Workshop 02 / ETL

Kevin Artunduaga - 2216155

Introducción

Este taller es un ejercicio sobre cómo construir un pipeline ETL usando Apache Airflow, la idea es extraer información usando tres fuentes de datos diferentes (API, archivo csv, base de datos), luego hacer algunas transformaciones y fusionar los datos transformados para finalmente cargarlos en google. conduzca como un archivo CSV y almacene los datos en una base de datos. Como último paso, crea un tablero a partir de los datos almacenados en la BD para visualizar la información de la mejor manera que consideres. Algunas cosas a considerar: le pido que complete este desafío dentro del plazo acordado en nuestra conversación.

Extract

Para la extracción de los datos había que tener algo claro y es que el excel de los grammy awards había que subirse primeramente en una base de datos para ello hice uso de mysql y creé una tabla llamada grammy_awards.

```
mysql> use ETL
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> describe grammy_awards;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| id           | int          | NO   | PRI | NULL    | auto_increment |
| year         | int          | YES  |     | NULL    |                |
| title        | varchar(255) | YES  |     | NULL    |                |
| published_at | varchar(255) | YES  |     | NULL    |                |
| updated_at   | varchar(255) | YES  |     | NULL    |                |
| category     | varchar(255) | YES  |     | NULL    |                |
| nominee      | varchar(255) | YES  |     | NULL    |                |
| artist       | varchar(255) | YES  |     | NULL    |                |
| winner       | tinyint(1)   | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
9 rows in set (0.00 sec)
```

Y como se puede apreciar cree 9 columnas y cada una con su respectivo tipo de dato, lo único que me tocó hacer fue quitar la columna de img y workers porque estas columnas no iban a ser usadas para el proceso, img primeramente porque es información irrelevante de una dirección de una imagen y workers hace parte de los trabajadores que hacen parte de la canción.

Ya luego para meter los datos cree un código llamado database.py, que hace la debida conexión con la base de datos con sus credenciales respectivas para mantener la seguridad, en donde toma el nombre de la tabla y va ingresando cada uno de los datos respectivamente.

```
for _, row in df.iterrows():
        query = f"INSERT INTO {table_name} (year, title, published_at,
updated_at, category, nominee, artist, winner) VALUES (%s, %s, %s, %s, %s,
%s, %s, %s)"
        cursor.execute(query, (row["year"], row["title"],
row["published_at"], row["updated_at"], row["category"], row["nominee"],
row["artist"], row["winner"]))
```

Ahora bien para el csv de spotify habia que leerlo directamente desde el csv, para ello cree una funcion llamada read_csv

```
def read_csv():
    csv_spotify = "/home/vinke1302/Apache/spotify_dataset.csv"
    df_spotify = pd.read_csv(csv_spotify, delimiter=',')
    logging.info("Spotify CSV readed sucesfully")
```

Este csv dispone de las columnas separadas por , y lo ubique de manera que estuviera en el entorno que estaba trabajando.

De esta manera ya están ambos datasets extraídos mediante distintos métodos usados.

Transform

En el proceso de transformación de datos, primero se abordó el conjunto de datos de Spotify, contenido en un archivo CSV. Para comenzar, se realizó la lectura del archivo y se procedió a reconfigurar los valores de la columna "duration_ms" de milisegundos a minutos y segundos, resultando en la creación de una nueva columna denominada "duration_min_sec", lo que facilitó la comprensión de la duración de las canciones. Además, se aplicó una categorización a la columna "popularity", dividiendo las canciones en cuatro rangos de popularidad (0-25, 25-50, 50-75, 75-100), y se guardó esta información en la columna "popularity_categorize". Asimismo, se eliminaron las columnas "mode" y "Unnamed: 0", ya que no aportaban valor significativo al análisis.

Tambien la columna "valence" se dividió en cuatro categorías, desde "0.0-0.25" hasta "0.75-1.0", lo que permitió segmentar las canciones según su valencia emocional. Se manejo un proceso parecido para la columna "speechiness" se categorizó en tres grupos: "Music," "Mixed," y "Speech," para distinguir entre pistas predominantemente musicales y aquellas con componentes hablados. A su vez con la columna "danceability" se dividió en cuatro categorías desde "0.0-0.25" hasta "0.75-1.0", lo que facilitó la identificación de las canciones según su grado de idoneidad para el baile. Y para la columna "energy" se segmentó en cuatro

categorías desde "0.0-0.25" hasta "0.75-1.0", lo que permitió clasificar las canciones según su nivel de energía.

En cuanto a la transformación de la base de datos de los premios Grammy, almacenada en mysql, se llevaron a cabo procedimientos igualmente cruciales. En primer lugar, se efectuó la lectura de la base de datos, y se procedió a eliminar aquellas filas que presentaban valores nulos en la columna "nominee", con el propósito de garantizar la integridad y utilidad de los datos. También, se eliminaron las columnas "published_at" y "updated_at", dado que su información no era relevante para el análisis y hacian parte de fechas no lo suficientemente relevantes para el propósito que se quería. Como última modificación, se renombró la columna "category" como "grammy_category", lo que simplificó la interpretación de los datos.

Tras completar estas transformaciones, los conjuntos de datos de Spotify y los premios Grammy se unieron en un solo DataFrame, lo que permitió relacionar la información de ambas fuentes de manera coherente. Las columnas se organizaron de manera ordenada, y se gestionaron los valores nulos en la columna "winner". Este proceso se realizó de manera cuidadosa para asegurar que los datos estuvieran listos para su carga en la base de datos MySQL y su posterior exportación a un archivo CSV, con el nombre "music_data.csv". La colaboración de estas transformaciones permitió obtener información valiosa que facilita el análisis y comprensión de la música galardonada en los premios Grammy y sus características en el conjunto de datos de Spotify.

Load

El proceso de carga de datos, denominado "load", se realizó con el objetivo de tomar los datos fusionados obtenidos en la etapa de "merge" y almacenarlos en una base de datos MySQL para su posterior análisis y consulta. Para llevar a cabo este proceso, se utilizó una conexión activa a la base de datos MySQL, previamente configurada. Primero, se procedió a crear una tabla en la base de datos con la estructura adecuada para acomodar los datos fusionados, lo que implicó definir columnas con tipos de datos específicos para cada atributo. Una vez creada la tabla, se realizó la inserción de los datos, los cuales se habían transformado y organizado previamente.

```
create_table_query = f"""
            CREATE TABLE IF NOT EXISTS music (
                track_id VARCHAR(255),
                artists VARCHAR(600),
                album_name VARCHAR(255),
                track_name VARCHAR(600),
                popularity_categorize VARCHAR(10),
                duration_min_sec VARCHAR(10),
                explicit TINYINT(1),
                danceability_category VARCHAR(10),
```

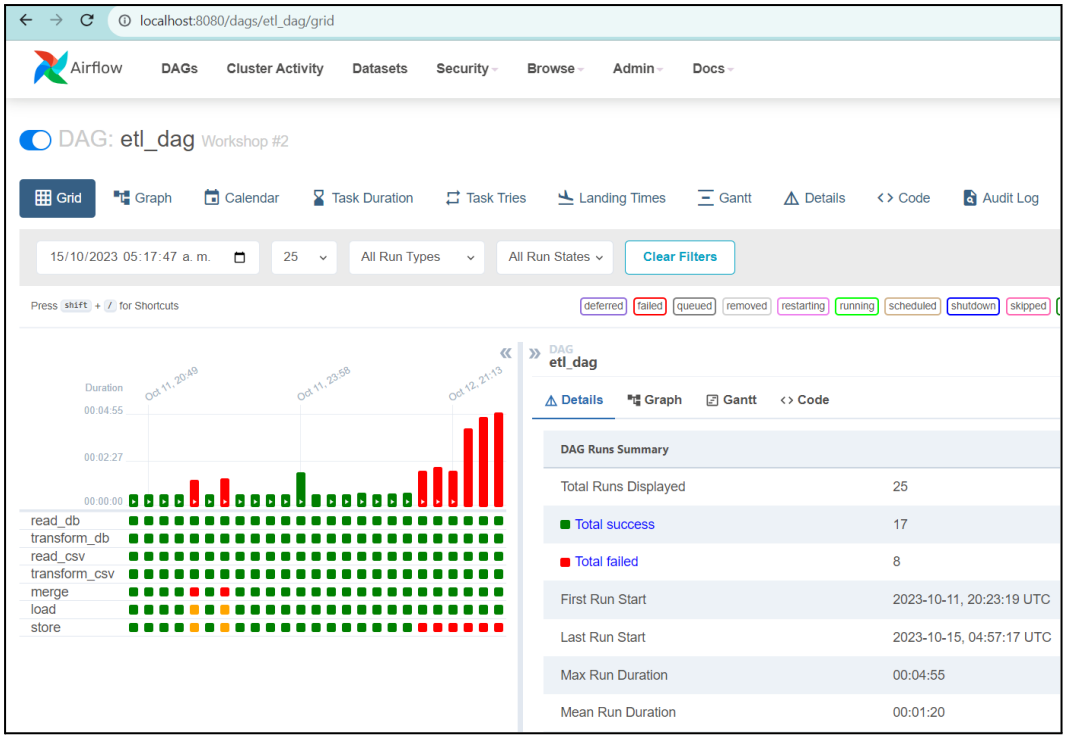
```
            energy_category VARCHAR(10),
            loudness FLOAT,
            speechiness_category VARCHAR(10),
            acousticness FLOAT,
            instrumentalness FLOAT,
            liveness FLOAT,
            valence_category VARCHAR(10),
            tempo FLOAT,
            time_signature INT,
            track_genre VARCHAR(255),
            grammy_category VARCHAR(255),
            nominee VARCHAR(255),
            winner TINYINT(1)
        );
    """
```

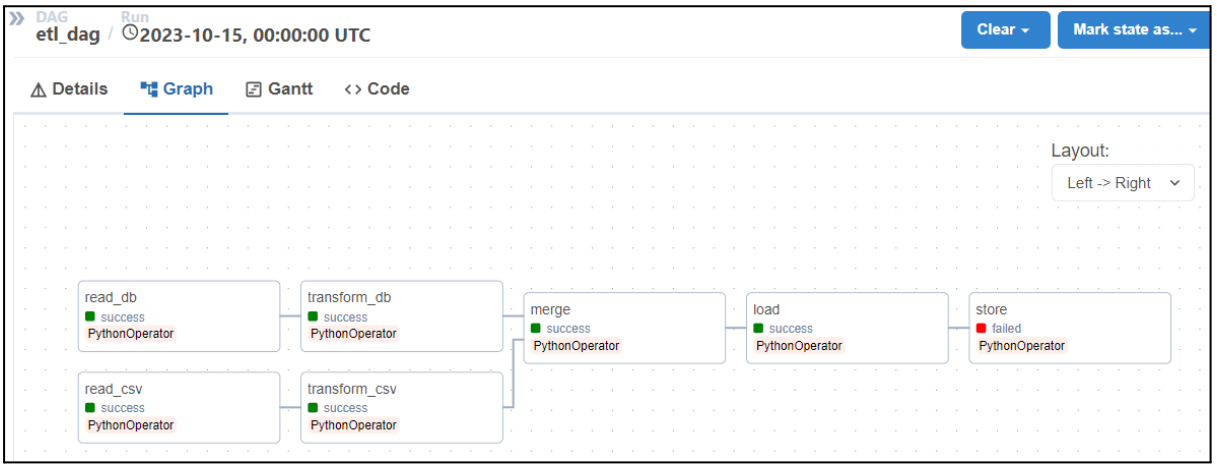
En la etapa de "merge", se unieron los datos del conjunto de Spotify y los datos de los premios Grammy, basándose en el nombre de la canción ("track_name") y el nombre de la cancion nominada a los premios Grammy ("nominee"). Esta fusión permitió combinar la información detallada de las canciones con los posibles reconocimientos obtenidos en los premios Grammy. En el proceso de "load", los datos resultantes de esta fusión se tomaron y se insertaron en la tabla recién creada en la base de datos.

Además, se realizaron ajustes adicionales, como la creación de una columna llamada "winner", que se llenó con valor 0 para indicar que inicialmente ninguna canción se consideraba ganadora de un premio Grammy, y la configuración de las categorías para los datos categóricos, como la popularidad de la canción, la duración, la valencia emocional, el contenido de habla, la idoneidad para el baile y el nivel de energía.

Finalmente, se generó un archivo CSV que contiene los datos almacenados en la base de datos MySQL. Este archivo es útil para futuros análisis y puede ser descargado y compartido fácilmente. En resumen, el proceso de "load" asegura que los datos estén listos y disponibles para su exploración y uso en análisis posteriores.

Airflow Pipeline

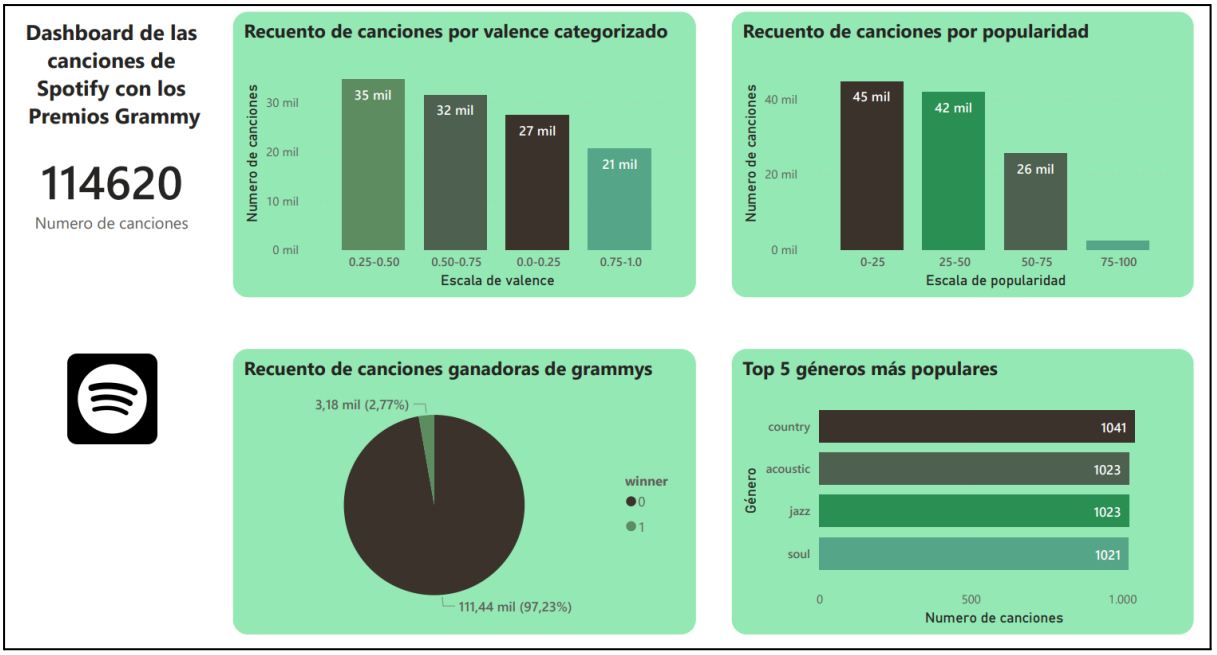




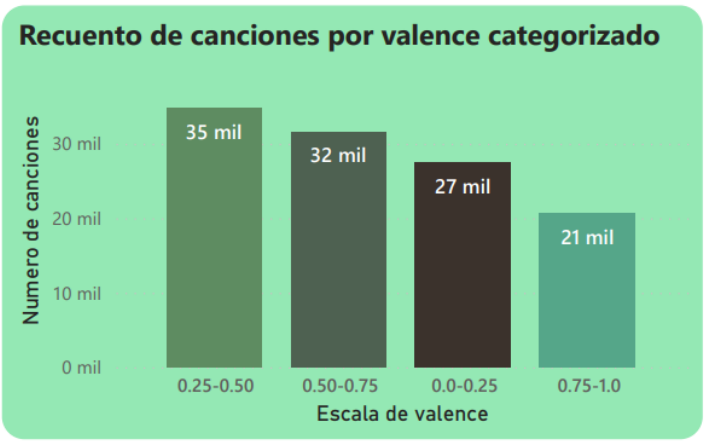
Para el airflow pipeline lamentablemente, no fue posible completar la función de "store" como se tenía previsto debido a dificultades técnicas imprevistas. A pesar de que las credenciales se generaron correctamente, durante la ejecución de la tarea, se produjo un error relacionado con la carga de estas credenciales al sistema. Este error, específicamente relacionado con la clave "_module", causó un problema en el proceso de autenticación con Google Drive.

A pesar de los esfuerzos por solucionar este problema, no se logró encontrar una solución. Se comprende la importancia de la tarea de "store" para asegurar que los datos estén disponibles en Google Drive para su posterior uso y análisis pero aunque se intentó solucionar no pudo ser posible.

Gráficos

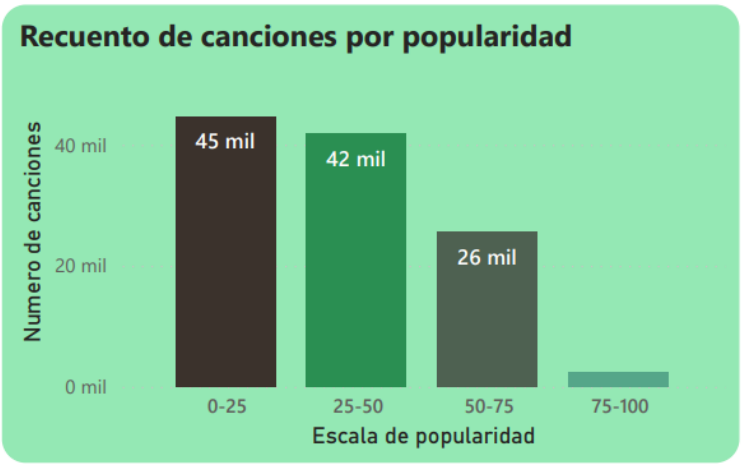


Gráfica 1: Recuento de canciones por valence categorizado



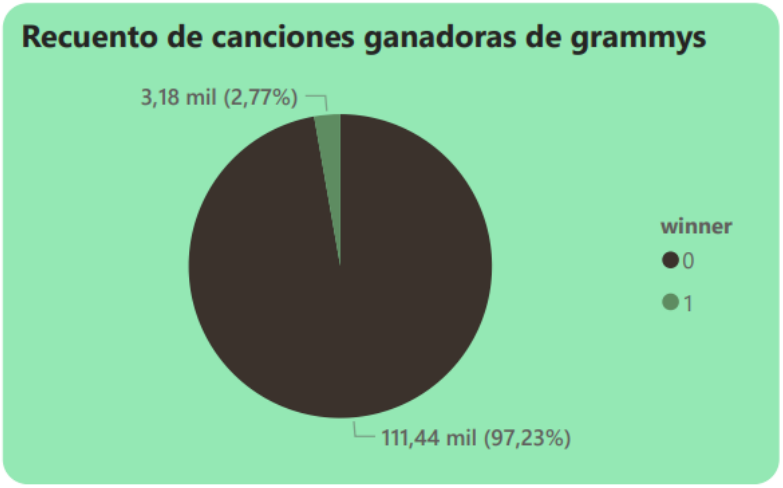
Esta gráfica muestra el número de canciones que se han publicado en Spotify en el último año, categorizadas según su valence, que es una medida de su positividad o negatividad. La gráfica muestra que la mayoría de las canciones tienen un valence positivo teóricamente neutro y que no es tan recurrente que las canciones son positivas, mientras que aproximadamente solo un 21 mil son completamente o altamente positivas

Gráfica 2: Recuento de canciones por popularidad



Esta gráfica muestra el número de canciones categorizadas según su popularidad. La gráfica muestra que la mayoría de las canciones publicadas tienen una popularidad baja o media , seguido de una popularidad alta que son unas muy pocas canciones que disponen de una calificación tan alta respecto a la popularidad.

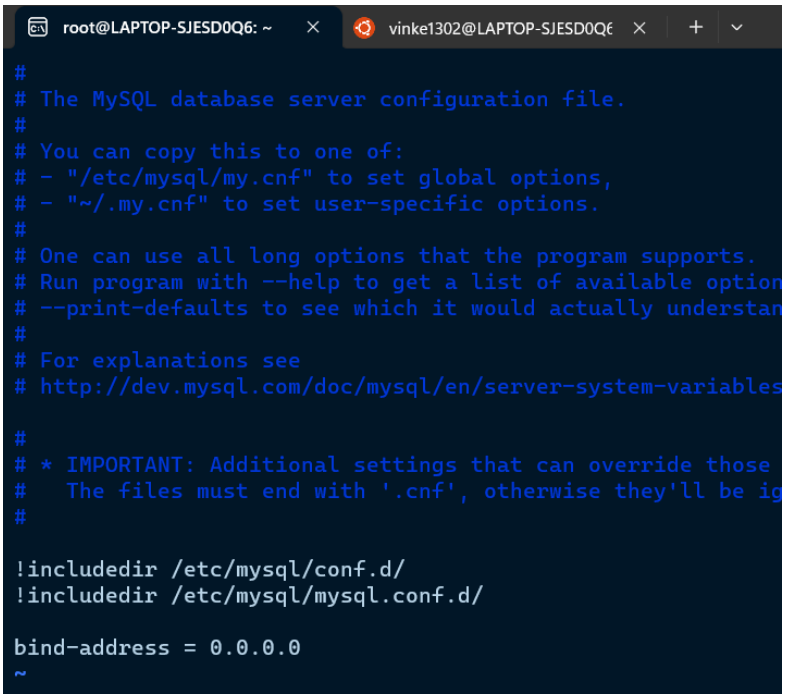
Gráfica 3: Dashboard de las canciones ganadoras de los Premios Grammy



Esta gráfica muestra un resumen de las canciones de Spotify que han sido nominadas o ganadoras de un Premio Grammy. La gráfica muestra que el 2,77% de las canciones han sido nominadas o ganadoras de un Premio Grammy mientras que el resto no han sido ganadoras lastimosamente.

Cabe recalcar que no tuve la oportunidad de poder conectar tampoco el Power BI a mysql porque al intentar cambiar las configuraciones de el localhost lo único que provocaba es que el servicio se cayera, así que para la conexión me tocó hacerlo mediante el csv final que es el que se llama music_data.csv

vim /etc/mysql/my.cnf



Error al iniciar mysql de esa manera: