

Descripción de la columna DATASET SPOTIFY

track_id: El ID de Spotify de la canción

artists: Los nombres de los artistas que interpretaron la canción. Si hay más de un artista, se separan por un ;

album_name: El nombre del álbum en el que aparece la canción

track_name: Nombre de la canción

popularity: La popularidad de una pista es un valor entre 0 y 100, siendo 100 la más popular. La popularidad se calcula mediante un algoritmo y se basa, en su mayor parte, en el número total de reproducciones que ha tenido la canción y lo recientes que son esas reproducciones. En general, las canciones que se reproducen mucho ahora tendrán una mayor popularidad que las canciones que se reprodujeron mucho en el pasado. Las canciones duplicadas (por ejemplo, la misma canción de un single y de un álbum) se valoran de forma independiente. La popularidad de artistas y álbumes se obtiene matemáticamente a partir de la popularidad de las canciones.

duration_ms: La duración de la pista en milisegundos

explicit: Si la canción tiene o no letra explícita (verdadero = sí la tiene; falso = no la tiene O desconocido)

danceability: La bailabilidad describe lo adecuada que es una pista para bailar basándose en una combinación de elementos musicales que incluyen el tempo, la estabilidad del ritmo, la fuerza del compás y la regularidad general. Un valor de 0,0 es el menosailable y 1,0 el másailable.

energy: La energía es una medida de 0,0 a 1,0 y representa una medida perceptiva de intensidad y actividad. Normalmente, las pistas energéticas son rápidas, ruidosas y ruidosas. Por ejemplo, el death metal tiene mucha energía, mientras que un prelude de Bach puntúa bajo en la escala.

key: La tonalidad de la canción. Los números enteros se asignan a los tonos utilizando la notación estándar Pitch Class. Por ejemplo, 0 = Do, 1 = Do#/Db, 2 = Re, y así sucesivamente. Si no se detecta ninguna tonalidad, el valor es -1

loudness: La sonoridad general de una pista en decibelios (dB)

mode: El modo indica la modalidad (mayor o menor) de una pista, el tipo de escala del que se deriva su contenido melódico. Mayor se representa con 1 y menor con 0

speechiness: La locuacidad detecta la presencia de palabras habladas en una pista. Cuanto más exclusivamente hablada sea la grabación (por ejemplo, programa de entrevistas, audiolibro, poesía), más se acercará a 1,0 el valor del atributo. Los valores superiores a 0,66 describen pistas que probablemente estén compuestas en su totalidad por palabras habladas. Los valores entre 0,33 y 0,66 describen pistas que pueden contener tanto música como voz, ya sea en secciones o en capas, incluyendo casos como la música rap. Los valores por debajo de 0,33 representan probablemente música y otras pistas no habladas.

acousticness: Medida de confianza de 0,0 a 1,0 para determinar si la pista es acústica. 1,0 representa una confianza alta en que la pista es acústica.

instrumentalness: Predice si una pista no contiene voces. Los sonidos "ooh" y "aah" se consideran instrumentales en este contexto. Las pistas de rap o spoken word son claramente "vocales". Cuanto más se acerque el valor de instrumental a 1,0, mayor será la probabilidad de que la pista no contenga voces.

liveness: Detecta la presencia de público en la grabación. Los valores más altos representan una mayor probabilidad de que la pista haya sido interpretada en directo. Un valor superior a 0,8 indica una gran probabilidad de que la pista se haya grabado en directo.

valance: Medida de 0,0 a 1,0 que describe la positividad musical que transmite una pista. Las pistas con una valencia alta suenan más positivas (por ejemplo, felices, alegres, eufóricas), mientras que las pistas con una valencia baja suenan más negativas (por ejemplo, tristes, deprimidas, enfadadas).

tempo: el tempo global estimado de una pista en pulsaciones por minuto (BPM). En terminología musical, el tempo es la velocidad o el ritmo de una pieza determinada y se deriva directamente de la duración media de los tiempos.

time_signature: Un compás estimado. El compás es una convención para especificar cuántos tiempos hay en cada compás. El compás oscila entre 3 y 7, indicando compases de 3/4 a 7/4.

track_genre: El género al que pertenece la pista

El conjunto de datos se distribuye en: **114000 Registros o filas y 21 columnas.**

```
¿Cuántas variables y observaciones  
tenemos en el conjunto de datos?
```

```
1 df.shape  
Executed at 2023.10.14 22:19:10 in 160ms
```

```
(114000, 21)
```

Observamos que solo hay 21 columnas, las cuales 15 columnas son numericas, 5 categoricas y una booleana .

```
¿Cuántas variables de cada tipo de dato tenemos en el conjunto de datos?
```

```
(
  df
  .dtypes
  .value_counts()
)
```

Executed at 2023.10.14 22:19:10 in 176ms

< 4 rows > | Length: 4, dtype: int64 pd.Series

	<unnamed>
float64	9
int64	6
object	5
bool	1

Tenemos 3 columnas con datos nulos.

- album_name
- artist
- track_name

```
De tener observaciones con valores nulos, ¿cuántas tenemos por cada variable?
```

```
1 (
2   df
3   .isnull()
4   .sum()
5   .sort_values(ascending=False)
6 )
7
```

Executed at 2023.10.14 22:19:11 in 143ms

< 1-10 > | Length: 21, dtype: int64 pd.Series

	<unnamed>
artists	1
album_name	1
track_name	1
Unnamed: 0	0
mode	0
time_signature	0
tempo	0

Podemos ver de que esto solo pertenece a un solo registro

```
# Para obtener todas las filas con valores nulos:
rows_with_null_values = df[df.isnull().any(axis=1)]
# Luego puedes mostrar o trabajar con estas filas:
print(rows_with_null_values)
```

Executed at 2023.10.14 23:04:39 in 171ms

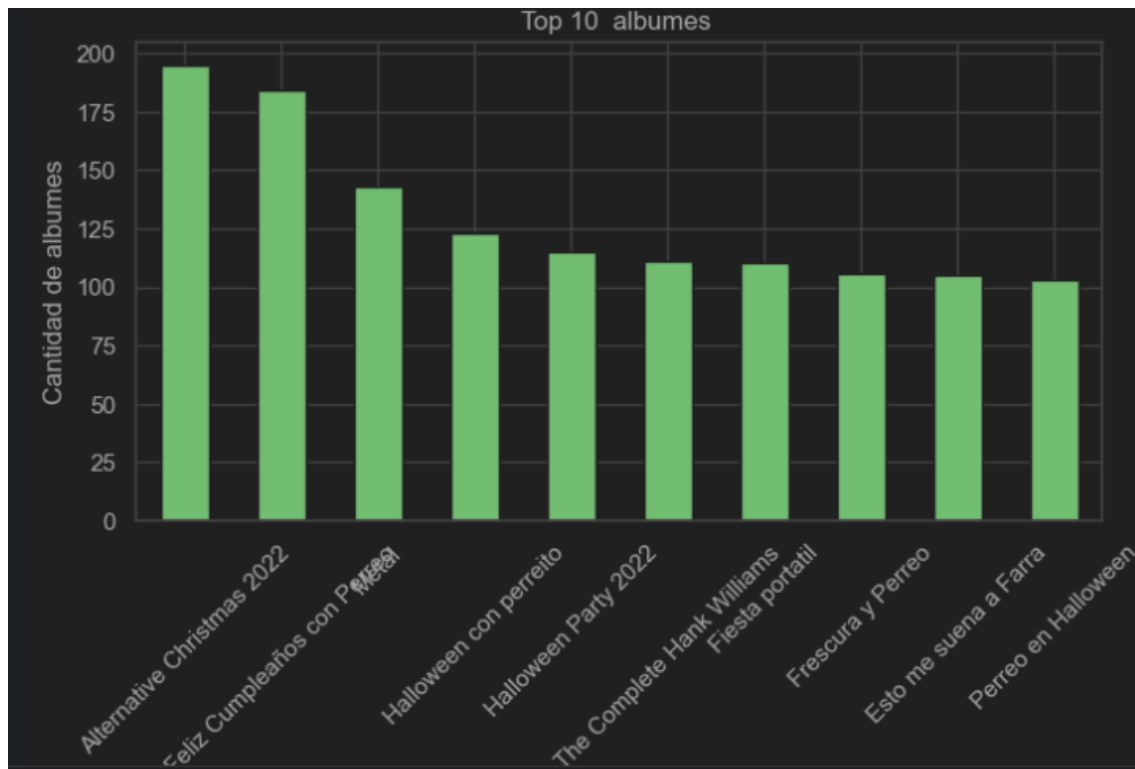
```
rows_with_null_values
```

Executed at 2023.10.14 23:04:39 in 62ms

< 1 row > | 1 rows x 21 columns pd.DataFrame

	Unnamed: 0	track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceabili
65900	65900	1KR4gIb7nGxHPI3D2ifs59	NaN	NaN	NaN	0	0	False	

Podemos ver los 10 mejores albums

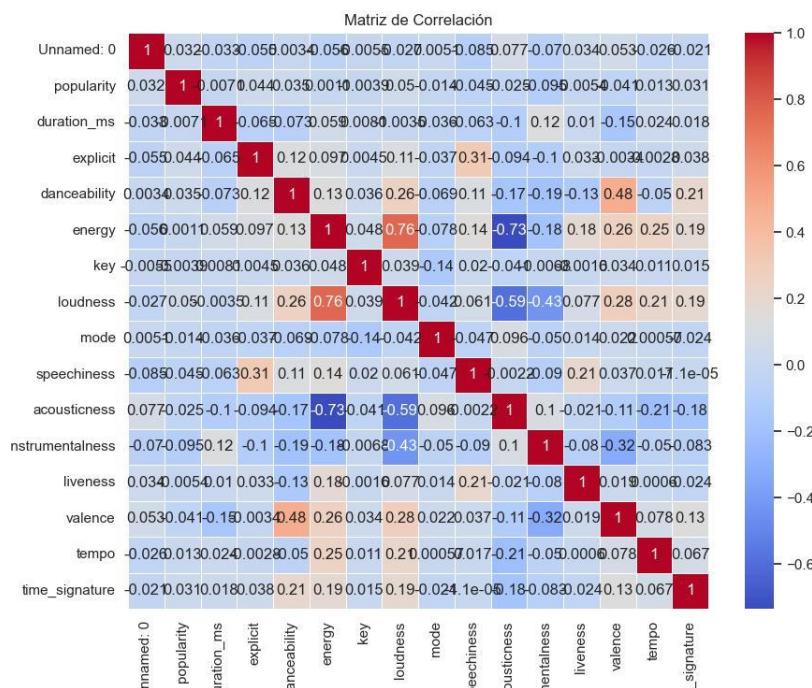


Podemos ver que los albums que tienen mas presencia son: Alternative Christmas 2022, Feliz Cumpleaños con Perreo y Metal.

Corplot

Los que tienen colores mas pronunciados a el color azul es por que tienen correlacion positiva, es decir que mientras el uno sube el otro tiende a subir tambien.

el color roja es por que tienen correlacion negativa , es decir que mientras el uno sube el otro tiende a bajar.



Loudness y energy tiene una gran correlacion, acousticness y energy tienen una correlacion negativa, es decir que mientras el uno sube el otro tiende a bajar.

Haciendo un analisis Descriptivos de las Variables:

Tipo Categoricas

El artista que mas se repite son The Beatles y el album que mas se encuentra en el dataset es Alternative Christmas 2022 con 195 apariciones.

```
df.describe(include=object)
```

Executed at 2023.10.14 22:49:58 in 188ms

	track_id	artists	album_name	track_name	track_genre
count	114000	113999	113999	113999	114000
unique	89741	31437	46589	73608	114
top	6S3JLDAGk3uu3NtZbPnuhS	The Beatles	Alternative Christmas 2022	Run Rudolph Run	acoustic
freq	9	279	195	151	1000

El conjunto de datos se distribuye en: **4810 Registros o filas y 10 columnas.**

```
¿Cuántas variables y observaciones tenemos en el conjunto de datos?
```

```
df.shape
```

Executed at 2023.09.30 14:23:31 in 17ms

```
(4810, 10)
```

Observamos que solo hay 10 columnas, las cuales una **columna numerica y una booleana** .

```
¿Cuántas variables de cada tipo de dato tenemos en el conjunto de datos?
```

```
(
    df
    .dtypes
    .value_counts()
)
```

Executed at 2023.09.30 14:23:12 in 41ms

```
object      8
int64       1
bool        1
dtype: int64
```

Tenemos 4 columnas con datos nulos.

- nominee:
- artist
- workerr
- img

< < 10 rows > > Length: 10, dtype: bool pd.Series		
	<unnamed>	
updated_at	False	
category	False	
nominee	• True	
artist	• True	
workers	• True	
img	• True	
winner	False	

Cuantos nulos hay por cada variable:

< < 10 rows > > Length: 10, dtype: int64 pd.Series		
	<unnamed>	
workers	2190	
artist	1840	
img	1367	
nominee	6	
year	0	
title	0	
published_at	0	

Para un total de 5403

Borramos los datos duplicados

¿Existen valores nulos explicitos en el conjunto de datos?

```
1 ( df
2   .isnull()
3   .any()
4 )
```

Executed at 2023.10.14 22:19:10 in 144ms

|< < 1-10 > >| Length: 21, dtype: bool [pd.Series ↗](#)

	<unnamed>
Unnamed: 0	False
track_id	False
artists	• True
album_name	• True
track_name	• True
popularity	False
duration_ms	False

Cuantos nulos hay por cada variable:

	<unnamed>
workers	2190
artist	1840
img	1367
nominee	6
year	0
title	0
published_at	0

Para un total de 5403

Borramos los datos duplicados

Cuantos Duplicados Tenemos?

```
1 duplicados = df.duplicated()
2 duplicados.sum()
```

Executed at 2023.10.14 18:29:09 in 285ms

0

Haciendo un analisis Descriptivos de las Variables:

Tipo Categoricas

La gran mayoría de las Canciones fueron publicadas en 2017-11-28T00:03:45-08:00, lo cual es muy raro.

```
df.describe(include=object)
```

Executed at 2023.09.30 14:34:18 in 169ms

	title	published_at	updated_at	category	nominee
count	4810	4810	4810	4810	4804
unique	62	4	10	638	4131
top	62nd Annual GRAMMY Awards (2019)	2017-11-28T00:03:45-08:00	2019-09-10T01:08:19-07:00	Song Of The Year	Bridge Over Troubled Wate
freq	433	4205	778	70	7

La Categoría Mas Nominada durante todos los años ha sido **Song Of The Year**, Tambien podemos evidenciar que la cancion mas nominadaes **Bridge Over Troubled Wate**, con 7 nominaciones.

Tipo Numericas

Solo las numéricas

```
df.describe(include=[np.number])
```

Executed at 2023.09.30 14:27:50 in 64ms

	year
count	4810.000000
mean	1995.566944
std	17.149720
min	1958.000000
25%	1983.000000
50%	1998.000000
75%	2010.000000
max	2019.000000

Podemos ver que los años, van desde **1958 a 2019**.

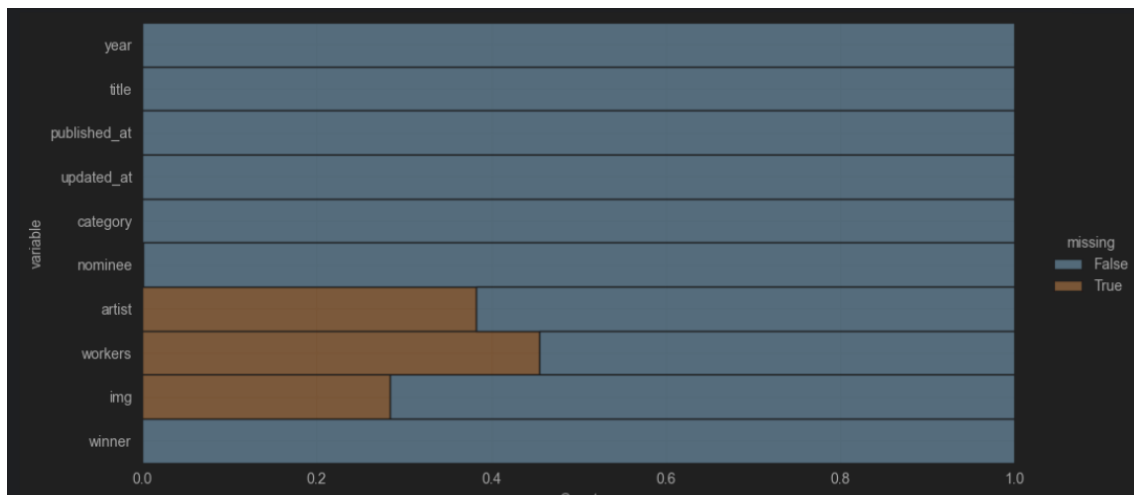
Como podemos observar, La Columna winner solo tiene un valor y es True, entonces lo mas probable es que no todos sea ganadores, si que no que todos fueron nominados.

```
df.describe(include=bool)
```

Executed at 2023.09.30 14:34:29 in 158ms

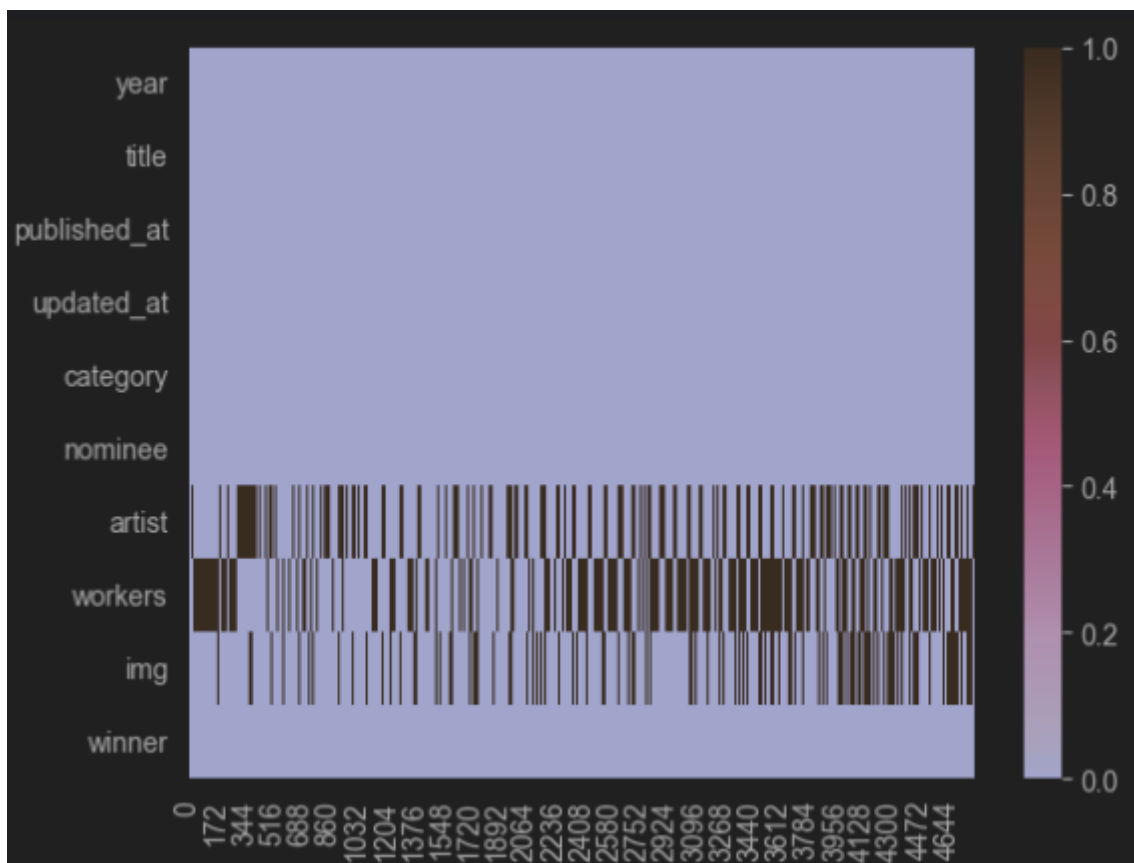
	winner
count	4810
unique	1
top	True
freq	4810

Hacemos un analisis de las proporciones de los nulos en las columnas.



Este grafico nos muestra la proporcion de nulos en cada una de las columnas, donde podemos ver de que la columna img tiene casi un 30 % de los datos nulos, y la columna artist tiene casi un 40% de los valores nulos, la columna workers mas de un 40%.

Ademas del analisis de las columnas, es importante ver como se distribuyen los nulos en las filas, para ver si hay nulos que tienen todos los registros nulos o su gran mayoria. Este grafico se interpreta de manera vertical donde cada raya, representa un registro, y depende lo larga que sea o por donde pase esa raya significa, que es donde ese registro tiene nulas esas columnas.



Podemos analizar que la gran mayoría de los registros se distribuyen solamente en 3 columnas:img,workers y artist.

Pero podremos borrar todos los nulos?

¿Cuántas observaciones perdemos si eliminamos los datos faltantes?

```
df_without_nulls = (
    df
    .dropna()
)

df_without_nulls
```

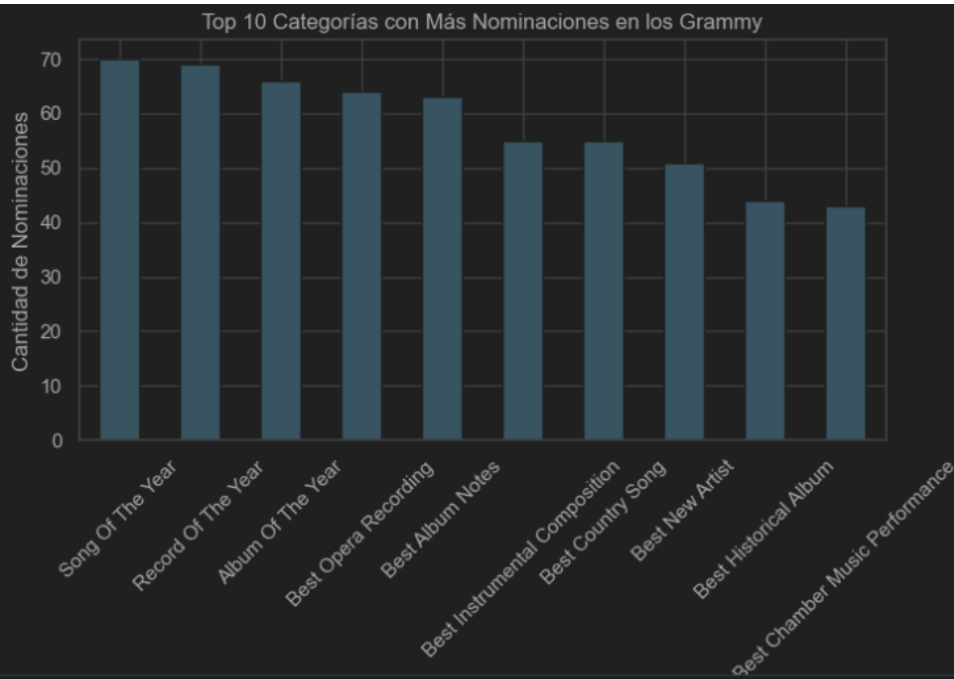
Executed at 2023.09.30 14:53:48 in 170ms

4449	1966	GRAMMY Awards (1966)	28T00:03:45-08:00	10T01:07:37-07:00	Record Of The Year	In The Night	Frank Sinatra	Bowen, producer	https://www.grammy.com/sites/com/files/sty	14	4	627
4450	1966	9th Annual GRAMMY Awards (1966)	2017-11-28T00:03:45-08:00	2019-09-10T01:07:37-07:00	Album Of The Year	A Man And His Music	Frank Sinatra	Sonny Burke, producer	https://www.grammy.com/sites/com/files/styles/...			True
4492	1965	8th Annual GRAMMY Awards (1965)	2017-11-28T00:03:45-08:00	2019-09-10T01:06:59-07:00	Record Of The Year	A Taste Of Honey	Herb Alpert And The Tijuana Brass	Herb Alpert & Jerry Moss, producers	https://www.grammy.com/sites/com/files/styles/...			True
4493	1965	8th Annual GRAMMY Awards (1965)	2017-11-28T00:03:45-08:00	2019-09-10T01:06:59-07:00	Album Of The Year	September Of My Years	Frank Sinatra	Sonny Burke, producer	https://www.grammy.com/sites/com/files/styles/...			True
4629	1962	5th Annual GRAMMY Awards (1962)	2017-11-28T00:03:45-08:00	2019-09-10T01:09:02-07:00	Album Of The Year (Other Than Classical)	The First Family	Vaughn Meader	Bob Booker & Earle Doud, producers	https://www.grammy.com/sites/com/files/styles/...			True

834 rows x 10 columns

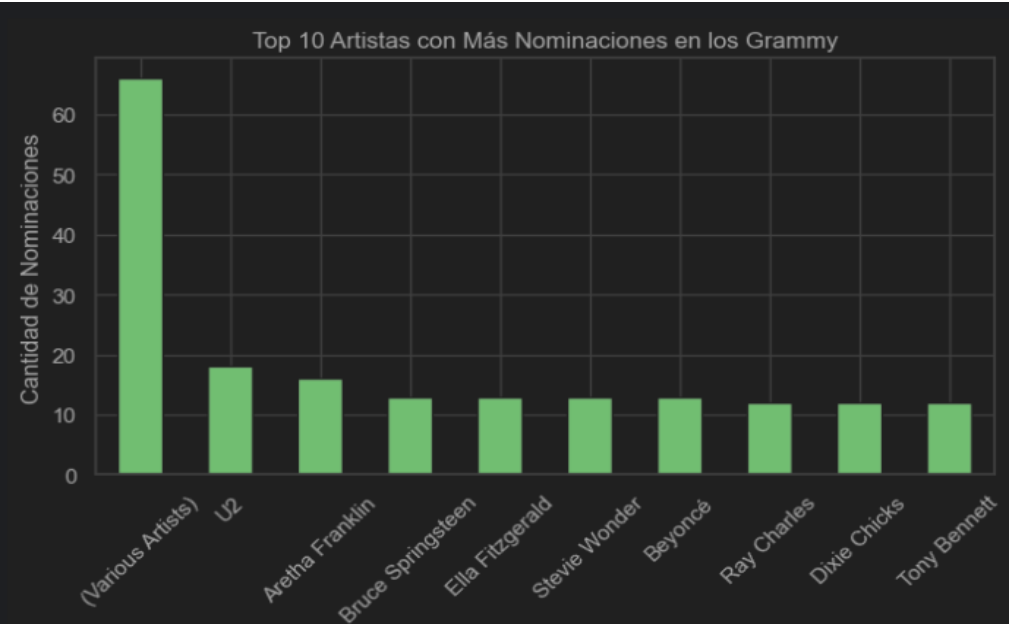
Miramos que quedaríamos con menos del 25% de nuestra informacion inicial, lo cual no es nada satisfactorio, entonces indagamos mas en las columnas que tienen nulos.

Top 10 Categorías con Mas Nominaciones:



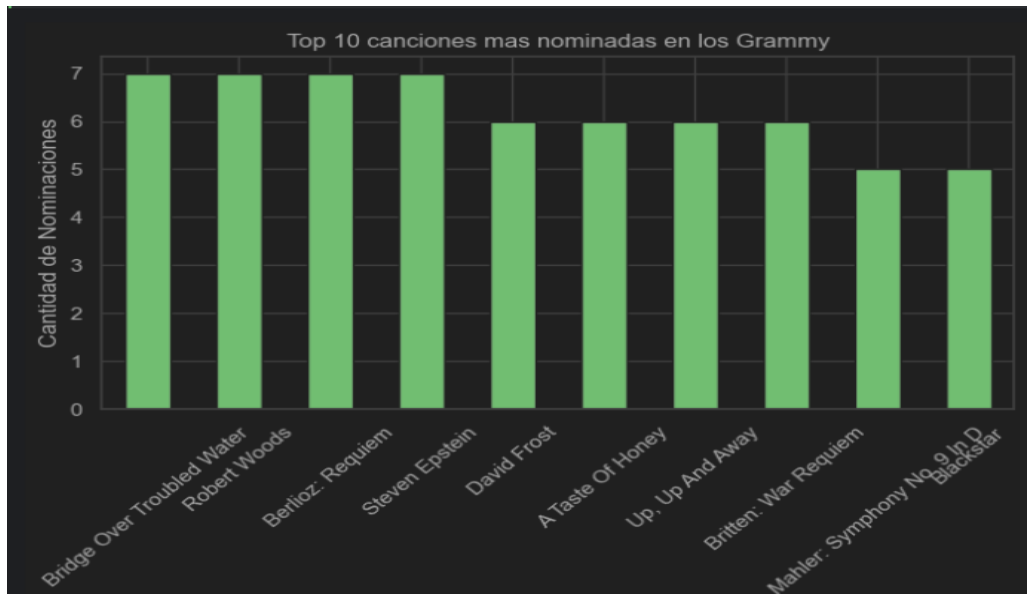
Podemos observar que las categorías Song, Record y Album Of The Year, son las categorías que mas se repiten, con una frecuencia de mas de 65 veces cada una, esto nos da a entender que todos los años almenos una vez se entrega este premio

Top 10 Artistas con Mas Nominaciones :



Se evidencia la existencia de un dato atipico , **Various Artists**. El cual no nos interesará analizar y será eliminado, ya que queremos analizar artistas en particular. Ademas podemos ver que los artistas con mas nominaciones (sacando el dato atipico) son **U2 ,Aretha Fraklin y Bruce Springsteen**.

Top 10 Artistas con Mas Nominaciones :



Se puede mirar un empate en el primer lugar ya que hay 4 canciones que tienen 7 nominaciones y son:**Bridge Over Troubled Water, Robert Woods, Beriloz Requiem y Steven Epstein**.

Transformaciones:

Para las Transformaciones tenemos el Archivo transfrom.py

Funciones: no_needed_columns , drop_duplicates, drop_null_rows,
Transform_winner_column

no_needed_columns

```
def no_needed_columns(df):  
    df.drop(['img', 'title', 'published_at', 'updated_at', 'workers'], axis=1, inplace=True)  
    return df
```

se eliminan img,workers,updated_at,published_at porque no nos aportan informacion relevante para nuestro analisis. title se eliminan porque se puede inferir de que año son los premios de los grammys con la columna year

Transform_winner_column

```
def Transform_winner_column(df):  
    df['winner'] = df['winner'].map({True: 1, False: 0})  
    return df
```

Entendiendo que True significa que ganaron los premios, pero al 100 % de la columna tiene True , nos hace inferir de que lo mas probable es que todos los que estan en el dataset ganaron, entonces cambiamos el valor True por 1.

Normalizamos las Columnas a Mayusculas , excepto winner y las numericas, para no tener problemas de Formato de las columnas.

drop_duplicates

```
def drop_duplicates(df):  
    df = df.drop_duplicates(keep='first')  
    return df
```

drop_null_rows

Verificamos si existen mas registros que empiecen con nombre REMIXER en la columna category
para validar si hay registros con el mismo nombre en la columna "category" y asi rellenoarlo, de no ser asi seran borrados

```
new_df = df[df["category"].str.startswith("REMIXER")]
new_df
```

Executed at 2023.09.30 16:36:20 in 176ms

#	year	title	published_at	updated_at	category
2261	2000	43RD ANNUAL GRAMMY AWARDS (2000)	2017-11-28T00:03:45-08:00	2019-09-10T01:11:09-07:00	REMIXER OF THE YEAR, NON-CLASSICAL
2359	1999	42ND ANNUAL GRAMMY AWARDS (1999)	2017-11-28T00:03:45-08:00	2019-09-10T01:09:02-07:00	REMIXER OF THE YEAR, NON-CLASSICAL
2454	1998	41ST ANNUAL GRAMMY AWARDS (1998)	2017-11-28T00:03:45-08:00	2019-09-10T01:08:19-07:00	REMIXER OF THE YEAR, NON-CLASSICAL
2547	1997	40TH ANNUAL GRAMMY AWARDS (1997)	2017-11-28T00:03:45-08:00	2019-09-10T01:07:37-07:00	REMIXER OF THE YEAR, NON-CLASSICAL

Como vemos no hay mas registros, entonces por eso son eliminados

Verificamos si existen mas registros que empiecen con nombre BEST NEW COUNTRY & en la columna category para validar si rellenamos o borramos estos registros

```
new_df = df[df["category"].str.startswith("BEST NEW COUNTRY &")]
new_df
```

Executed at 2023.09.30 16:41:07 in 246ms

#	year	title	published_at	updated_at	category
4525	1965	8TH ANNUAL GRAMMY AWARDS (1965)	2017-11-28T00:03:45-08:00	2019-09-10T01:06:59-07:00	BEST NEW COUNTRY & WESTERN ARTIST
4573	1964	7TH ANNUAL GRAMMY AWARDS (1964)	2017-11-28T00:03:45-08:00	2019-09-10T01:06:11-07:00	BEST NEW COUNTRY & WESTERN ARTIST OF

```
def drop_null_rows(df):
    df.drop([2261,2359,2454,2547,4525,4573], axis=0, inplace=True)
    return df
```

Eliminamos los registros ya que tienen columnas importantes para nuestro analisis vacias y ademas no se vio la forma de rellenarlos

def change_categories:

En esta funcion agrupamos las categorias , ya que existen mas de 600 categorias, las reducimos a 41, las cuales pueden ser analizadas mucho mas sencillo.

1.(explicando con ejemplo)

En el Caso de las categorias:

'Best Song Written For Visual Media',
 'Best Compilation Soundtrack For Visual Media',
 'Best Score Soundtrack For Visual Media',
 'Best Music Video',
 'Best Music Film'

Seran cambiadas su nombre por

Soundtracks/Music Videos y asi sucesivamente con todos.(Puede revisar el archivo categories para mirar todas las agrupaciones que se hicieron en esta columna)


```
categories = {
    'Soundtracks/Music Video': [
        'Best Song Written For Visual Media',
        'Best Compilation Soundtrack For Visual Media',
        'Best Score Soundtrack For Visual Media',
        'Best Music Video',
        'Best Music Film'
    ],
    'Production/Engineering': [
        'Best Instrumental Arrangement',
        'Best Arrangement, Instrumental or A Cappella',
        'Best Arrangement, Instruments and Vocals',
        'Best Recording Package',
        'Best Boxed Or Special Limited Edition Package',
    ]
}
```

Esta función **etiquetar_categoria** que está dentro de la función **def change_categories**, lo que nos ayuda es a recorrer la variable **categories** y con la función **apply**, agrupamos cada una de las variables explicadas en el **punto 1.(explicando con ejemplo)**

```
},}
def etiquetar_categoria(category):
    for clave, palabras_clave in categories.items():
        for palabra_clave in palabras_clave:
            if palabra_clave in category:
                return clave
    return category
df['category'] = df['category'].apply(etiquetar_categoria)
return df
```

Verificamos que después de usar la función se reduce a 41 los diferentes datos de la columna

```
valores_unicos_category1 = df["category"].nunique()
valores_unicos_category1
```

Executed at 2023.10.14 22:17:05 in 31ms

627

```
valores_unicos_category1 = df["category"].nunique()
valores_unicos_category1
```

Executed at 2023.10.14 22:11:13 in 62ms

41

Preguntas a Responder:

1. ¿Qué nominado ha recibido la mayor cantidad de premios en general o en un solo año?
2. ¿Qué obra ha recibido más premios en un evento?
3. ¿Qué país recibió la mayor cantidad de premios en una ceremonia y en general?
4. ¿Qué género musical ha ganado más premios?
5. ¿Cuál es el medio de pulsaciones por minuto de los ganadores?
6. ¿Qué género ha ganado más premios?

Configuracion Airflow

Ingresamos al contenedor por bash a traves del comando **docker exec -it <id_del_contenedor> bash**

```
C:\Users\Guatavo>docker exec -it d3c1b490a436 bash
airflow@d3c1b490a436:/opt/airflow$ ls
airflow-webserver.pid  airflow.cfg  config  dags  logs  plugins  webserver_config.py
```

Hacemos el comando para crear un usuario y su contraseña

+

```
airflow@d3c1b490a436:/opt/airflow$ airflow users create -r Admin -u gustavo316 -e lionester316@gmail.com -f gustavo -l manyoma -p Autonomia316
/home/airflow/.local/lib/python3.8/site-packages/flask_limiter/extension.py:336 UserWarning: Using the in-memory storage for tracking rate limits as no storage was explicitly specified. This is not recommended for production use. See: https://flask-limiter.readthedocs.io#configuring-a-storage-backend for documentation about configuring the storage backend.
[2023-10-12T04:37:30.072+0000] {manager.py:211} INFO - Added user %s
User "gustavo316" created with role "Admin"
airflow@d3c1b490a436:/opt/airflow$
```

EDA SPOTIFY

Solo 3 columnas tienen datos nulos, artists, album_name, track_name, y cada una tiene un dato nulo.

De tener observaciones con valores nulos, ¿cuántas tenemos por cada variable?

```
1 (
2   df
3   .isnull()
4   .sum()
5   .sort_values(ascending=False)
6 )
7
```

Executed at 2023.10.12 12:22:05 in 141ms

Length: 21, dtype: int64 pd.Series

	<unnamed>
artists	1
album_name	1
track_name	1
Unnamed: 0	0
mode	0
time_signature	0

Valence: Medida de 0,0 a 1,0 que describe la positividad musical que transmite una pista. Las pistas con una valencia alta suenan más positivas (por ejemplo, felices, alegres, eufóricas), mientras que las pistas con una valencia baja suenan más negativas (por ejemplo, tristes, deprimidas, enfadadas).

track_genre: The genre in which the track belongs

Load :

Podemos ver de que el merge fue subido a la base de datos

```

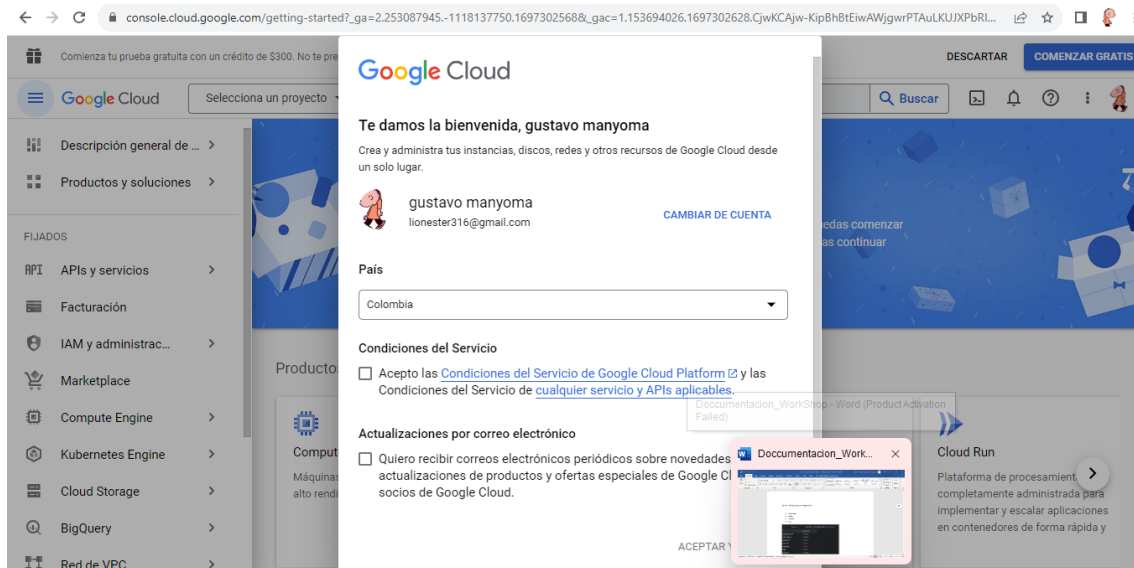
[4810 rows x 10 columns]
      track_id      artists ...      artist winner
0  6Vc5wAMmXdKIAM7WUoEb7N  A Great Big World;Christina Aguilera ... A Great Big World With Christina Aguilera 1
1  5TvE3pk05pyFIGdSY9j4DJ  A Great Big World;Christina Aguilera ... A Great Big World With Christina Aguilera 1
2  78TKt1SLwK8pZAKKw3MyQL  A Great Big World;Christina Aguilera ... A Great Big World With Christina Aguilera 1
3  7xLhousIHDxoGgeJNh04Ye      A Great Big World ... A Great Big World With Christina Aguilera 1
4  6kRZF0ZSuoioHtg6gR0KDz      A Great Big World ... A Great Big World With Christina Aguilera 1
...      ...      ...      ...      ...
3182  61h02ZrTkc0XNt8eBIMJX9      Phil Wickham ...      Tim Menzies 1
3183  7d1CAK8HdnI6UGHqZkFvJT      for KING & COUNTRY ...      NaN 1
3184  5B1fUGBm5LF3V9U1AYpLpf      Hillsong Worship ...      Steven Curtis Chapman 1
3185  4iFGj9LMN5xYuGeDcNifo9      Jadon Lavik ...      Larry Hart 1
3186  3jId15LURV3fJNmV5nDS74      Hillsong Worship ...      Al Jarreau 1

[3187 rows x 31 columns]
Engine cerrado

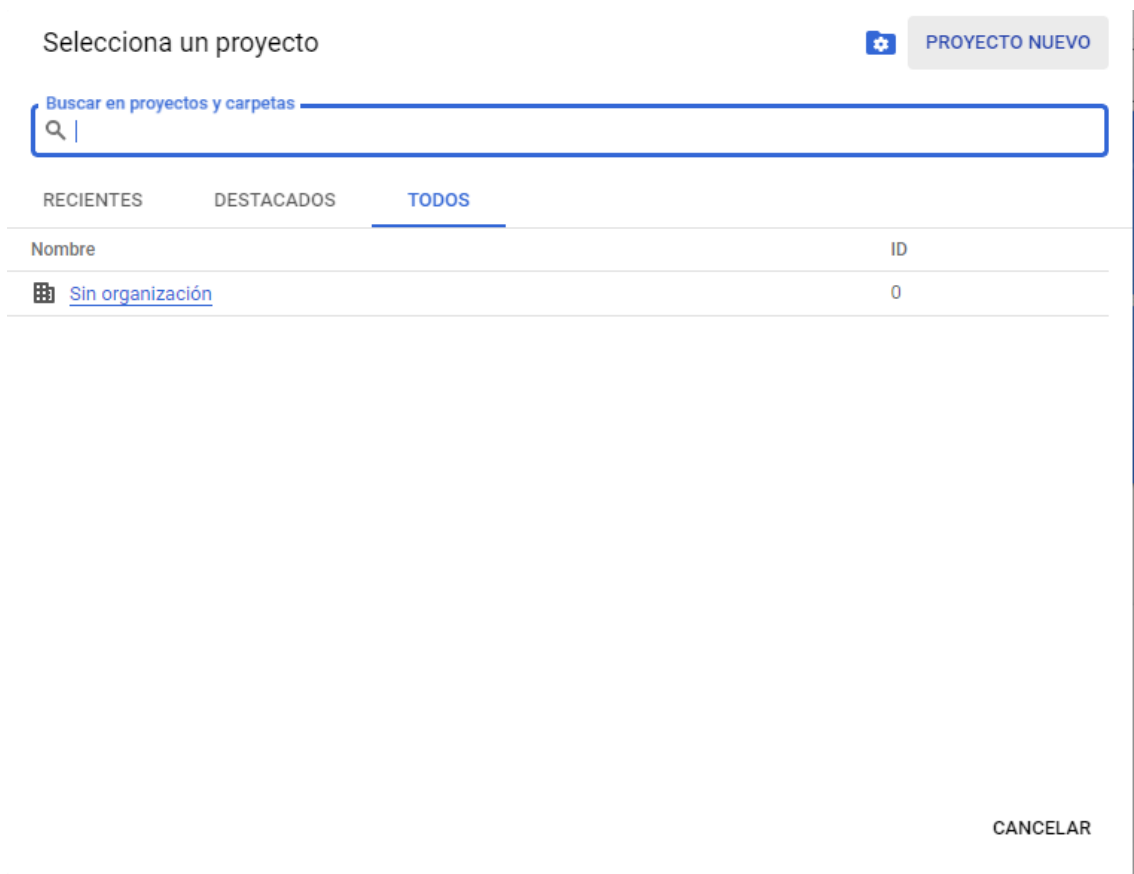
```

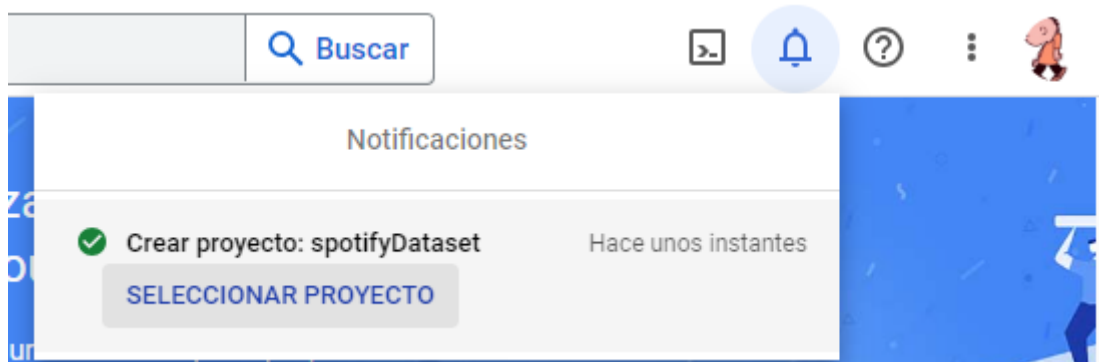
Google Drive

Para poder subir nuestro archivo merged.csv a traves de codigo, tenemos que usar la plataforma de google cloud para poder hacerlo, entonces creamos un usuario y nos logueamos, aceptando terminos y condiciones.

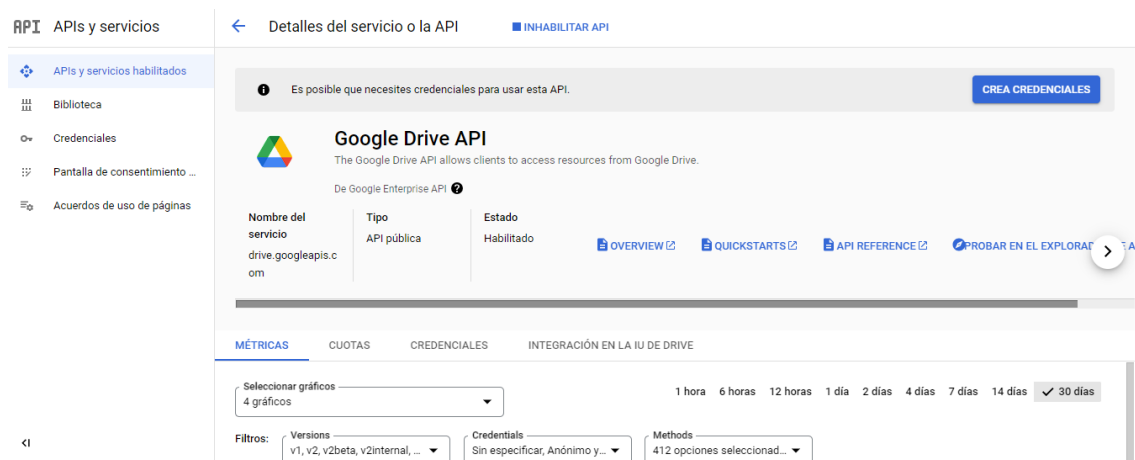


Debemos crear un nuevo proyecto, el cual llamamos **spotifyDataset**

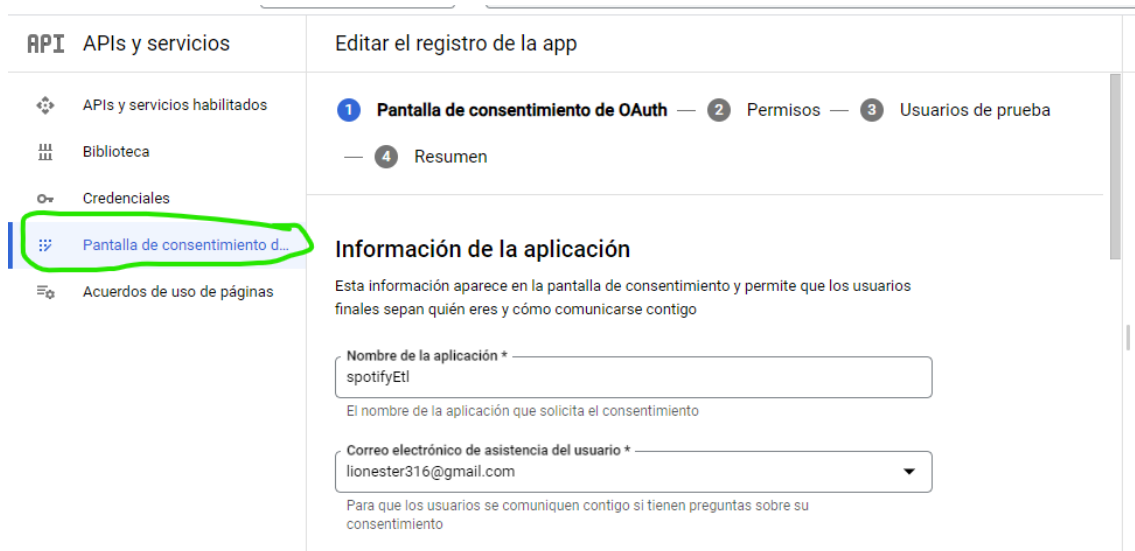




Buscamos dentro de GCP Google Drive API



En este caso usaremos OAuth y registramos nuestra app en la Pantalla de Consentimiento.



Copiamos el link de la barra de búsqueda y lo ponemos en esta seccion:

Dominio de la app

Para protegerlos a ti y a tus usuarios, Google solo permite que las apps que usan OAuth puedan emplear los dominios autorizados. Se mostrará la siguiente información a los usuarios en la pantalla de consentimiento.

Página principal de la aplicación

<https://console.cloud.google.com/apis/credentials/consent/edit;newAppInternalUser>

Proporciona a los usuarios un vínculo a tu página principal

Vínculo a la Política de Privacidad de la aplicación

<https://console.cloud.google.com/apis/credentials/consent/edit;newAppInternalUser>

Proporciona a los usuarios un vínculo a tu página pública de Política de Privacidad

Vínculo a las Condiciones del Servicio de la aplicación

<s/credentials/consent/edit;newAppInternalUser=false?project=spotifydataset-402017>

Proporciona a los usuarios un vínculo a tu página pública de Condiciones del Servicio

Colocamos nuestro correo y le damos Guardar y continuar

API	APIs y servicios	Editar el registro de la app
	APIs y servicios habilitados	Proporciona a los usuarios un vínculo a tu página pública de Condiciones del Servicio
	Biblioteca	
	Credenciales	
	Pantalla de consentimiento d...	
	Acuerdos de uso de páginas	
		Dominios autorizados ? Cuando un dominio se usa en la pantalla de consentimiento o en la configuración del cliente de OAuth, debe contar con un registro previo aquí. Si debes verificar la app, ve Google Search Console para comprobar si tus dominios están autorizados. Más información sobre el límite de dominios autorizados. <div>Dominio autorizado 1 * google.com</div> <div>+ AGREGAR UN DOMINIO</div>
		Información de contacto del desarrollador <div>Direcciones de correo electrónico * lionester316@gmail.com</div> <p>Google enviará notificaciones sobre cualquier cambio en tu proyecto a estas direcciones de correo electrónico.</p> <div>GUARDAR Y CONTINUAR CANCELAR</div>

- ✓ Pantalla de consentimiento de OAuth — 2 **Permisos** — 3 Usuarios de prueba —
4 Resumen

En esta seccion de **Permisos** no configuramos nada entonces le damos siguiente

En la seccion de usuarios de prueba ponemos nuestro correo:

X

Agregar usuarios

Mientras el estado de publicación sea "Prueba", solo los usuarios de prueba podrán acceder a la app. El límite de usuarios permitidos antes de que se verifique la app es de 100, y se calcula según el ciclo de vida completo de la app.

LEARN MORE

lionester316@gmail.com

?

1 / 100

AGREGAR

Filtro

Ingresar el nombre o el valor de la propiedad

?

Información del usuario	
lionester316@gmail.com	

GUARDAR Y CONTINUAR

CANCELAR

Y asi quedaron las Configuraciones:

Editar el registro de la app

Nombre de la app

spotifyEtl

Correo electrónico de asistencia

lionester316@gmail.com

Logotipo de la app

No se proporcionó

Vínculo a la página principal de la aplicación

https://console.cloud.google.com/apis/credentials/consent/edit;newAppInternalUser=false?project=spotifydataset-402017

Vínculo a la Política de Privacidad de la aplicación

https://console.cloud.google.com/apis/credentials/consent/edit;newAppInternalUser=false?project=spotifydataset-402017

Vínculo a las Condiciones del Servicio de la aplicación

https://console.cloud.google.com/apis/credentials/consent/edit;newAppInternalUser=false?project=spotifydataset-402017

Ahora procedemos a la Creacion de las Credenciales, Nos conectaemos atraves de el ID de cliente de OAuth.

RPI

APIs y servicios

APIs y servicios habilitados

Biblioteca

Credenciales

Pantalla de consentimiento ...

Acuerdos de uso de páginas

Credenciales

+ CREAR CREDENCIALES

🗑 BORRAR

↩ RESTABLECER CREDENCIALES BORRADAS

Crea credenciales para acceso

Clave de API

ID de cliente de OAuth

Cuenta de servicio

Claves de API

☐ Nombre

No hay claves de API para mostrar

ID de clientes OAuth

☐ Nombre

No hay clientes de OAuth para mostrar

Cuentas de servicio

☐ Correo electrónico

No hay cuentas de servicio para mostrar

Acciones

Acciones

Acciones

Acciones

Empezamos con las Configuraciones:

←

Crear ID de cliente 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 *

spotifyDataset

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.

i

Los dominios de los URI que agregues a continuación se incorporarán automáticamente a tu [pantalla de consentimiento de OAuth](#) como [dominios autorizados](#).

Orígenes autorizados de JavaScript

?

Para usar con solicitudes de un navegador

URI 1 *

http://localhost:1080

CORRECCION: En la URL es <http://localhost:8080>

Le damos crear y descargamos el JSON.

Se creó el cliente de OAuth

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

i

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

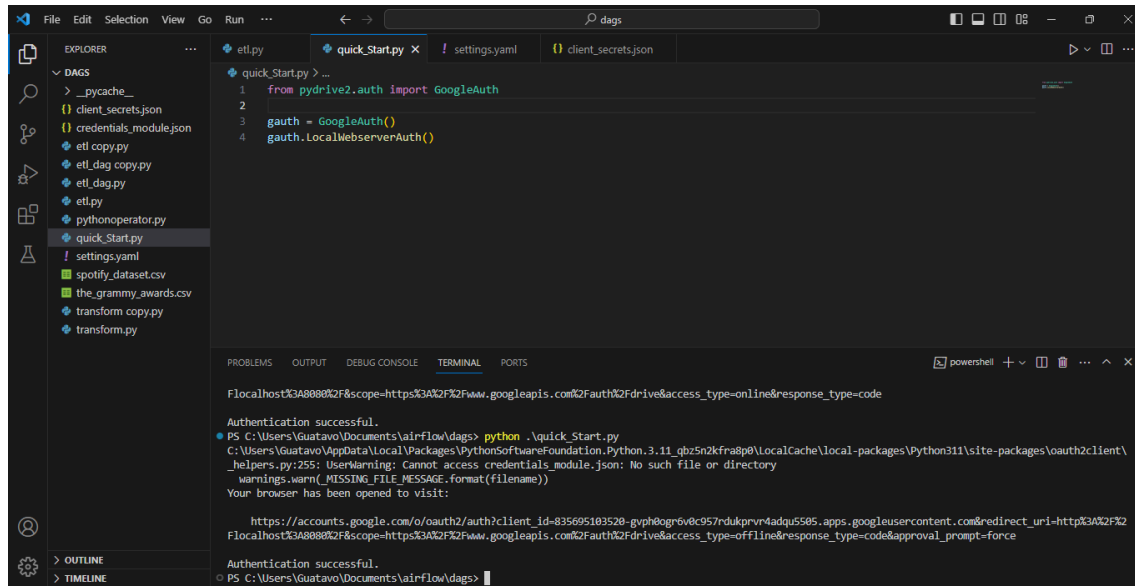
ID de cliente	835695103520-gvph0ogr6v0c957rdukprvr4adqu5505.apps.googleusercontent.com
Secreto del cliente	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
Fecha de creación	14 de octubre de 2023, 12:11:01 GMT-5
Estado	<div>✓</div> Habilitada

↓

 DESCARGAR JSON

ACEPTAR

Creamos un archivo **quick_Start.py** el cual nos permitira autenticarnos.

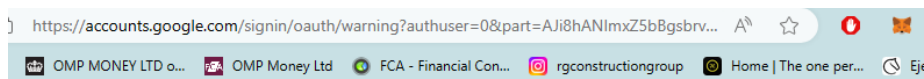


The screenshot shows a VS Code editor with a file explorer on the left. The file explorer shows a project named 'dags' with several files including 'quick_Start.py'. The main editor window shows the content of 'quick_Start.py' with the following code:

```
1 from pydrive2.auth import GoogleAuth
2
3 gauth = GoogleAuth()
4 gauth.LocalWebserverAuth()
```

Below the editor, a terminal window is open, showing the output of running the script. It displays a long URL for authentication, followed by 'Authentication successful.' and a warning message about a missing file. The terminal also shows the command 'python quick_Start.py' being executed.

Corremos el archivo, nos lleva a la web donde le damos continuar



nos autenticamos iniciando sesión.

spotifyEtl quiere acceder a tu cuenta de Google



lionester316@gmail.com

Cuando permitas este acceso, **spotifyEtl** podrá:



Ver, modificar, crear y eliminar archivos de Google Drive. [Más información](#)

Confirma que confías en spotifyEtl

Puede que estés compartiendo información sensible con este sitio o esta aplicación. Puedes ver o retirar el acceso en cualquier momento en tu [cuenta de Google](#).

Descubre cómo te ayuda Google a [compartir datos de forma segura](#).

Consulta la [Política de Privacidad](#) y los [Términos del Servicio](#) de spotifyEtl.

Cancelar

Continuar

The authentication flow has completed.

Luego hacemos unas configuraciones para crear persistencia, es decir para que no nos pida autenticarnos cada vez que queramos subir un archivo. Debemos recordar que El `client_id` y el `client_secret` los sacamos de la app que creamos en GCP

```
! settings.yaml
1 client_config_backend: settings
2 client_config:
3   client_id: 835695103520-gvph0ogr6v0c957rdukprvr4adqu5505.apps.googleusercontent.com
4   client_secret: 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
5
6 save_credentials: True
7 save_credentials_backend: file
8 save_credentials_file: credentials_module.json
9
10 get_refresh_token: True
11
12 oauth_scope:
13   - https://www.googleapis.com/auth/drive
```

Corremos el archivo `load_to_Drive.py` que tiene 2 funciones, `login()` que nos permite loguearnos y crear la conexión con Google Drive y `subir_archivo()` que sube el archivo recibiendo los parametros (`ruta_del_archivo_a_subir`, `id_de_la_carpeta`)

```
23
24 # Carga de archivos
25 def subir_archivo(ruta_archivo, id_folder):
26     credenciales = login()
27     archivo = credenciales.CreateFile({'parents': [{"kind": "drive#fileLink", \
28                                               "id": id_folder}]})
29     archivo['title'] = ruta_archivo.split("/")[-1]
30     archivo.SetContentFile(ruta_archivo)
31     archivo.Upload()
32
33
34 if __name__ == "__main__":
35     ruta="C:/Users/Guatavo/Documents/airflow/dags/merged_data.csv"
36     subir_archivo(ruta, "1WZbU6FwuU5SIMST5uDKXMY-VFXWPNS9n")
```

https://accounts.google.com/o/oauth2/auth?client_id=835695103520-gvph0ogr6v0c957rdukprvr4adqu5505.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8080%2F&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive&access_type=offline&response_type=code&approval_prompt=force

Authentication successful.

PS C:\Users\Guatavo\Documents\airflow\dags> pwd

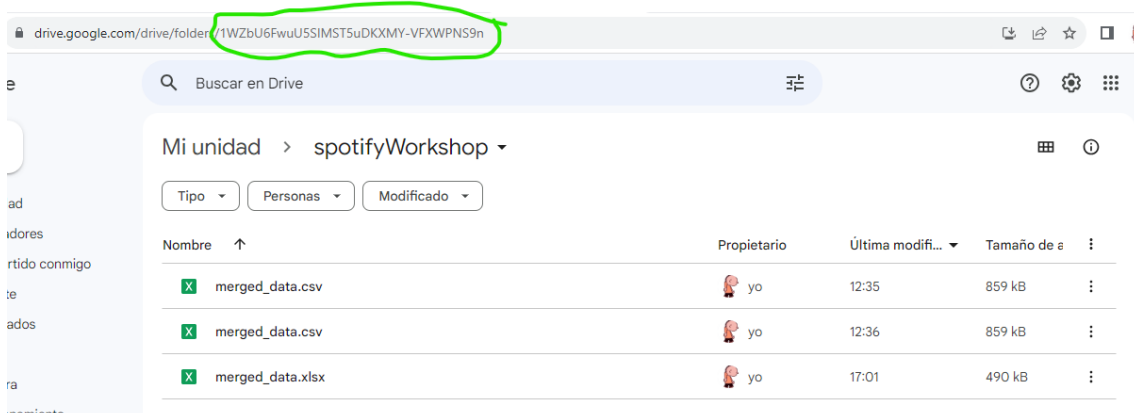
Path

C:\Users\Guatavo\Documents\airflow\dags

PS C:\Users\Guatavo\Documents\airflow\dags> python .\load_To_Drive.py

PS C:\Users\Guatavo\Documents\airflow\dags>

Corremos el archivo, y efectivamente podemos ver de que se subio el archivo.



Lo que esta en verde es el `id_de_la_carpeta`.