva York, México, Londres) mediante una API de OpenWeatherMap.
* Proceso:
  1. Leer Configuración: Se lee un archivo de configuración (config.txt) para obtener las credenciales necesarias para la API, como la clave de API y el nombre de la base de datos.
  2. Realizar Solicitudes a la API: Para cada ciudad, se realiza una solicitud a la API de OpenWeatherMap utilizando la clave de API. La respuesta JSON incluye información meteorológica actual.
  3. Compilación de Datos: Los datos obtenidos son recopilados y convertidos en un DataFrame de pandas para su posterior procesamiento.

**2. Transformación de Datos**

* Descripción: El proceso de transformación convierte los datos extraídos en un formato adecuado para su almacenamiento y análisis.
* Proceso:
  1. Limpieza de Datos: Se limpia el DataFrame para rellenar valores faltantes en columnas numéricas con ceros. Esto asegura que los datos sean consistentes y no contengan valores nulos que puedan afectar el análisis.

**3. Carga de Datos**

* Descripción: La fase de carga inserta o actualiza los datos transformados en una base de datos SQLite.
* Proceso:
  1. Conexión a SQLite: Se establece una conexión con la base de datos SQLite utilizando el nombre de la base de datos especificado en el archivo de configuración.
  2. Creación de Tablas: Se crean las tablas necesarias si no existen. Estas tablas incluyen city\_catalog, weather, weather\_details, y weather\_summary.
  3. Inserción/Actualización de Datos:
     + Catalogar Ciudades: Se inserta o actualiza la información de las ciudades en la tabla city\_catalog. Esto asegura que cada ciudad esté registrada en la base de datos.
     + Actualizar Datos Meteorológicos: Se insertan o actualizan los datos de clima en las tablas weather, weather\_details, y weather\_summary. Las operaciones INSERT OR IGNORE y ON CONFLICT funcionan para evitar duplicados en los registros.

**4. Automatización con Dagster**

* Descripción: El pipeline ETL está automatizado utilizando Dagster, una plataforma de orquestación de datos.
* Proceso:
  1. Definición de Operaciones: Cada fase del pipeline (extracción, transformación y carga) está definida como una operación (@op) en Dagster.
  2. Definición del Trabajo: Se define un trabajo (@job) en Dagster que coordina la ejecución de las operaciones en el orden correcto.
  3. Ejecución y Monitoreo: Dagster gestiona la ejecución del pipeline y proporciona interfaces para monitorear el estado del pipeline y gestionar sus ejecuciones.

MODELO ENTIDAD-RELACION

A diagram of a weather forecast

Description automatically generated

El modelo entidad-relacion esta permitido para gestionar informacion meteorológica y de las ciudades. Las tablas han sido estructuradas para garantizar una correcta normalización de los datos y permite una construcción de reportes sencilla y precisa.

Relaciones entre tablas

Las relaciones entre las tablas se establecen a través de la llave foránea city\_id, que permite unir la información meteorológica básica, los detalles adicionales y los resúmenes con el catálogo de ciudades

Beneficios del Diseño

Organización Eficiente: Al dividir la información en tablas especializadas, se facilita la gestión y el acceso a los datos específicos.

Facilidad de Reportes: La estructura de llaves primarias y foráneas permite generar reportes complejos y precisos mediante consultas SQL que unen múltiples tablas.

Flexibilidad y Escalabilidad: La normalización de datos permite la expansión y actualización del modelo sin afectar la integridad de la información existente.

Para implementar soporte de streaming y procesar un mayor volumen de datos, se requeriría la integración de herramientas más robustas como Apache Spark. Esta tecnología permitiría extraer y transformar la información de manera eficiente, ya sea en lotes o en flujos de datos distribuidos, sin comprometer el rendimiento del sistema.