

WORKSHOP 2

MARÍA FERNANDA TELLO VERGARA

2225338

JAVIER ALEJANDRO VERGARA

ETL

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE

OCTUBRE 04 2024

CONTEXT

We are going to extract information using different data sources (csv file, database), then make some transformations and merge the transformed data to finally upload it to google drive as a CSV file and store the data in a DB. As a last step, create a dashboard from the data stored in the DB to visualize the information in the best way you consider.

DESCRIPTION

We will use the Spotify dataset to read it in Python and Airflow, create some transformations and load it into a database, on the other hand we will use the Grammys dataset to load it into a database, then using Airflow we will read the data from the database, perform some transformations, merge with the Spotify dataset and load into the database.

TOOLS

- Python (Pandas, Matplotlib, SQLAlchemy dotenv).
- Postgress.
- Jupyter Notebook.
- Datasets.
- Docker
- Encryption of credentials using a .env file (Environment variables).
- Apache airflow.
- Google drive.
- Power BI.

STEP BY STEP

We check the installed Python version (3.12.3) and install Apache Airflow version 2.10.1.

```
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop$ python3 --version
Python 3.12.3
mari...@maria-fernanda-VirtualBox:~/Desktop$
```

```
mari...@maria-fernanda-VirtualBox:~/Desktop$ pip install "apache-airflow[celery]==2.10.1" --constraint "https://raw.githubusercontent.com/apache/airflow/constraints-2.10.1/constraints-3.8.txt"
```

We create the workshop2 directory, initialize a Git repository and configure a virtual environment in Python.

```
you can override this, at the front of breaking your Python code  
-system-packages.  
hint: See PEP 668 for the detailed specification.  
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop$ mkdir workshop2  
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop$ ls  
workshop2  
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop$ cd workshop2  
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ ls  
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ git init  
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ rm -rf venv  
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ python3 -m venv venv  
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ ls  
venv  
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$
```

We enter the virtual environment:

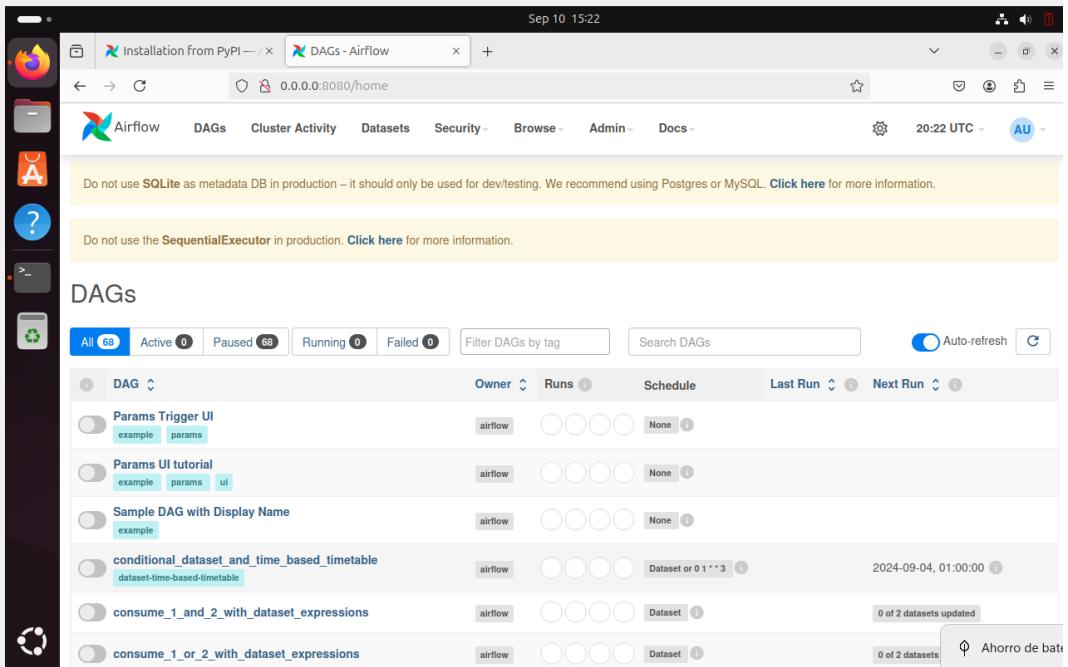
```
venv  
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ source venv/bin/activate  
(venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ █
```

We created the Dags directory, logged into it, and set the AIRFLOW_HOME environment variable to the current path of the Dags directory.

```
(venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ mkdir Dags
(venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ 
(venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ ls
Dags  venv
(venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ cd Dags
(venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ ls
(venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ export AIRFLOW_HOME=$(pwd)
(venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ echo $AIRFLOW_HOME
/home/maria-fernanda/Desktop/workshop2/Dags
(venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ █
```

We start Airflow in standalone mode after setting the AIRFLOW_HOME environment variable. The process includes initializing the database, creating an admin user, and configuring the Airflow web server.

We open the Airflow web interface in the browser, showing the list of available DAGs (workflows).

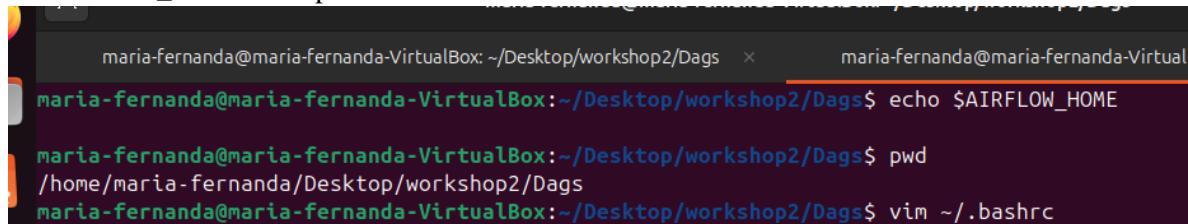


The screenshot shows the Airflow web interface at <http://0.0.0.0:8080/home>. The top navigation bar includes links for Airflow, DAGs, Cluster Activity, Datasets, Security, Browse, Admin, and Docs. It also shows the date (Sep 10 15:22), time (20:22 UTC), and user (AU). Two yellow warning boxes are present: one about not using SQLite as metadata DB and another about not using SequentialExecutor in production.

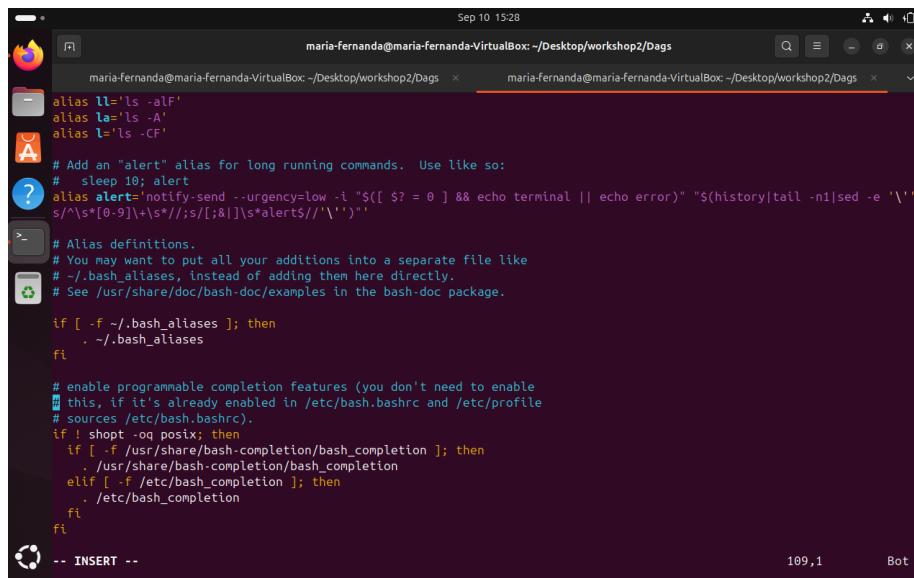
The main section is titled "DAGs" and displays a table of DAGs. The columns are: DAG, Owner, Runs, Schedule, Last Run, and Next Run. The table lists several DAGs:

- Params Trigger UI (Owner: airflow, Schedule: None)
- Params UI tutorial (Owner: airflow, Schedule: None)
- Sample DAG with Display Name (Owner: airflow, Schedule: None)
- conditional_dataset_and_time_based_timetable (Owner: airflow, Schedule: Dataset or 0 1 * * 3, Last Run: 2024-09-04, 01:00:00)
- consume_1_and_2_with_dataset_expressions (Owner: airflow, Schedule: Dataset, Status: 0 of 2 datasets updated)
- consume_1_or_2_with_dataset_expressions (Owner: airflow, Schedule: Dataset, Status: 0 of 2 datasets)

We check the AIRFLOW_HOME environment variable and its path with the echo \$AIRFLOW_HOME and pwd commands.



```
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ echo $AIRFLOW_HOME
/home/maria-fernanda/Desktop/workshop2/Dags
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ pwd
```



```
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ vim ~/.bashrc
```

The terminal shows the contents of the `~/.bashrc` file, which contains various aliases and functions. The `alias ll='ls -alF'` alias is explicitly mentioned.

```
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $(($? == 0) && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]+\s*\//;s/\s*[\&]\s*/\''")'"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
  . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc.
if ! shopt -o posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
```

Set the variable in the system to not be cleared.

```
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export AIRFLOW_HOME=/home/maria-fernanda/Desktop/workshop2/Dags
```

```
Maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ 
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ 
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ vim ~/.bashrc
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.name
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.name 'MafeTello'
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.name
MafeTello
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ 
```

Edited Git configurations to set the global username to MafeTello using the git config user.name command.

```
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ 
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ vim ~/.bashrc
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.name
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.name 'MafeTello'
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.name
MafeTello
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.email
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.email 'maria_fernanda.tello@uao.edu.co'
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.email
maria_fernanda.tello@uao.edu.co
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ 
```

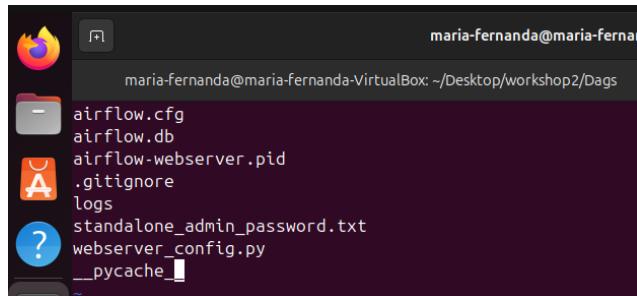
We configured the Git email with git config user.email, setting maria_fernanda.tello@uao.edu.co. Then, we added the remote repository to Git with git remote add origin, pointing to the repository on GitHub: <https://github.com/MafeTello/workshop2.git>. Finally, the connection to the remote repository was verified with git remote -v and a .gitignore file was created.

```
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ 
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ vim ~/.bashrc
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.name
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.name 'MafeTello'
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.name
MafeTello
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.email
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.email 'maria_fernanda.tello@uao.edu.co'
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git config user.email
maria_fernanda.tello@uao.edu.co
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git remote add origin https://github.com/MafeTello/workshop2.git
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git remote -v
origin https://github.com/MafeTello/workshop2.git (fetch)
origin https://github.com/MafeTello/workshop2.git (push)
```

```
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ VIM .gitignore
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ vim .gitignore
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$
```

(We did not upload these files to the repository)

We added the pycache.



Here we made the Git configurations, made the first commit and uploaded the changes to the remote repository on GitHub.

```
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ ls
Dags .git .gitignore venv
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ vim .gitignore
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ 
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    Dags/
nothing added to commit but untracked files present (use "git add" to track)
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ ls
Dags venv
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ cd Dags
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ ls
airflow.cfg airflow.db airflow-webserver.pid logs standalone_admin_password.txt webserver_config.py
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git add .
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git commit -m 'first commit'
[maintain (root-commit) 07bcbe1] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 Dags/.gitignore
marta-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git push -u origin main
Username for 'https://github.com': MafeTello
Password for 'https://MafeTello@github.com': 
```

We generate the tokens in git hub, and they are used as a password to do the git push.

Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

	Personal access tokens (classic)
Generate new token	Revoke all
Tokens you have generated that can be used to access the GitHub API .	
Make sure to copy your personal access token now. You won't be able to see it again!	
ghp_n819ha4k39MBC6rPNUcmB1U3VSK1NB1t3ev6 Delete	
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.	

The changes were pushed to the remote repository on GitHub, creating and linking the main branch with origin/main.

```
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/Dags$ git push -u origin main
Username for 'https://github.com': MafeTello
Password for 'https://MafeTello@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Writing objects: 100% (4/4), 261 bytes | 261.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MafeTello/workshop2.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
mari...@mari...-VirtualBox:~/Desktop/workshop2/Dags$
```

Install PostgreSQL on Ubuntu.

```
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ sudo apt install postgresql postgresql-contrib
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libpq5 libtypes-serialiser-perl postgresql-16 postgresql-client-16
  postgresql-client-common postgresql-common
Suggested packages:
  postgresql-doc postgresql-doc-16
The following NEW packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libpq5 libtypes-serialiser-perl postgresql postgresql-16
  postgresql-client-16 postgresql-client-common postgresql-common postgresql-contrib
0 upgraded, 11 newly installed, 0 to remove and 8 not upgraded.
Need to get 17.3 MB of archives.
After this operation, 50.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://co.archive.ubuntu.com/ubuntu noble amd64 libjson-perl all 4.10000-1 [81.9 kB]
Get:2 http://co.archive.ubuntu.com/ubuntu noble-updates/main amd64 postgresql-client-common all 257build1.1 [36.4 kB]
Get:3 http://co.archive.ubuntu.com/ubuntu noble-updates/main amd64 postgresql-common all 257build1.1 [161 kB]
Get:4 http://co.archive.ubuntu.com/ubuntu noble/main amd64 libcommon-sense-perl amd64 3.75-3build3 [20.4 kB]
Get:5 http://co.archive.ubuntu.com/ubuntu noble/main amd64 libtypes-serialiser-perl all 1.01-1 [11.6 kB]
```

We accessed the PostgreSQL console with the command sudo -i -u postgres.

```
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
mari...@mari...-VirtualBox:~/Desktop/workshop2$ sudo systemctl start postgresql.service
mari...@mari...-VirtualBox:~/Desktop/workshop2$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: enabled)
   Active: active (exited) since Tue 2024-09-10 16:21:22 -05; 50s ago
     Main PID: 18424 (code=exited, status=0/SUCCESS)
        CPU: 2ms

Sep 10 16:21:22 mari...-VirtualBox systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Sep 10 16:21:22 mari...-VirtualBox systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
mari...@mari...-VirtualBox:~/Desktop/workshop2$ sudo -i -u postgres
postgres@mari...-VirtualBox:~$ psql
psql (16.4 (Ubuntu 16.4-0ubuntu0.24.04.2))
Type "help" for help.

postgres=#
```

\q (salir)

PostgreSQL port settings and machine IP address are verified

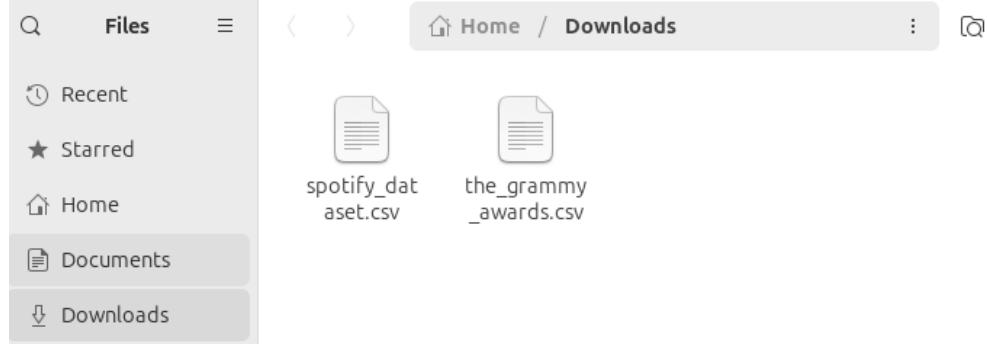
```
Sep 10 16:21:22 maria-fernanda-VirtualBox systemd[1]: Started PostgreSQL database service.
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ sudo -i -u postgres
postgres@maria-fernanda-VirtualBox:~$ psql
psql (16.4 (Ubuntu 16.4-0ubuntu0.24.04.2))
Type "help" for help.

postgres=# exit
postgres@maria-fernanda-VirtualBox:~$ psql
psql (16.4 (Ubuntu 16.4-0ubuntu0.24.04.2))
Type "help" for help.

postgres=# show port;
 port
 -----
 5432
(1 row)

postgres=# exit
postgres@maria-fernanda-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::a00:2ff:fe5eb prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:5f:f5:eb txqueuelen 1000 (Ethernet)
            RX packets 189138 bytes 261617114 (261.6 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 23875 bytes 4405403 (4.4 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

We download the files to upload them to the database.



A table named `grammy_awards` was created in PostgreSQL with several columns. The database relations were then listed using the `\d` command, confirming the existence of the table, and the detailed structure of the table was displayed using `\d grammy_awards`.

```
postgres=# CREATE TABLE grammy_awards (
    year INT,
    title VARCHAR(255),
    published_at TIMESTAMP,
    updated_at TIMESTAMP,
    category VARCHAR(255),
    nominee VARCHAR(255),
    artist VARCHAR(255),
    link TEXT,
    image TEXT,
    winner BOOLEAN
);
CREATE TABLE
postgres#
```

Schema	Name	Type	Owner
public	grammy_awards	table	postgres

```
postgres=# \d grammy_awards
              Table "public.grammy_awards"
   Column    |          Type          | Collation | Nullable | Default
---+-----+-----+-----+-----+-----+
year      | integer            |           |       |
title     | character varying(255) |           |       |
published_at | timestamp without time zone |           |       |
updated_at | timestamp without time zone |           |       |
category   | character varying(255) |           |       |
nominee    | character varying(255) |           |       |
artist     | character varying(255) |           |       |
link       | text                |           |       |
image      | text                |           |       |
winner     | boolean             |           |       |

```

We give read and write permissions so that any user can read and write the file.

```
maria-fernanda@maria-fernanda-VirtualBox:~... x maria-fernanda@maria-fernanda-VirtualBox:~... x maria-fernanda@maria-fernanda-VirtualBox:~...
mari...nanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ ls
Dags venv
mari...nanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ cd ..
mari...nanda@maria-fernanda-VirtualBox:~/Desktop$ ls
workshop2
mari...nanda@maria-fernanda-VirtualBox:~/Desktop$ cd ..
mari...nanda@maria-fernanda-VirtualBox:$ ls
airflow Desktop Documents Downloads Music Pictures Public snap Templates Videos
mari...nanda@maria-fernanda-VirtualBox:$ pwd
/home/maria-fernanda
mari...nanda@maria-fernanda-VirtualBox:$ cd Downloads/
mari...nanda@maria-fernanda-VirtualBox:~/Downloads$ ls
spotify_dataset.csv the_grammy_awards.csv
mari...nanda@maria-fernanda-VirtualBox:~/Downloads$ ll
total 21092
drwxr-xr-x 2 maria-fernanda maria-fernanda 4096 Sep 10 16:30 .
drwxr-x--- 16 maria-fernanda maria-fernanda 4096 Sep 10 16:00 ../
-rw-rw-r-- 1 maria-fernanda maria-fernanda 20118244 Sep 10 16:30 spotify_dataset.csv
-rw-rw-r-- 1 maria-fernanda maria-fernanda 1468421 Sep 10 16:29 the_grammy_awards.csv
mari...nanda@maria-fernanda-VirtualBox:~/Downloads$ chmod 777 the_grammy_awards.csv
mari...nanda@maria-fernanda-VirtualBox:~/Downloads$ ll
total 21092
drwxr-xr-x 2 maria-fernanda maria-fernanda 4096 Sep 10 16:30 .
drwxr-x--- 16 maria-fernanda maria-fernanda 4096 Sep 10 16:00 ../
-rw-rw-r-- 1 maria-fernanda maria-fernanda 20118244 Sep 10 16:30 spotify_dataset.csv
-rwxrwxrwx 1 maria-fernanda maria-fernanda 1468421 Sep 10 16:29 the_grammy_awards.csv*
mari...nanda@maria-fernanda-VirtualBox:~/Downloads$
```

We give permissions to home.

```
-rw-rw-r-- 1 maria-fernanda maria-fernanda 20118244 Sep 10 16:30 spotify_dataset.csv  
-rw-r--r-- 1 maria-fernanda maria-fernanda 1468421 Sep 10 16:29 the_grammy_awards.csv  
maria-fernanda@maria-fernanda-VirtualBox:~/Downloads$ sudo chmod 777 /home/maria-fernanda/  
[sudo] password for maria-fernanda:  
maria-fernanda@maria-fernanda-VirtualBox:~/Downloads$
```

```
postgres=# \COPY grammy_awards(year, title, published_at, updated_at, category, nominee, artist, link, image, winner)
FROM '/home/maria-fernanda/Downloads/the_grammy_awards.csv'
DELIMITER ','
CSV HEADER;
COPY 4810
postgres=#
```

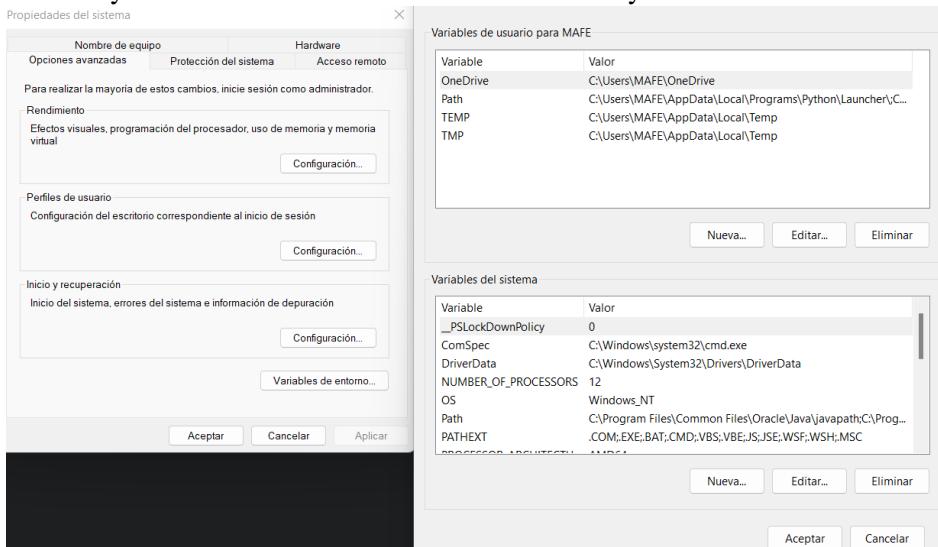
```
COPY 4810
postgres=# select * from grammy_awards limit 10
postgres# ;
postgres# |
```

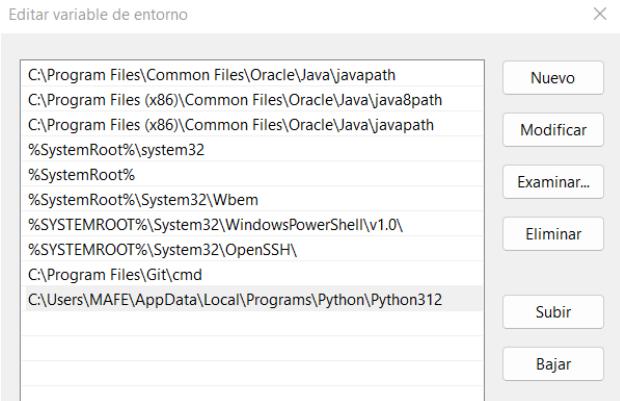
We use `git clone` to clone the workshop2 repository from GitHub to the local machine, download all the files from the remote repository to the local directory.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + ×

PS D:\Users\WAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP2> git clone https://github.com/WafeTello/workshop2
Cloning into 'workshop2'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
PS D:\Users\WAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP2> 
```

We verify that the environment variables are correctly added.





We verify that the Python version is running.

```

Símbolo del sistema - python
Microsoft Windows [Versión 10.0.22000.2538]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\MAFE>python
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

```

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

● (venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ ls
Dags Data eachtime.txt Notebooks venv

ERROR: No matching distribution found for ipykernel
○ (venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ pip install ipykernel
Collecting ipykernel
  Downloading ipykernel-6.29.5-py3-none-any.whl.metadata (6.3 kB)
Collecting comm>=0.1.1 (from ipykernel)
  Downloading comm-0.2.2-py3-none-any.whl.metadata (3.7 kB)
Collecting debugpy>=1.6.5 (from ipykernel)
  Downloading debugpy-1.8.6-cp312-cp312-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (1.1 kB)
Collecting ipython>=7.23.1 (from ipykernel)
  Downloading ipython-8.27.0-py3-none-any.whl.metadata (5.0 kB)
Collecting jupyter-client>=6.1.12 (from ipykernel)
  Downloading jupyter_client-8.6.3-py3-none-any.whl.metadata (8.3 kB)
Collecting jupyter-core!=5.0.*,>=4.12 (from ipykernel)
  Downloading jupyter_core-5.7.2-py3-none-any.whl.metadata (3.4 kB)

```

```

maria-fernanda@maria-fernanda-VirtualBox:~/Desktop$ 
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop$ ls
workshop2
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop$ cd workshop2
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ ls
Dags Data eachtime.txt Notebooks src venv
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ sudo -i -u postgres
[sudo] password for maria-fernanda:
postgres@maria-fernanda-VirtualBox:~$ psql
psql (16.4 (Ubuntu 16.4-0ubuntu0.24.04.2))
Type "help" for help.

postgres=# 

```

Due to some issues with Windows permissions, I opted to download Docker.

```
└── docker
    ├── pgadmin_data_mafe
    ├── postgres_data_mafe
    └── docker-compose.yml      u
        └── Notebooks
```

Here we can see that it is already installed.

```
marifernanda@marifernanda-VirtualBox:~/Desktop/workshop2$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:91fb4b041da273d5a3273b6d587d62d518300a6ad268b28628f74997b93171b2
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
```

```
db_mafe_test | 2024-10-01 02:06:47.769 UTC [46] LOG: background worker "logical replication launcher" (PID 52) exited
with exit code 1
db_mafe_test | 2024-10-01 02:06:47.770 UTC [47] LOG: shutting down
db_mafe_test | 2024-10-01 02:06:47.774 UTC [47] LOG: checkpoint starting: shutdown immediate
db_mafe_test | 2024-10-01 02:06:48.097 UTC [47] LOG: checkpoint complete: wrote 922 buffers (5.6%); 0 WAL file(s) added,
0 removed, 0 recycled; write=0.132 s, sync=0.174 s, total=0.327 s; sync files=301, longest=0.004 s, average=0.001 s;
distance=4255 kB, estimate=4255 kB; lsn=0/1912140, redo lsn=0/1912140
db_mafe_test | 2024-10-01 02:06:48.104 UTC [46] LOG: database system is shut down
db_mafe_test | done
db_mafe_test | server stopped
db_mafe_test | PostgreSQL init process complete; ready for start up.
db_mafe_test | 2024-10-01 02:06:48.213 UTC [1] LOG: starting PostgreSQL 16.4 (Debian 16.4-1.pgdg120+2) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
db_mafe_test | 2024-10-01 02:06:48.213 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
db_mafe_test | 2024-10-01 02:06:48.214 UTC [1] LOG: listening on IPv6 address "::", port 5432
db_mafe_test | 2024-10-01 02:06:48.221 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.PGSQL.5432"
db_mafe_test | 2024-10-01 02:06:48.237 UTC [62] LOG: database system was shut down at 2024-10-01 02:06:48 UTC
db_mafe_test | 2024-10-01 02:06:48.258 UTC [1] LOG: database system is ready to accept connections
^CGracefully stopping... (press Ctrl+C again to force)
[+] Stopping 1/1
  ✓ Container db_mafe_test Stopped
  canceled
marifernanda@marifernanda-VirtualBox:~/Desktop/workshop2/docker$ marifernanda@marifernanda-VirtualBox:~/Desktop/workshop2/docker$
```

I installed the python-decouple and python-dotenv packages, which are commonly used to handle environment variables in Python projects.

```
/home/marifernanda/Desktop/workshop2/venv/lib/python3.12/site-packages/decouple/*
Proceed (Y/n)? y
Successfully uninstalled decouple-0.0.7
(venv) marifernanda@marifernanda-VirtualBox:~/Desktop/workshop2$ pip install python-decouple python-dotenv
Collecting python-decouple
  Downloading python_decouple-3.8-py3-none-any.whl.metadata (14 kB)
Collecting python-dotenv
  Downloading python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
  Downloading python_decouple-3.8-py3-none-any.whl (9.9 kB)
  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Installing collected packages: python-decouple, python-dotenv
Successfully installed python-decouple-3.8 python-dotenv-1.0.1
(venv) marifernanda@marifernanda-VirtualBox:~/Desktop/workshop2$
```

Here we can see that the container is already running

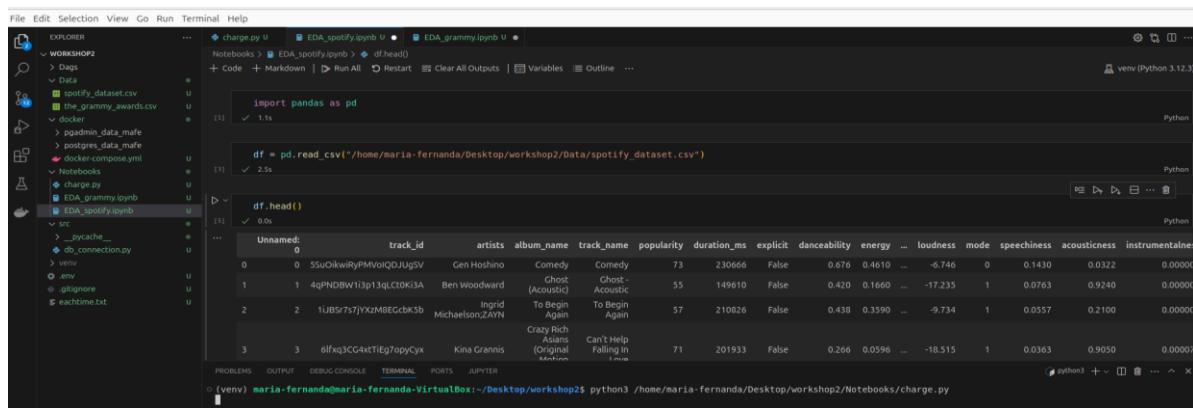
```
marifernanda@marifernanda-VirtualBox:~/Desktop/workshop2/docker$ sudo docker
compose up -d
[sudo] password for marifernanda:
[+] Running 1/1
  ✓ Container db_mafe_test Started
marifernanda@marifernanda-VirtualBox:~/Desktop/workshop2/docker$ 0.8s
```

We execute the project.

```
task requires SQLAlchemy<1.3, but you have SQLAlchemy 2.0.33 which is incompatible.
Successfully installed sqlalchemy-2.0.35
• (venv) marifernanda@marifernanda-VirtualBox:~/Desktop/workshop2$ python3 /home/marifernanda/Desktop/workshop2/Notebooks/charge.py
• (venv) marifernanda@marifernanda-VirtualBox:~/Desktop/workshop2$
```

Following this we identify the transformations that we must make in the EDA of both datasets.

EDA_grammy

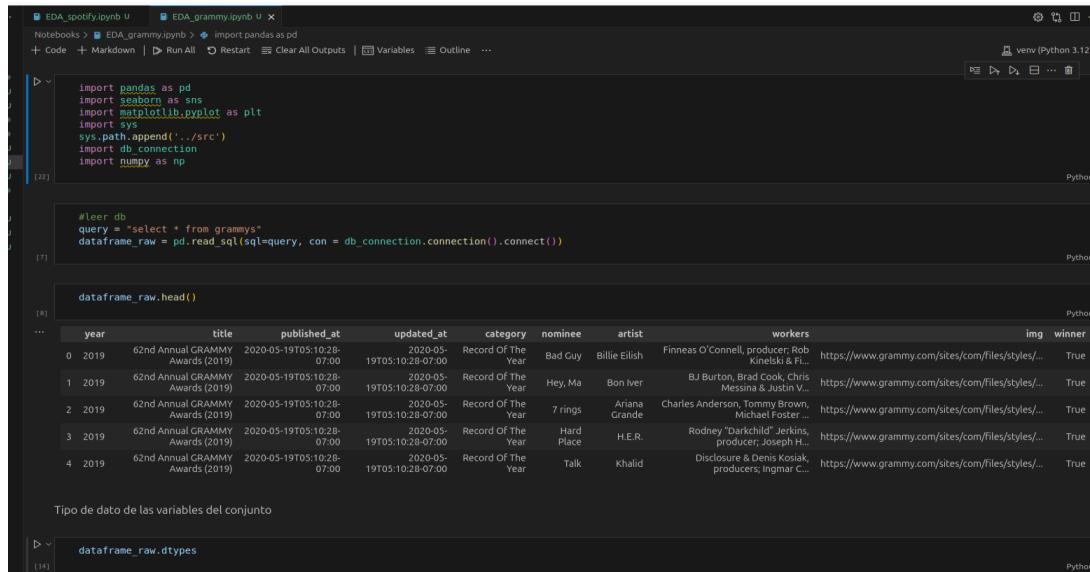


```
File Edit Selection View Go Run Terminal Help
EXPLORER WORKSHOP2 Dags Data spotify_dataset.csv the_grammy_awards.csv docker pgadmin_data_mafe postgres_data_mafe docker-compose.yml Notebooks charge.py EDA_grammy.ipynb EDA_spotify.ipynb
import pandas as pd
df = pd.read_csv("~/home/marifernanda/Desktop/workshop2/Data/spotify_dataset.csv")
df.head()
```

	track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	energy	loudness	mode	speechiness	acousticness	instrumentalness
0	55uOikwirPMvoiQDUGSV	Gen Hoshino	Comedy	Comedy	73	230666	False	0.676	0.4610	-0.746	0	0.1430	0.0322	0.00000
1	4qPnDB7t13p13qLC0K3A	Ben Woodward	Ghost (Acoustic)	Ghost	55	149610	False	0.420	0.1660	-17.235	1	0.0763	0.9240	0.00000
2	1UB5r7s7YXkM8EGcbK5b	Ingrid Michaelson/ZAYN	To Begin Again	To Begin Again	57	210826	False	0.438	0.3590	-9.734	1	0.0557	0.2100	0.00000
3	0lfxq3CG4xTie7opyCyx	Kina Grannis	Crazy Rich Asians (Original Motion Picture Soundtrack)	Can't Help Falling In Love	71	201933	False	0.266	0.0596	-10.515	1	0.0363	0.9050	0.00000

```
(venv) marifernanda@marifernanda-VirtualBox:~/Desktop/workshop2$ python3 /home/marifernanda/Desktop/workshop2/Notebooks/charge.py
```

EDA_spotify



```
File Edit Selection View Go Run Terminal Help
EXPLORER WORKSHOP2 Dags Data EDA_grammy.ipynb EDA_spotify.ipynb
Notebooks > EDA_spotify.ipynb > ⚡ Import pandas as pd
+ Code + Markdown | ▶ Run All | ⚡ Restart | Clear All Outputs | Variables | Outline ... Python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import sys
sys.path.append('../src')
import db_connection
import numpy as np

# Leer db
query = "select * from grammys"
dataframe_raw = pd.read_sql(sql=query, con = db_connection.connection().connect())

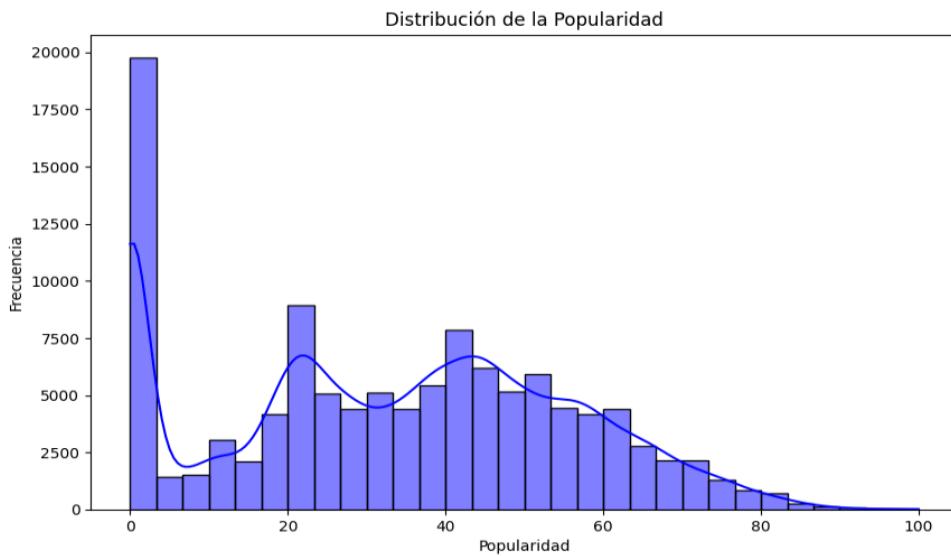
dataframe_raw.head()
```

year	title	published_at	updated_at	category	nominee	artist	workers	img	winner
0 2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	19T05:10:28-07:00	Record Of The Year	Bad Guy	Billie Eilish	Finneas O'Connell, producer; Rob Kinelski & ...	https://www.grammy.com/sites/com/files/styles/...	True
1 2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	19T05:10:28-07:00	Record Of The Year	Hey, Ma	Bon Iver	BJ Burton, Brad Cook, Chris Messina & Justin V...	https://www.grammy.com/sites/com/files/styles/...	True
2 2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	19T05:10:28-07:00	Record Of The Year	7 rings	Ariana Grande	Charles Anderson, Tommy Brown, Michael Foster ...	https://www.grammy.com/sites/com/files/styles/...	True
3 2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	19T05:10:28-07:00	Record Of The Year	Hard Place	H.E.R.	Rodney 'Darkchild' Jerkins, producer; Joseph H...	https://www.grammy.com/sites/com/files/styles/...	True
4 2019	62nd Annual GRAMMY Awards (2019)	2020-05-19T05:10:28-07:00	19T05:10:28-07:00	Record Of The Year	Talk	Khalid	Disclosure & Denis Kosiak, producers; Ingmar C...	https://www.grammy.com/sites/com/files/styles/...	True

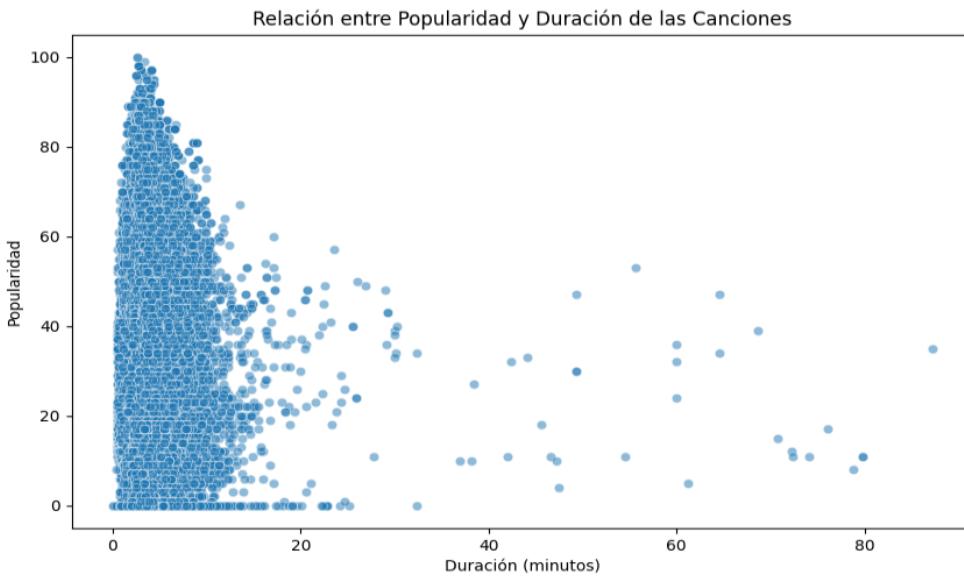
Tipo de dato de las variables del conjunto

```
dataframe_raw.dtypes
```

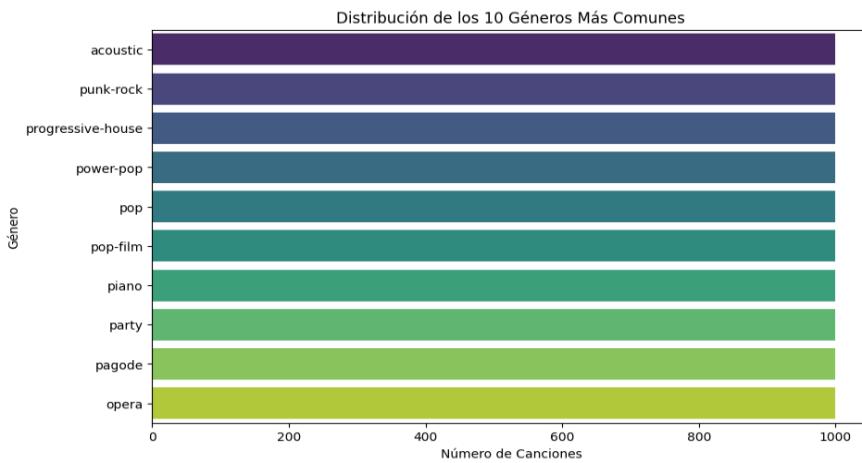
Eda visualizations of spotify dataset



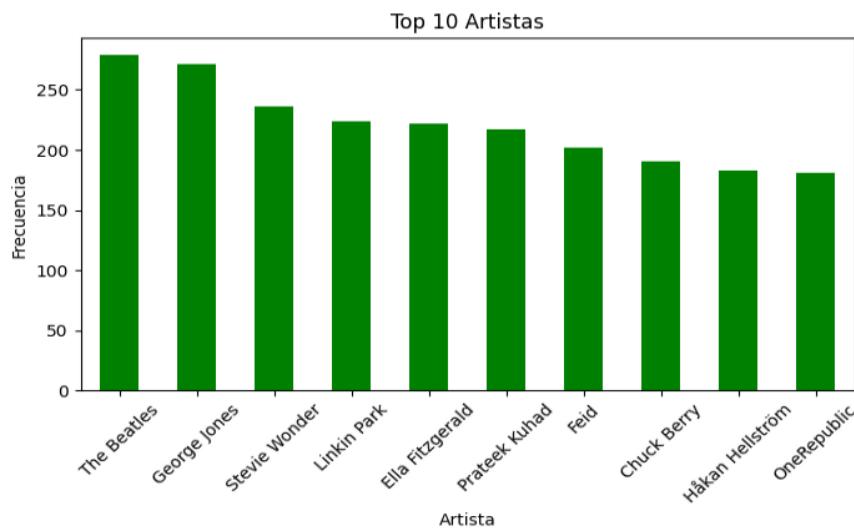
This graph shows the popularity distribution of songs in the dataset. Most songs have low popularity (around 0-20), with a significant peak in the 0-10 range, indicating that a large number of songs are unpopular. However, a more dispersed distribution is observed towards higher values, showing that some songs reach moderate and high popularity (up to 100), although in smaller numbers.



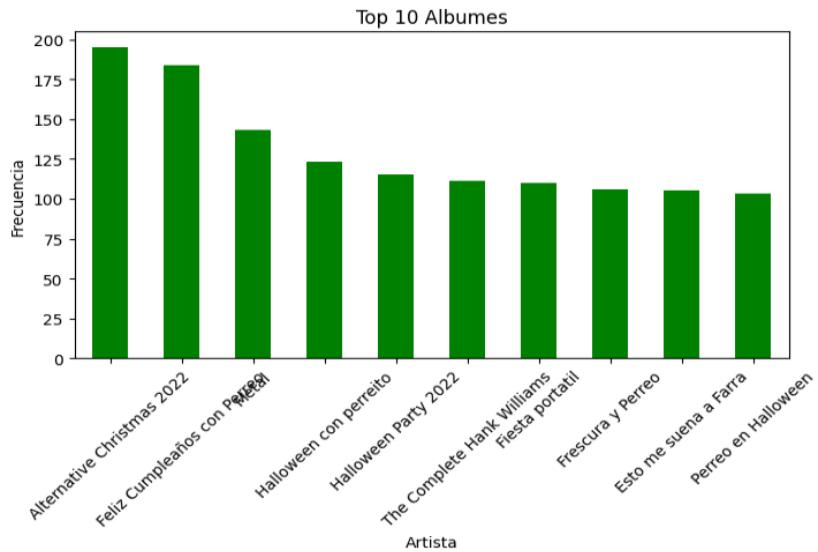
This scatter plot shows the relationship between popularity and song length. Most songs are between 2 and 5 minutes long, regardless of their popularity. There doesn't seem to be a clear correlation between length and popularity, as songs with different lengths have popularity ranging from 0 to 100. However, songs with extremely long lengths (more than 10 minutes) tend to have lower popularity.



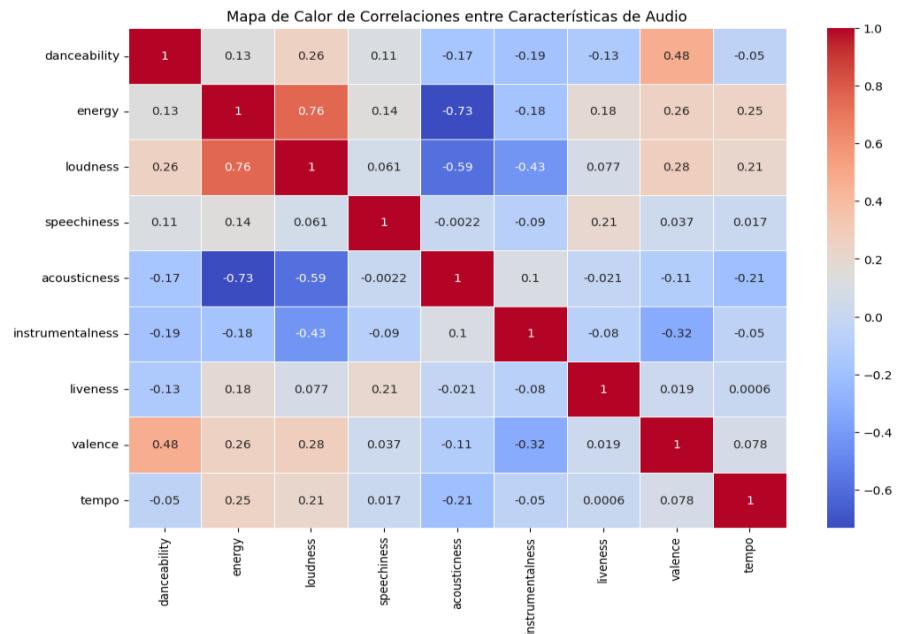
This bar chart shows the distribution of the 10 most common music genres in the Spotify dataset. It can be seen that the genres "acoustic" and "punk-rock" are the most represented, with over 1000 songs each. As we go down the list, genres like "opera" and "pagode" are present, but with a smaller number of songs.



The bar chart shows the top 10 most frequent artists in the Spotify dataset, with The Beatles and George Jones standing out as the most represented, with around 250 songs each. They are followed by Stevie Wonder, Linkin Park, and Ella Fitzgerald, with over 200 songs. Other artists such as Prateek Kuhad, Feid, and Chuck Berry also have a notable presence in the dataset, although in smaller numbers.



The bar chart shows the top 10 albums in the Spotify dataset. The most represented album is "Alternative Christmas 2022," with around 200 songs. It is followed by "Feliz Cumpleaños con Perreo" with approximately 150 songs. Other notable albums include "Halloween con perro," "Halloween Party 2022," and "The Complete Hank Williams," all with a considerable number of songs.

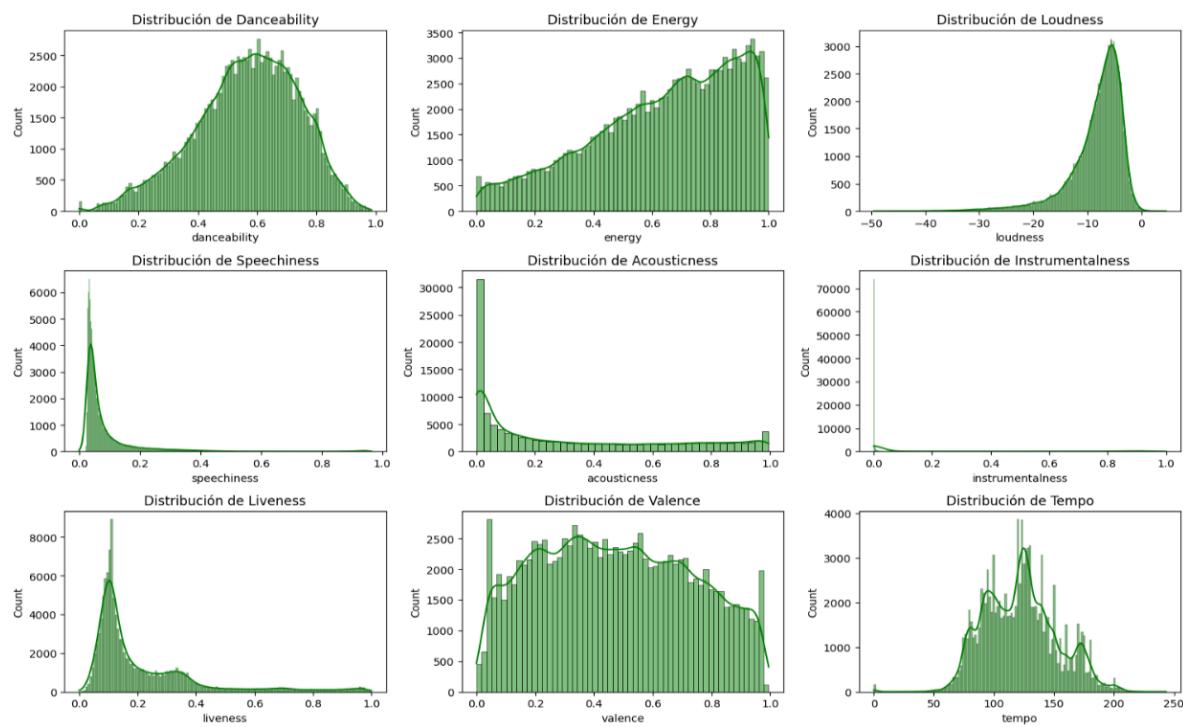


This heatmap shows the correlations between different audio characteristics of songs in the Spotify dataset.

Here are some important relationships:

- * There is a strong positive correlation between energy and loudness (0.76), indicating that more energetic songs tend to be louder.

- * Danceability and valence have a moderate correlation (0.48), suggesting that more danceable songs tend to have a more positive or upbeat tone.
- * Acousticness has a strong negative correlation with energy (-0.73) and loudness (-0.59), implying that more acoustic songs tend to be less energetic and softer.



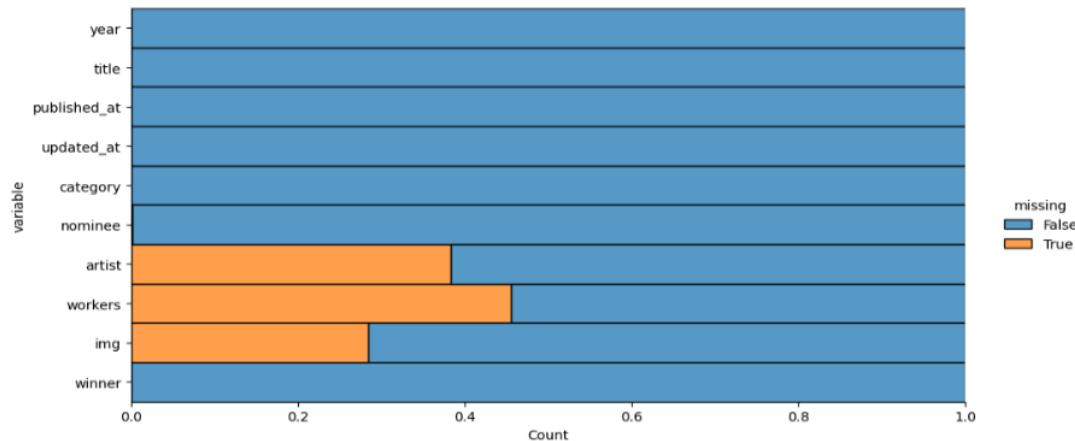
This set of graphs shows the distributions of various audio features in the Spotify dataset.

Here are some key takeaways:

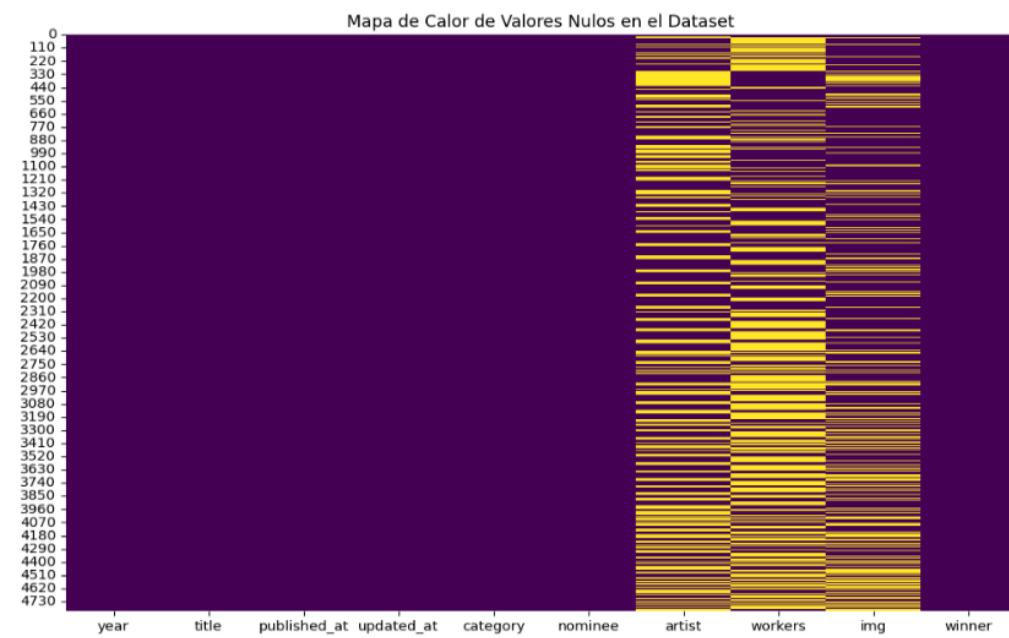
1. Danceability: Most songs have a danceability value between 0.5 and 0.8, suggesting that many songs are quite danceable.
2. Energy: The energy of the songs is fairly evenly distributed, with a slight bias toward high values, implying that the songs tend to be energetic.
3. Loudness: Most songs have a loudness value between -10 and 0 dB, indicating that they are generally quite loud.
4. Speechiness: Most songs have very low speechiness values, meaning that most are primarily musical rather than spoken.
5. Acousticness: A large portion of the songs have a low acousticness value, suggesting that many songs are not acoustic.
6. Instrumentalness: Most songs have a value close to 0, indicating that most songs have lyrics and are not instrumental.

7. Liveness: Most songs have low liveness values, suggesting that most were not recorded live.
8. Valence: Valence is fairly evenly distributed, suggesting a mix of songs with positive and negative emotions.
9. Tempo: Tempo shows a bimodal distribution, with peaks around 100 and 120 BPM, which is typical for moderate to fast-paced songs.

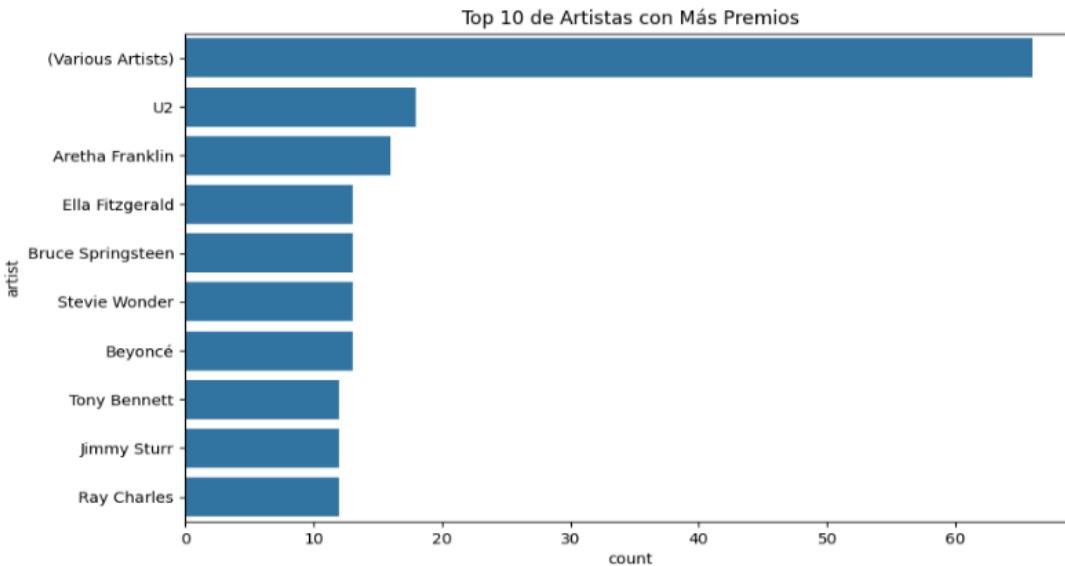
Eda visualizations of Grammy dataset



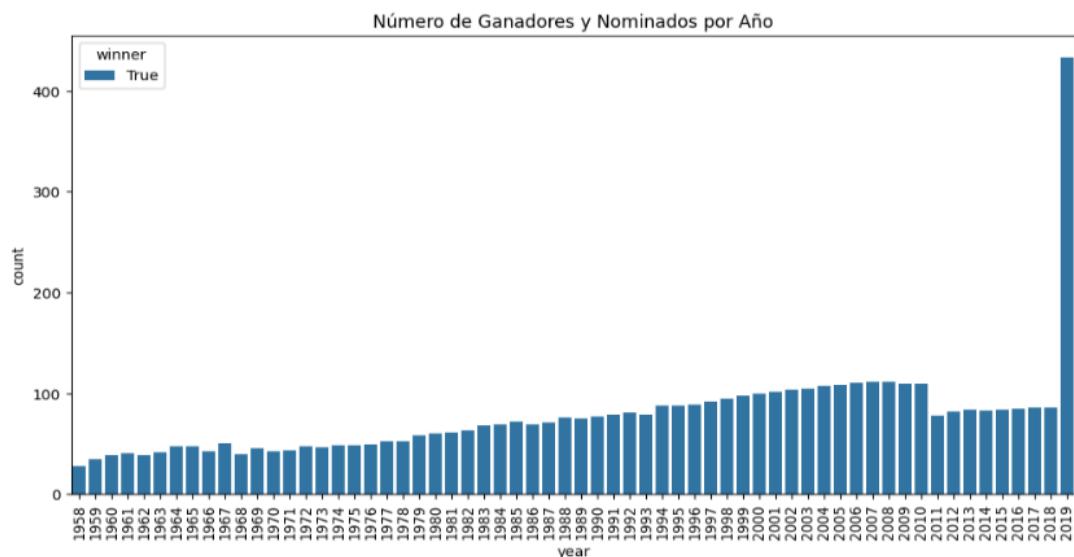
In this chart we can see that the columns `year`, `title`, `published_at`, `updated_at`, `category`, and `winner` are completely filled, while the columns `artist`, `workers`, and `img` have a significant amount of missing data, with approximately 40%, 50%, and 30% of null values respectively.



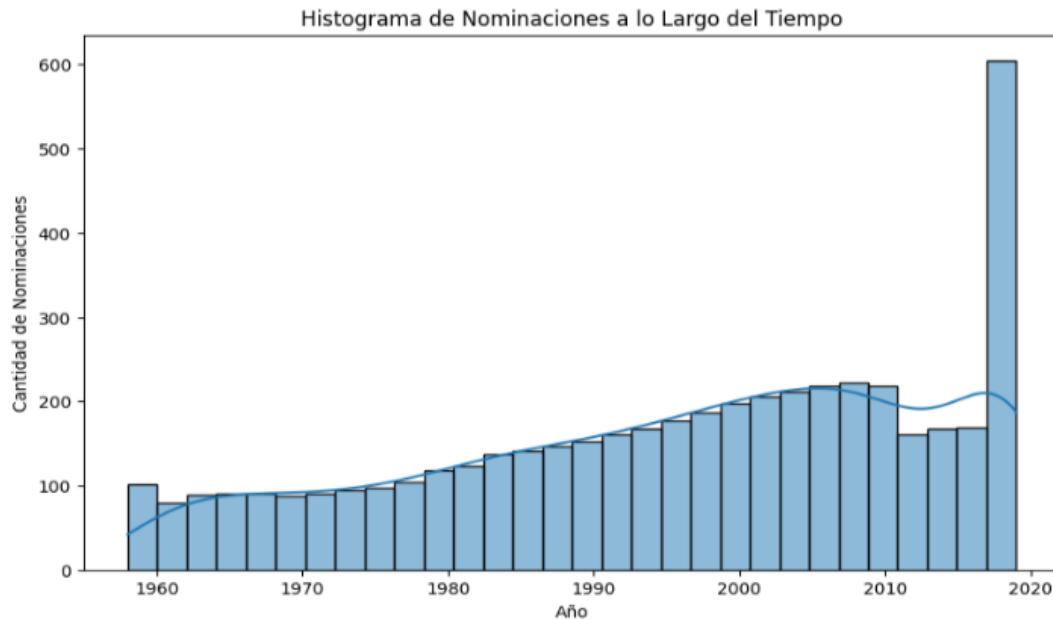
The null value heatmap shows that columns such as `year`, `title`, `category`, and `winner` have no missing values, while columns `artist`, `workers`, and `img` have a significant amount of null values. The `workers` column is the most affected, with approximately 50% missing data, followed by `artist` and `img`.



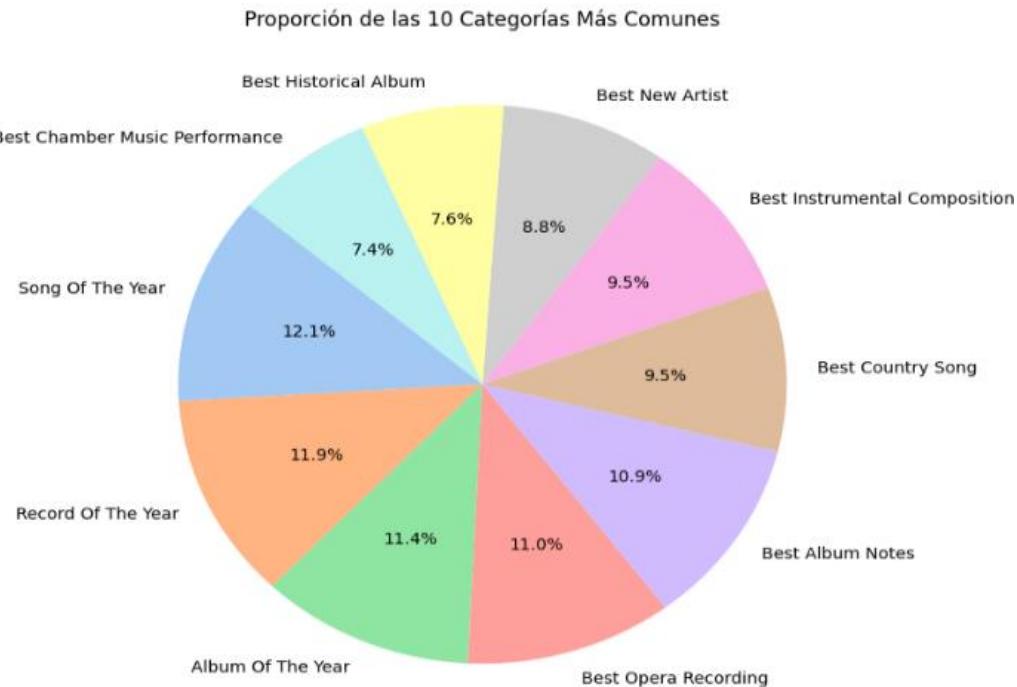
The chart shows the top 10 artists with the most Grammy wins. In first place is "Various Artists," which could refer to collaborations or compilations of several artists that have won a significant number of awards, far outnumbering the others. Among individual artists, U2 tops the list, followed by Aretha Franklin, Ella Fitzgerald, and other big names such as Bruce Springsteen, Stevie Wonder, and Beyoncé.



The chart shows the number of winners per year at the Grammy Awards from 1958 to 2019. There is a general upward trend in the number of winners over the years, with the most notable increase beginning in the 1980s. The number of winners seems to stabilize in the 2000s, with slight fluctuations. However, in 2019 there is a significant jump, which could be due to an increase in the number of categories or special awards that year.



This histogram shows the number of Grammy Award nominations over time. Since the 1960s, there has been a gradual increase in nominations, with more marked growth beginning in the 1980s and a notable peak in 2020.



The pie chart shows the proportion of the 10 most common categories at the Grammy Awards. "Song of the Year" is the most represented category, with 12.1% of the total, followed by "Record of the Year" and "Album of the Year," with shares close to 11%. Other significant categories include "Best New Artist" and "Best Instrumental Composition," each with around 9-10%. The least frequent categories in this group are "Best Chamber Music Performance" and "Best Historical Album."

Now we are going to carry out the process with airflow.

After defining the functions to do in Airflow, we wrote the code for each one and for the dag.

GRAMMY LIST

We read the data from the postgres db

We eliminate duplicates.

We eliminate the columns img, workers, updated_at and title.

We eliminate repeated data in nominee and category

We convert boolean values of the "winner" column to 1 and 0.

SPOTIFY LIST.

We read the data from the csv

We eliminate null data.

Transformation: we eliminate the column Unnamed: 0 and track_id.

Dag.py

```

WORKSHOP2
└── Dags
    ├── dag.py
    ├── Grammys.py
    ├── Load.py
    ├── Merge.py
    └── Spotify.py
    └── logs
        ├── dag_processor_manager
        └── scheduler
    └── .gitignore
    └── airflow-webserver.pid
    └── airflow.cfg
    └── airflow.db

Dags > dags > dag > dag.py > ...
1  from datetime import datetime, timedelta
2  from airflow import DAG
3  from airflow.operators.python import PythonOperator
4  import sys
5
6  # Agregar la ruta a los archivos de funciones
7  sys.path.append("/home/maria-fernanda/Desktop/workshop2/Dags/dags/dag/dag.py")
8
9  # Importar las funciones
10 from Spotify import ExtractSpotify, task_transform_dropna, task_transform_drop_columns
11 from Grammys import [
12     ExtractGrammys,
13     task_transform_drop_duplicates,
14     task_transform_drop_columns as grammys_drop_columns,
15     task_transform_drop_rows,
16     task_transform_convert_winner
17 ]
18 from Merge import merge_data

```

Grammys.py

Dags > dags > Grammys.py > task_transform_drop_duplicates

```

1 import pandas as pd
2 import json
3 from datetime import datetime
4
5 # Configuración del logger
6 def ExtractGrammys(**kwargs):
7     """
8     Extrae los datos desde la base de datos Postgres y los convierte a JSON.
9     """
10    try:
11        print(f"[{datetime.now()}] - Iniciando extracción de datos de Grammys")
12        query = "SELECT * FROM grammys"
13        db_connection = kwargs.get('db_connection') # Asumiendo que tienes una conexión disponible
14        df = pd.read_sql(sql=query, con=db_connection)
15        print(f"[{datetime.now()}] - Datos extraídos exitosamente")
16        return df.to_json(orient="records")
17    except Exception as err:
18        print(f"[{datetime.now()}] - Error en ExtractGrammys: {err}")
19        raise
20
21 def task_transform_drop_duplicates(**kwargs):
22     """
23     Elimina las filas duplicadas del DataFrame.
24     """
25    try:
26        print(f"[{datetime.now()}] - Iniciando transformación: Eliminando duplicados")
27        df = pd.read_json(kwargs['ti'].xcom_pull(task_ids='ExtractGrammys'))
28        df.drop_duplicates(inplace=True)
29        df.to_json(kwargs['ti'].xcom_push(value=df.to_json(), task_ids='task_transform_drop_duplicates'))
30    except Exception as err:
31        print(f"[{datetime.now()}] - Error en task_transform_drop_duplicates: {err}")
32        raise

```

Spotify.py

Dags > dags > Spotify.py

```

1 import pandas as pd
2 import json
3 from datetime import datetime
4
5 # Configuración del logger
6 def ExtractSpotify(**kwargs):
7     """
8     Extrae los datos desde un archivo CSV y los convierte a JSON.
9     """
10    try:
11        print(f"[{datetime.now()}] - Iniciando extracción de datos de Spotify")
12        df = pd.read_csv("/home/maria-fernanda/Desktop/workshop2/Data/spotify_dataset.csv")
13        print(f"[{datetime.now()}] - Datos extraídos exitosamente")
14        return df.to_json(orient="records")
15    except Exception as err:
16        print(f"[{datetime.now()}] - Error en ExtractSpotify: {err}")
17        raise
18
19 def task_transform_dropna(**kwargs):
20     """
21     Elimina las filas con datos nulos.
22     """
23    df = pd.read_json(kwargs['ti'].xcom_pull(task_ids='ExtractSpotify'))
24    df.dropna(inplace=True)
25    df.to_json(kwargs['ti'].xcom_push(value=df.to_json(), task_ids='task_transform_dropna'))
26
```

Load.py

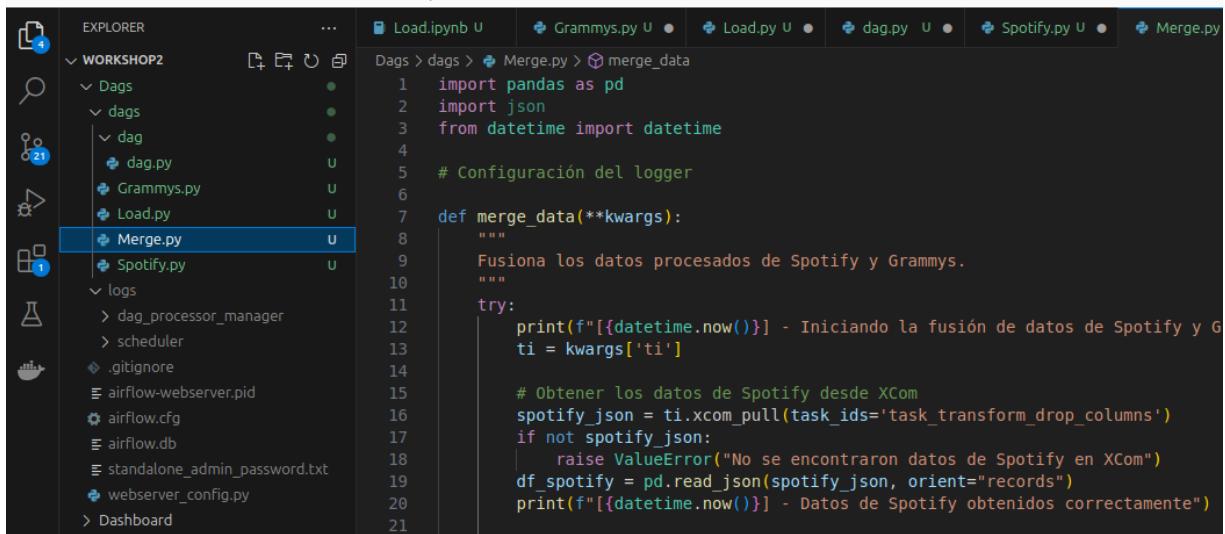
Dags > dags > Load.py > load_to_postgres

```

1 import pandas as pd
2 import json
3 from datetime import datetime
4 import sys
5 sys.path.append("src")
6 import db_connection
7
8 def load_to_postgres(**kwargs) -> None:
9     """
10     Carga los datos fusionados a la base de datos Postgres.
11     """
12     try:
13         print(f"[{datetime.now()}] - Iniciando carga de datos a Postgres")
14         ti = kwargs['ti']
15         db_connection.create_table(ti.xcom_pull(task_ids='task_transform_dropna'))
16         db_connection.insert_data(ti.xcom_pull(task_ids='task_transform_dropna'))
17     except Exception as err:
18         print(f"[{datetime.now()}] - Error en load_to_postgres: {err}")
19         raise

```

Merge.py



```

EXPLORER          ...   Load.ipynb U  Grammys.py U  Load.py U  dag.py U  Spotify.py U  Merge.py
WORKSHOP2
  Dags
    dags
      dag
        dag.py
        Grammys.py
        Load.py
        Merge.py
        Spotify.py
      logs
        dag_processor_manager
        scheduler
      .gitignore
      airflow-webserver.pid
      airflow.cfg
      airflow.db
      standalone_admin_password.txt
      webserver_config.py
      Dashboard
Dags > dags > Merge.py > merge_data
1 import pandas as pd
2 import json
3 from datetime import datetime
4
5 # Configuración del logger
6
7 def merge_data(**kwargs):
8     """
9         Fusiona los datos procesados de Spotify y Grammys.
10    """
11    try:
12        print(f"[{datetime.now()}] - Iniciando la fusión de datos de Spotify y G
ti = kwargs['ti']
13
14        # Obtener los datos de Spotify desde XCom
15        spotify_json = ti.xcom_pull(task_ids='task_transform_drop_columns')
16        if not spotify_json:
17            raise ValueError("No se encontraron datos de Spotify en XCom")
18        df_spotify = pd.read_json(spotify_json, orient="records")
19        print(f"[{datetime.now()}] - Datos de Spotify obtenidos correctamente")
20
21

```

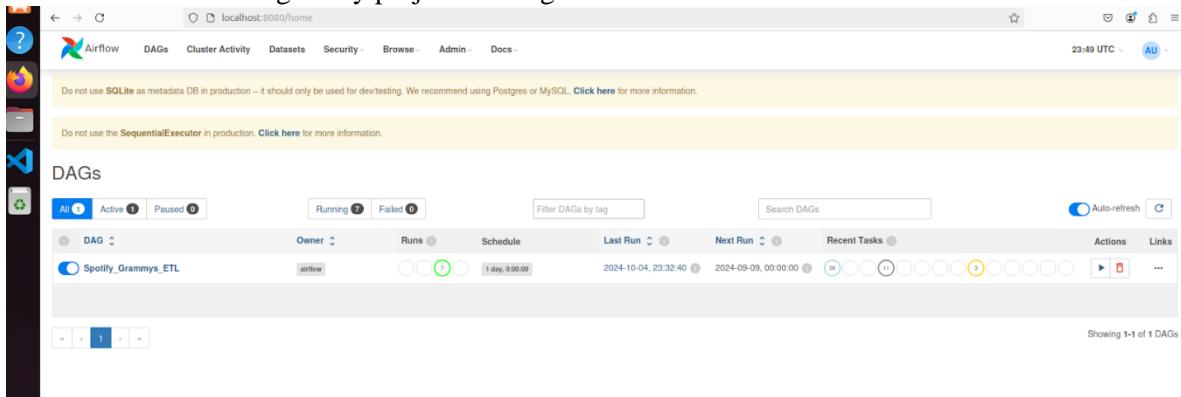
Run the airflow.

```

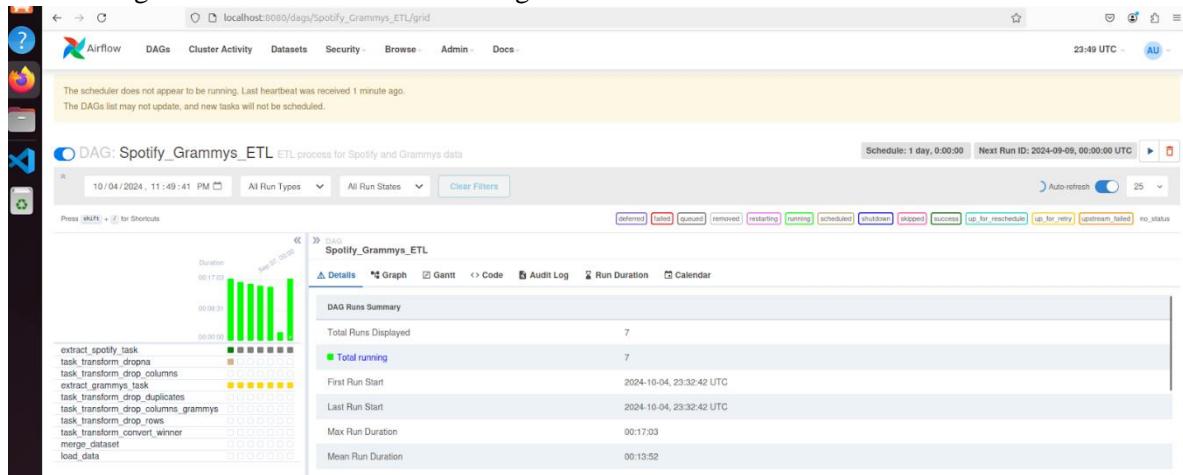
maria-fernanda@maria-fernanda-VirtualBox:~$ ls
airflow  Documents  Music  Public  Templates
Desktop  Downloads  Pictures  snap  Videos
maria-fernanda@maria-fernanda-VirtualBox:~$ cd Desktop/
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop$ ls
workshop2
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop$ cd workshop2/
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ ls
Dags      Data      eachtime.txt  merge.ipynb  src
Dashboard  docker  Load.ipynb  Notebooks  venv
maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ source venv/bin/activate
(venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ airflow standalone

```

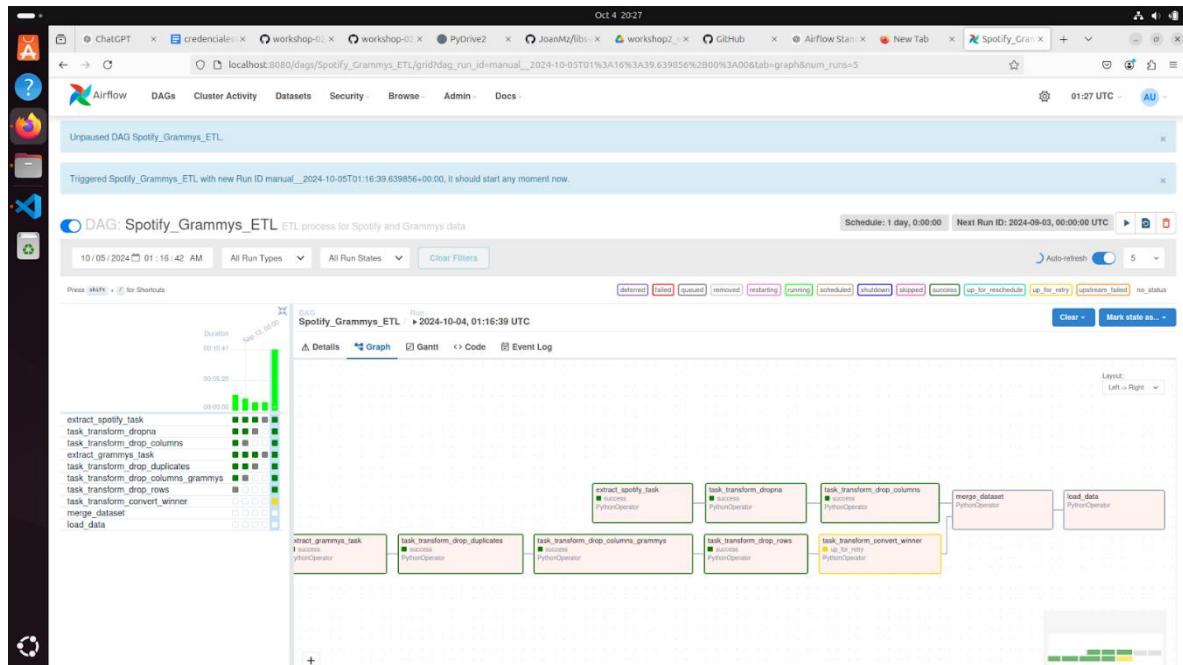
Here we can see the dag of my project running in airflow.



In this image we can see some tasks running.



Finally, we can see most of the tasks running perfectly, and we can also see that the connection to the Postgres database was made correctly, since the Grammys dataset runs perfectly from the first task.



GET ON THE DRIVE

We create a new Project

console.cloud.google.com/welcome?pli=1&project=academic-moon-433222-u8

Te damos una cuota gratuita con un crédito de \$300. No te preocupes, no se te cobrará si se acaban los créditos. [Más información](#)

Selecciona un recurso

PROYECTO NUEVO

UAO.EDU.CO

Buscar en proyectos y carpetas

RECENTES PRESENTADO TODO

Nombre	IDENTIFICACIÓN
✓ ⭐ Mi primer proyecto ⓘ	luna academica-433222-u8
⭐ ⓘ Mi proyecto 89990 ⓘ	incondicional-fx-382221

Estás trabajando en
Número de proyecto: 576135

Tienes 10 projects restantes en tu cuota. Solicita un incremento o borra algunos proyectos. [Más información](#)

[ADMINISTRAR CUOTAS](#)

Nombre del proyecto * Workshop2PyDrive

ID del proyecto: workshop2pydrive. No se puede cambiar más adelante. [EDITAR](#)

Organización * uao.edu.co

Selecciona una organización para vincularla a un proyecto. No podrás cambiar esta selección más adelante.

Ubicación * uao.edu.co [EXPLORAR](#)

Organización de la alfombra superior.

CREAR **CANCELAR**

We enable the Google Drive API for our project.

≡ Google Cloud **Taller2PyDrive**

← Detalles del producto

API de Google Drive

[API de Google Enterprise](#)

Crea y administra recursos en Google Drive.

PERMITIR **PROBAR ESTA API**

Configure the consent screen.

Menú de navegación () servicios

- APIs y servicios habilitados
- Biblioteca
- Credenciales
- Pantalla de consentimiento de OAuth
- Acuerdos de uso de páginas

Editar el registro de la app

1 Pantalla de consentimiento de OAuth — 2 Permisos — 3 Usuarios de prueba — 4 Resumen

Información de la aplicación

Esta información aparece en la pantalla de consentimiento y permite que los usuarios finales sepan quién eres y cómo comunicarse contigo

Nombre de la aplicación *

El nombre de la aplicación que solicita el consentimiento

We create the credentials.

[← Crear ID de cliente de OAuth](#)

Un ID de cliente se usa con el fin de identificar una sola app para los servidores de OAuth de Google. Si la app se ejecuta en varias plataformas, cada una necesitará su propio ID de cliente. Consulta [Configura OAuth 2.0](#) para obtener más información. [Obtén más información](#) sobre los tipos de clientes de OAuth.

Tipo de aplicación * —
Aplicación web

Nombre * —
Workshop2PyDrive

El nombre de tu cliente de OAuth 2.0. Este nombre solo se usa para identificar al cliente en la consola y no se mostrará a los usuarios finales.

Credentials created.

Se creó el cliente de OAuth

Puedes acceder al ID de cliente y el secreto desde "Credenciales" en API y servicios

El acceso OAuth está restringido a los [usuarios de prueba](#) que aparecen en la [pantalla de consentimiento de OAuth](#)

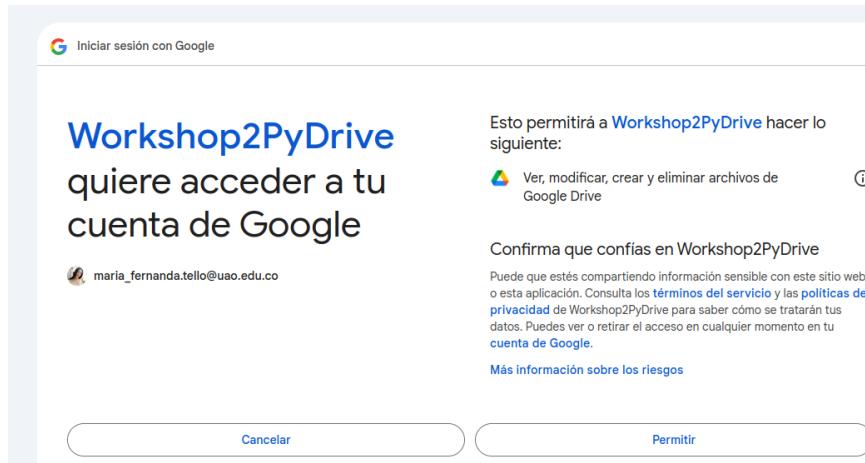
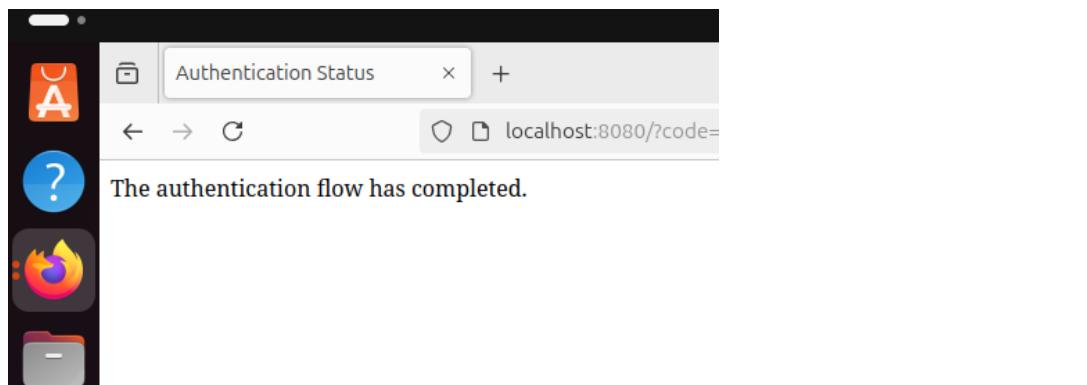
ID de cliente	411865893581-a8mv5umrg0gr89ts349oljecbl4ct80.apps.googleusercontent.com	
---------------	---	--

Secreto del cliente	GOCSPX-piPiZ0w1z3RckVnfinnrVEcipNR	
---------------------	------------------------------------	--

Fecha de creación	4 de octubre de 2024, 10:05:17 GMT-5
-------------------	--------------------------------------

Estado	Activado
--------	----------

DESCARGAR JSON

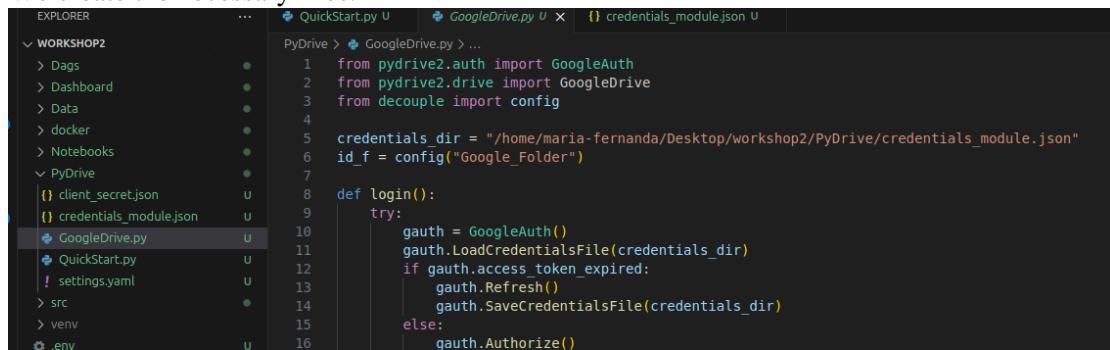



The terminal window title is "Authentication Status". The URL in the address bar is "localhost:8080/?code=". The content of the window says "The authentication flow has completed."

```
(venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/PyDrive$ python3 QuickStart.py
/home/maria-fernanda/Desktop/workshop2/venv/lib/python3.12/site-packages/oauth2client/_helpers.py:1
    warnings.warn(_MISSING_FILE_MESSAGE.format(filename))
Your browser has been opened to visit:

    https://accounts.google.com/o/oauth2/auth?client_id=411865893581-a8mt5umrg0gr89ts349oljecbl4c
e=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive&access_type=offline&response_type=code&approval_
> Authentication successful.
○ (venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/PyDrive$
○ (venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/PyDrive$
○ (venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2/PyDrive$
```

We create the necessary files.



The code editor shows a file named "GoogleDrive.py" with the following content:

```
from pydrive2.auth import GoogleAuth
from pydrive2.drive import GoogleDrive
from decouple import config

credentials_dir = "/home/maria-fernanda/Desktop/workshop2/PyDrive/credentials_module.json"
id_f = config("Google_Folder")

def login():
    try:
        gauth = GoogleAuth()
        gauth.LoadCredentialsFile(credentials_dir)
        if gauth.access_token_expired:
            gauth.Refresh()
            gauth.SaveCredentialsFile(credentials_dir)
        else:
            gauth.Authorize()
```

We configure the GoogleDrive.py file, and perform a test to make sure it works when uploading files to my Google Drive.

```

    title[title] = path_file.split("/")[-1]
    file.SetContentFile(path_file)
    file.Upload()

if __name__ == "__main__":
    Create_file("testWorkshop2.txt", "holi holi", "1uX723HTbBtuEdze_6xVfD0nrH3059G_t")
    Upload_file("../Data/merge.csv", "1uX723HTbBtuEdze_6xVfD0nrH3059G_t")
    pass

```

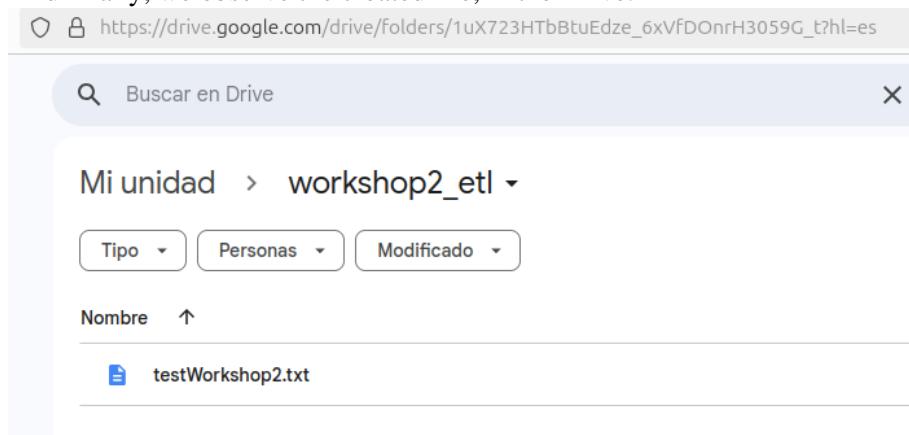
Here we can see it running.

```

• (venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ /home/maria-fernanda/Desktop/workshop2/venv/bin/python
/home/maria-fernanda/Desktop/workshop2/PyDrive/GoogleDrive.py
○ (venv) maria-fernanda@maria-fernanda-VirtualBox:~/Desktop/workshop2$ █

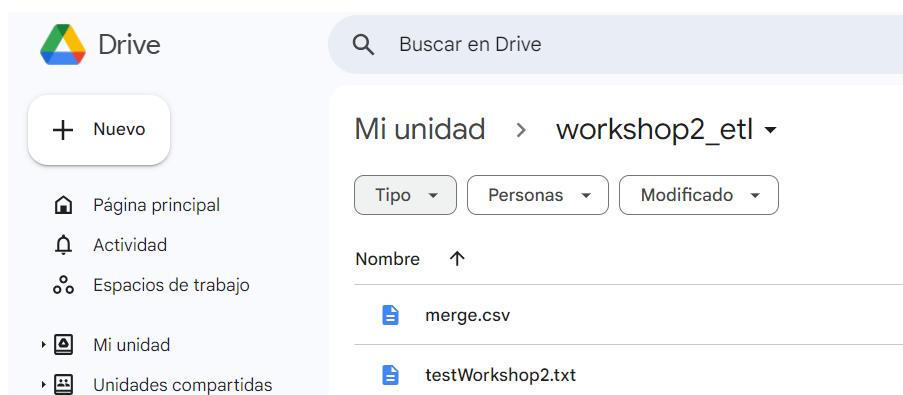
```

And finally, we observe the created file, in the Drive.



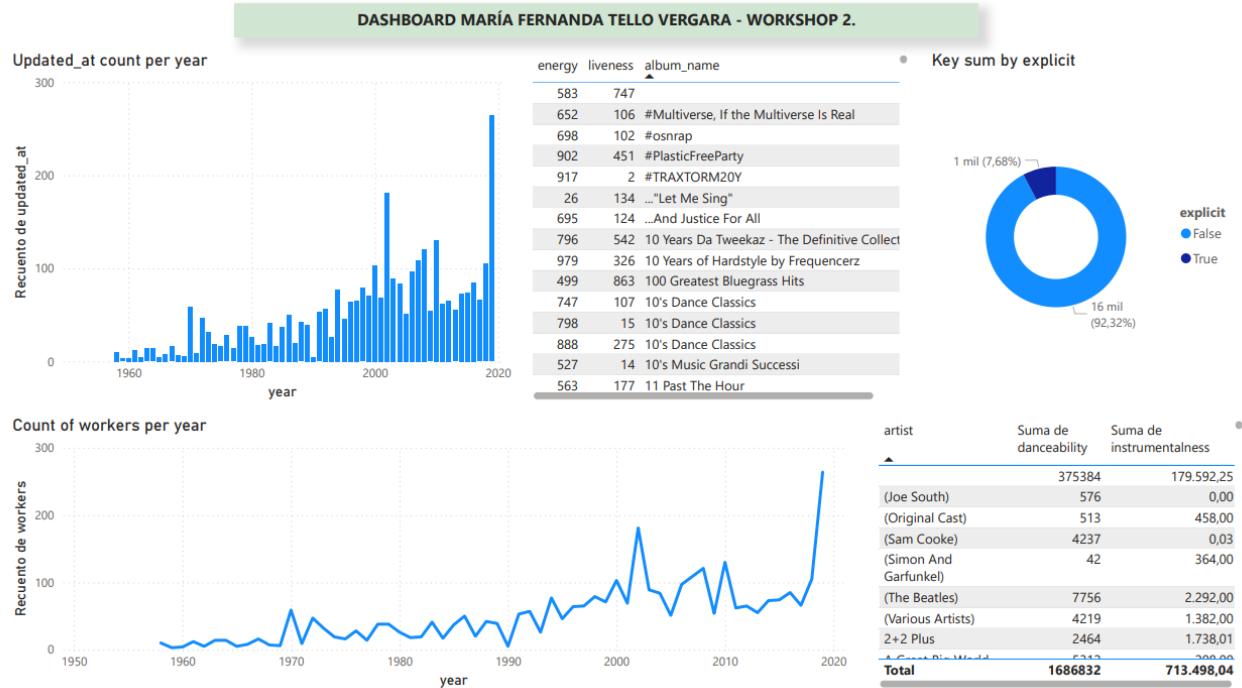
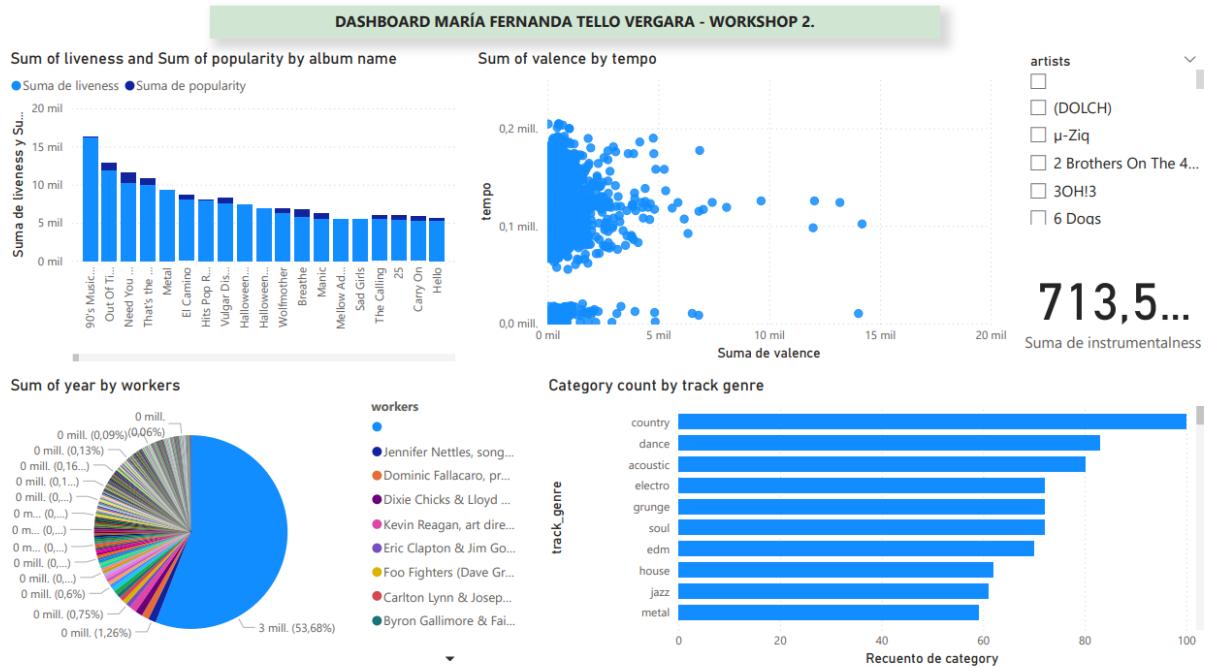
The screenshot shows a Google Drive interface. At the top, there's a search bar with 'Buscar en Drive'. Below it, a breadcrumb navigation shows 'Mi unidad > workshop2_etl'. There are three filter buttons: 'Tipo', 'Personas', and 'Modificado'. A sorting button 'Nombre ↑' is also present. Under the folder, two files are listed: 'testWorkshop2.txt' and 'merge.csv'.

Furthermore, in the following image we can see that it actually loads the merge.csv file, in the same folder.



This screenshot shows the same Google Drive interface as the previous one, but with a different set of files in the 'workshop2_etl' folder. It lists 'merge.csv' at the top and 'testWorkshop2.txt' below it. The left sidebar shows standard Google Drive navigation options like 'Nuevo', 'Página principal', 'Actividad', 'Espacios de trabajo', 'Mi unidad', and 'Unidades compartidas'.

VISUALIZATIONS



Here are some of the questions we'll be answering in the Power BI dashboard:

- How much has the number of updates been per year?
- How has the number of workers changed over the years?
- Which albums have the highest sum of liveness and popularity?
- What proportion of the music is explicit or not?
- How are tempo and valence distributed between different tracks?
- Which workers contributed to the songs with the highest liveness and popularity?
- What are the most common musical genres in the tracks?
- Which artists have the highest sum of "danceability" and instrumentality?

Finally, we uploaded all the folders to the GitHub repository!