Project Documentation: Workshop 2 - ETL Process Using Apache Airflow

Author: Isabella Pérez Caviedes – 2230603

Date: 10/04/2025

Project: Workshop 2 – ETL Process Using Airflow

Repository: https://github.com/isabellaperezcav/Workshop_02

Version: 1.0

This document presents the solution developed for "Workshop 2: ETL Process Using Airflow", a challenge designed to assess skills in building ETL pipelines using Apache Airflow. The implemented solution extracts data from three sources (Spotify CSV, Grammy Awards DB, and Last.fm API), performs transformations, merges them, stores the results in a PostgreSQL database and Google Drive as a CSV file, and generates visualizations in Power BI.

Introduction

"Workshop 2: ETL Process Using Airflow" is a hands-on exercise focused on creating an ETL (Extract, Transform, Load) pipeline using Apache Airflow. The main objective is to demonstrate competencies in managing data from multiple sources (API, CSV, and database), transforming, integrating, and storing them, as well as generating useful visualizations from the processed data. This project uses datasets related to the music industry (Spotify Tracks, Grammy Awards, and Last.fm data) to analyze trends and popularity of genres and artists.

The scope includes:

- Extraction of data from three different sources
- Transformation and merging of data
- Loading results into a database and Google Drive

- Dashboard creation with visualizations

Project Requirements

Data Sources:

- **Spotify Tracks Dataset (CSV):** Information on songs with genres and audio features
- **Grammy Awards Dataset (DB):** Grammy Awards from 1958 to 2019
- **Last.fm API:** Additional data on artists and songs

Technologies:

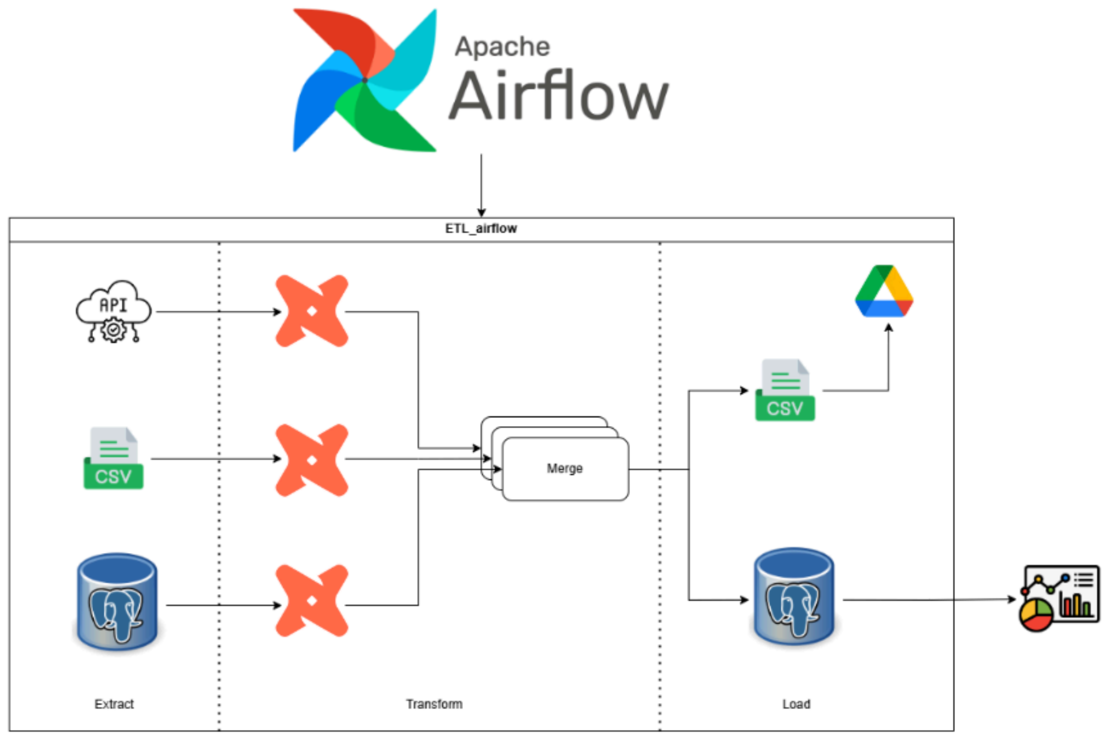
- Python
- Jupyter Notebook
- Apache Airflow
- Database (PostgreSQL)
- CSV
- Google Drive
- Visualization Tool (Power BI)

Solution Architecture

The solution follows a classic ETL architecture, implemented in Apache Airflow as a DAG (Directed Acyclic Graph). The general flow is as follows:

- **Extraction:**
  - Spotify: Reading from CSV file
  - Grammy: Initial load into PostgreSQL and subsequent read
  - Last.fm: API query
- **Transformation:**
  - Data cleaning, normalization, and enrichment

- **Load:**
  - Storage in PostgreSQL and export to Google Drive
- **Visualization:**
  - Power BI connected to PostgreSQL to generate reports



Pipeline

Implementation

Environment Setup

Requirements:

- Python 3.9+
- Apache Airflow 2.9+
- PostgreSQL
- Python libraries: `pandas`, `requests`, `psycopg2`, `google-auth`, `google-drive-api`

- **Installation:**

  ```
  pip install -r requirements.txt
  airflow db init
  airflow webserver --port 8080
  airflow standalone
  ```

Data Sources

- **Spotify Tracks Dataset:**

  CSV file (`spotify_dataset.csv`) with columns such as `track_id`, `artists`, `track_genre`, `popularity`, etc.

  Size: 1,140,000 rows x 21 columns

- **Grammy Awards Dataset:**

  Historical data of nominations and awards

  Loaded into PostgreSQL

- **Last.fm API:**

  Dynamic queries to access extended information about artists, albums, and listening trends

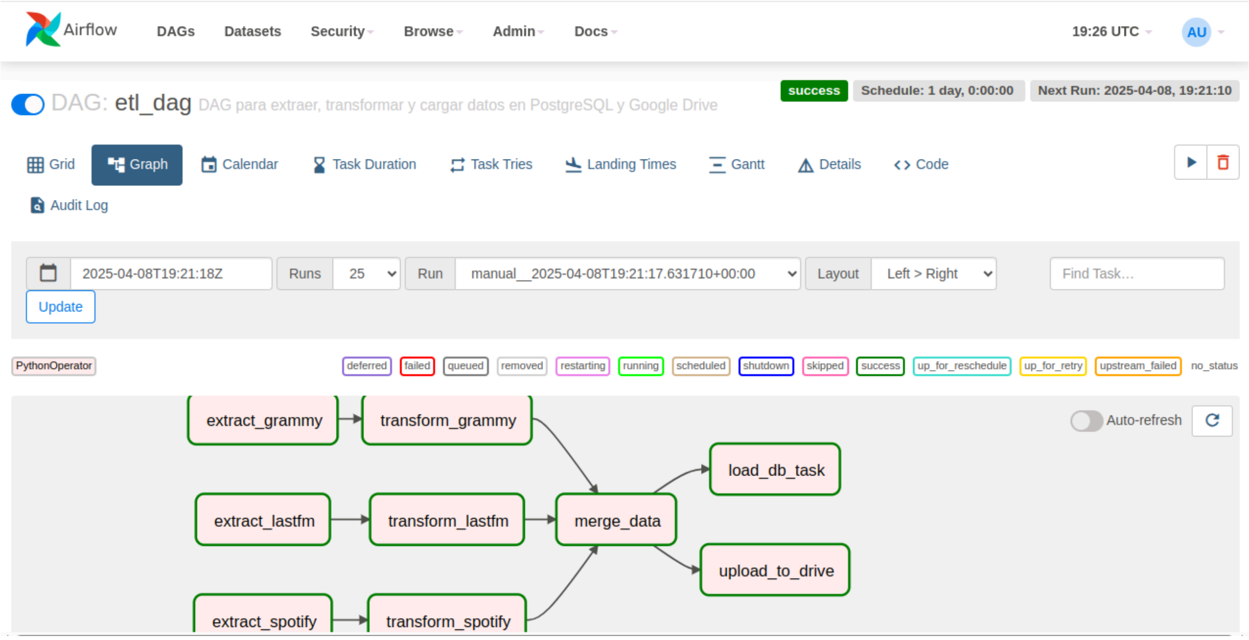
---

DAG Design in Airflow

The DAG (`etl_dag`) is structured with the following tasks:

- `extract_spotify`: Reads Spotify CSV
- `transform_spotify`: Cleans and normalizes data
- `extract_grammy`: Reads Grammy data from PostgreSQL
- `transform_grammy`: Processes Grammy data
- `extract_lastfm`: Queries the Last.fm API
- `transform_lastfm`: Normalizes API data
- `merge_data`: Merges the three datasets

- `load_db_task`: Stores data in PostgreSQL
- `upload_to_drive`: Exports result to Google Drive as CSV


DAG: etl_dag DAG para extraer, transformar y cargar datos en PostgreSQL y Google Drive

success Schedule: 1 day, 0:00:00 Next Run: 2025-04-08, 19:21:10

2025-04-08T19:21:18Z Runs 25 Run manual__2025-04-08T19:21:17.631710+00:00 Layout Left > Right

Airflow DAG interface

Transformations Performed

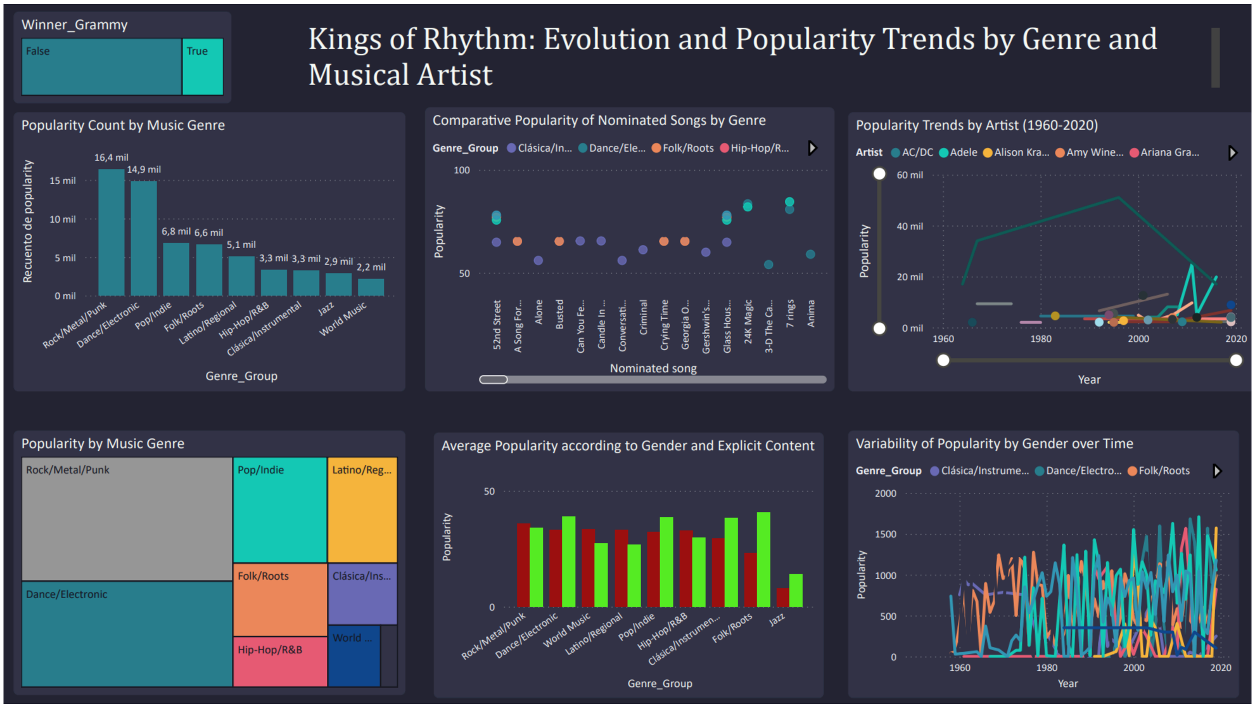
- **Spotify:** Duplicate removal, normalization of artist names (including column name)
- **Grammy:** Date conversion, filtering relevant nominees
- **Last.fm:** Extraction of popularity and trends from the API
- **Merge:** Joined by `artist` and `track_id` (when applicable), merged data from Spotify, Grammy, and Last.fm

Storage

- **PostgreSQL:** Table `merged_music_data` with columns such as `track_id`, `artist`, `genre`, `popularity`, `grammy_winner`
- **Google Drive:** CSV file uploaded to folder `Work_002_ETL` (Google Drive Folder)

Visualizations

A Power BI dashboard was created with the following visualizations:



Dashboard

Key Dashboard Findings

- **Dominant Genres:** "Rock/Metal/Punk" leads with 16.4K in popularity, followed by "Dance/Electronic" (14.9K) and "Pop/Indie" (6.8K)
- **Popularity by Genre:** "Rock/Metal/Punk" and "Pop/Indie" have the highest averages, while "Classical/Instrumental" and "Latin/Regional" are lower
- **Impact of Nominations:** Pop/Indie dominates in popularity, followed by a few instances of Dance/Electronic and Hip-Hop/R&B.
- **Artist Trends:** AC/DC and Adele show significant fluctuations, with marked increases and declines, while Ariana Grande rises since 2010
- **Explicit Content:** "Hip-Hop/R&B" has the highest popularity with explicit content
- **Temporal Variability:** "Dance/Electronic" shows high variability with peaks in 2000 and 2020; "Classical/Instrumental" is the most stable genre in popularity

The variance within each genre shows the diversity of acceptance that songs can have even within the same group

- **Grammy Effect:** Grammy winners (`True`) like Adele show more sustained popularity than non-winners

Usage Instructions

1. **Clone the Repository:**

   ```
   git clone <https://github.com/isabellaperezcav/Workshop_02.git>
   cd Workshop_02
   ```

2. **Configure Environment Variables:**

   - Create a `.env` file with PostgreSQL, Google Drive, and Last.fm API credentials.
   - Example:

     ```
     DB_HOST=localhost
     DB_NAME=music_db
     DB_USER=user
     DB_PASS=password
     GOOGLE_DRIVE_CREDENTIALS=/path/to/credentials.json  ## Provided by Google
     LASTFM_API_KEY=your_api_key
     ```

3. **Run the DAG:**

   - Start Airflow and activate the DAG `etl_dag` from the web interface (http://localhost:8080)

4. **Connect Power BI:**

   - Use the PostgreSQL connector to access the `merged_music_data` table once the Airflow process completes

Results and Visualizations

The processed data shows clear trends in the popularity of music genres and awarded artists.

- Genres like Electronic and Rock dominate in popularity
- Artists with Grammy awards tend to show more consistency in popularity over time

Conclusions and Recommendations

The implemented solution meets all the workshop objectives, demonstrating efficient handling of multiple data sources and modern technologies such as Airflow and Power BI. For future iterations, it is recommended to:

- Optimize Last.fm API queries to reduce execution time
- Add more transformations to analyze correlations between awards and popularity

References

- Repositorio GitHub: https://github.com/isabellaperezcav/Workshop_02
- Google Drive Folder: https://drive.google.com/drive/folders/1ncY6d8R5kpfsZyqh0B-WFZdhlTe3mAGh?usp=sharing
- Spotify Dataset: https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset
- Grammy Dataset: https://www.kaggle.com/datasets/unanimad/grammy-awards