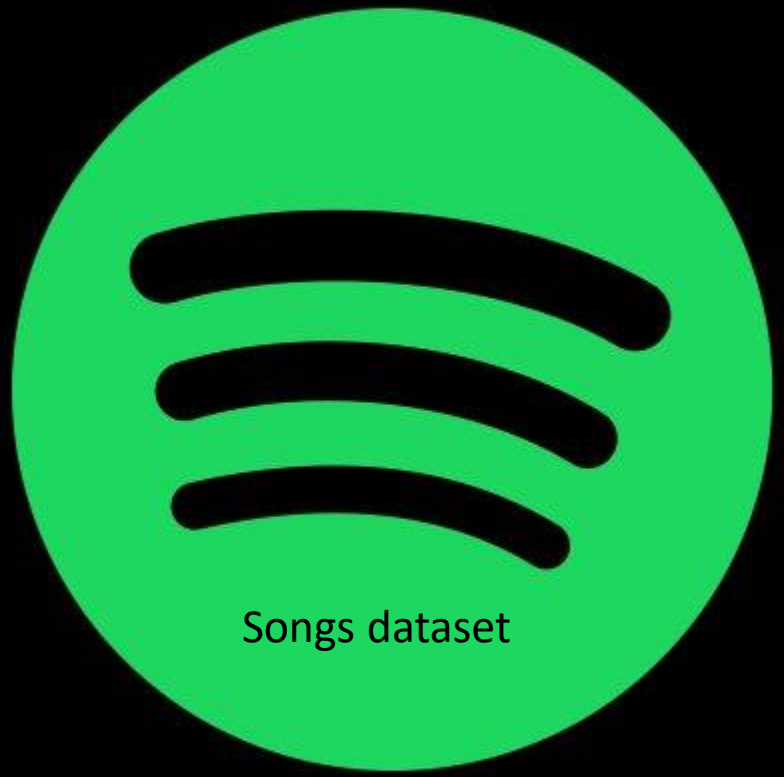
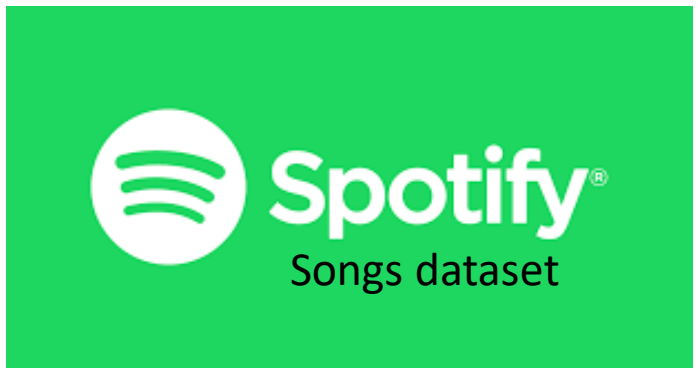


Songs dataset

Spotify

Tarea #1 curso Bigdata Expert

Richard Douglas G.



Songs dataset

- Un dataset compuesto por un listado de canciones según algunas características propias de las canciones, como lo son acousticness, artists, danceability, duration_ms, energy, name, liveness, loudness, popularity, entre otros
- Entre los años 1920 a 2021

```
# Importacion de bibliotecas

import argparse # Biblioteca para crear enlaces entre interfaces y mantener estructuras de datos
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler # Normalizador de datos
from sklearn.preprocessing import MinMaxScaler # Normalizador de minimos
from sklearn.decomposition import PCA # Biblioteca de componentes principales
from sklearn.cluster import KMeans # K-medias
import seaborn as sns
from kneed import KneeLocator # Biblioteca que permite pintar puntos de inflexion en funciones (Elbow Method)
import plotly.graph_objects as go
from plotly.subplots import make_subplots
sns.set()
```

```
data.info()  # tenemos caracteristicas numéricas y de tipo objeto. se debe proceder a determinar cuales se deben usar
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 174389 entries, 0 to 174388
Data columns (total 19 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   acousticness      174389 non-null  float64
 1   artists           174389 non-null  object 
 2   danceability      174389 non-null  float64
 3   duration_ms       174389 non-null  int64  
 4   energy            174389 non-null  float64
 5   explicit          174389 non-null  int64  
 6   id                174389 non-null  object 
 7   instrumentalness  174389 non-null  float64
 8   key               174389 non-null  int64  
 9   liveness          174389 non-null  float64
 10  loudness          174389 non-null  float64
 11  mode              174389 non-null  int64  
 12  name              174389 non-null  object 
 13  popularity        174389 non-null  int64  
 14  release_date      174389 non-null  object 
 15  speechiness       174389 non-null  float64
 16  tempo             174389 non-null  float64
 17  valence           174389 non-null  float64
 18  year              174389 non-null  int64  
dtypes: float64(9), int64(6), object(4)
memory usage: 25.3+ MB
```

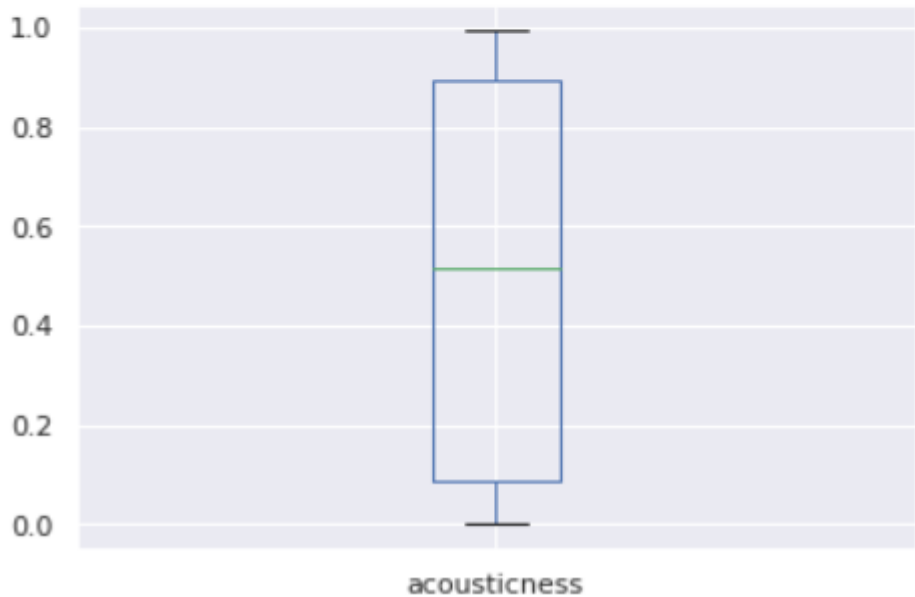
- Un dataset compuesto por un total de
- 174389 rows (observaciones) × 19 columns (características)

EDA

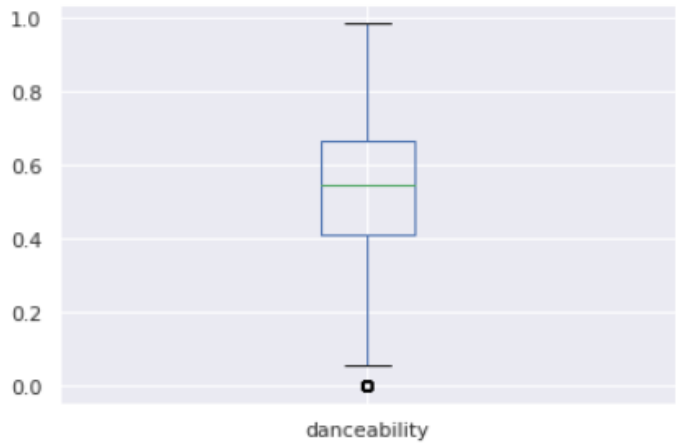
Analisis Exploratorio (EDA) de las Caracteristicas

1.Analisis Exploratorio de la Caracteristica **acousticness**

```
[23] datos ['acousticness'].plot(kind='box')  # la mediana es de alrededor de 0.5, el primer quartil (25%) corresponde a 0.0877
     plt.show()                               # no se observan valores atipicos.
```



```
[39] datos ['danceability'].plot(kind='box')          # se realiza un gráfico de  boxplot  la mediana ronda los 0.54
     plt.show()                                      # el valor mínimo sobre cero y el tercer quartil a 0.66
```




```
[40] datos['danceability'].quantile(0.25)     # primer quartil indica que un 25% de los datos es menor o igual a 0.414 en la caracteristica danceability

     0.414
```

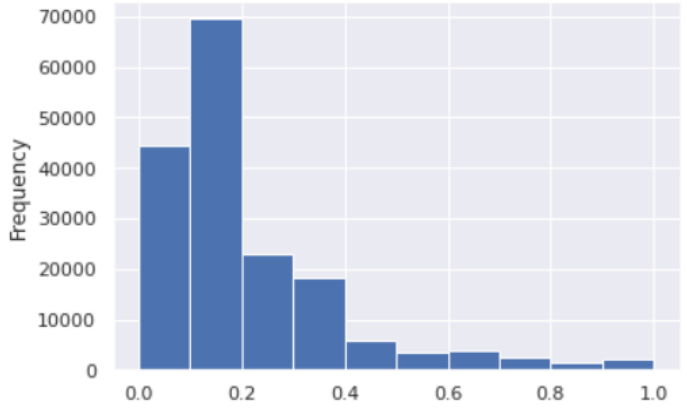
```
[41] datos['danceability'].quantile(0.5)      # el segundo quartil indica que el 50% de las canciones presenta un valor de danceability menor o igual a 0.5479

     0.5479999999999999
```

Se realizan los histogramas de las características para observar el comportamiento de las distribuciones

Característica con distribución con comportamiento positivo

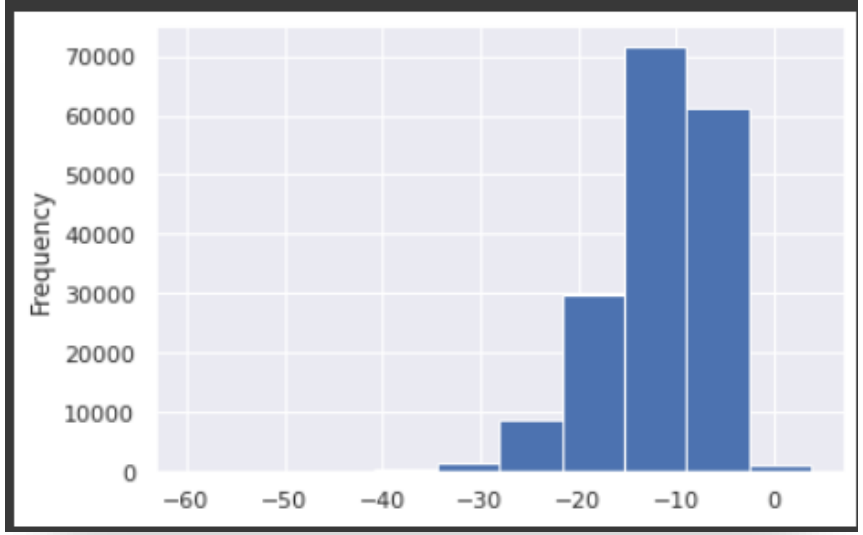
Histograma de la caracteristica **liveness**

```
[ ]  datos['liveness'].plot.hist();
```



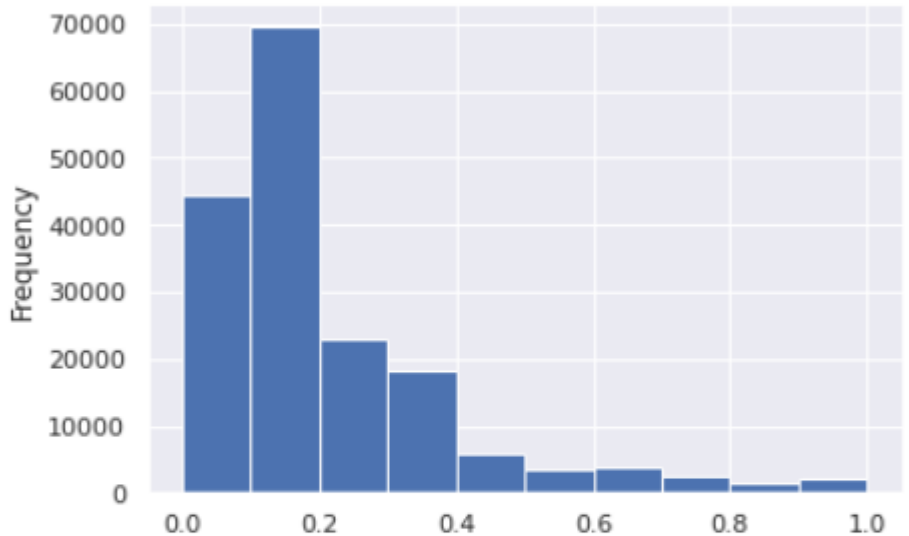
Característica con distribución con comportamiento negativo

```
datos['loudness'].plot.hist();
```



Histograma de la caracteristica **liveness**

```
[102] datos['liveness'].plot.hist();   # presenta una distribución con comportamiento positivo (agrupamiento hacia la izquierda)
```



Medidas de tendencia central y otros valores de relevancia en el análisis

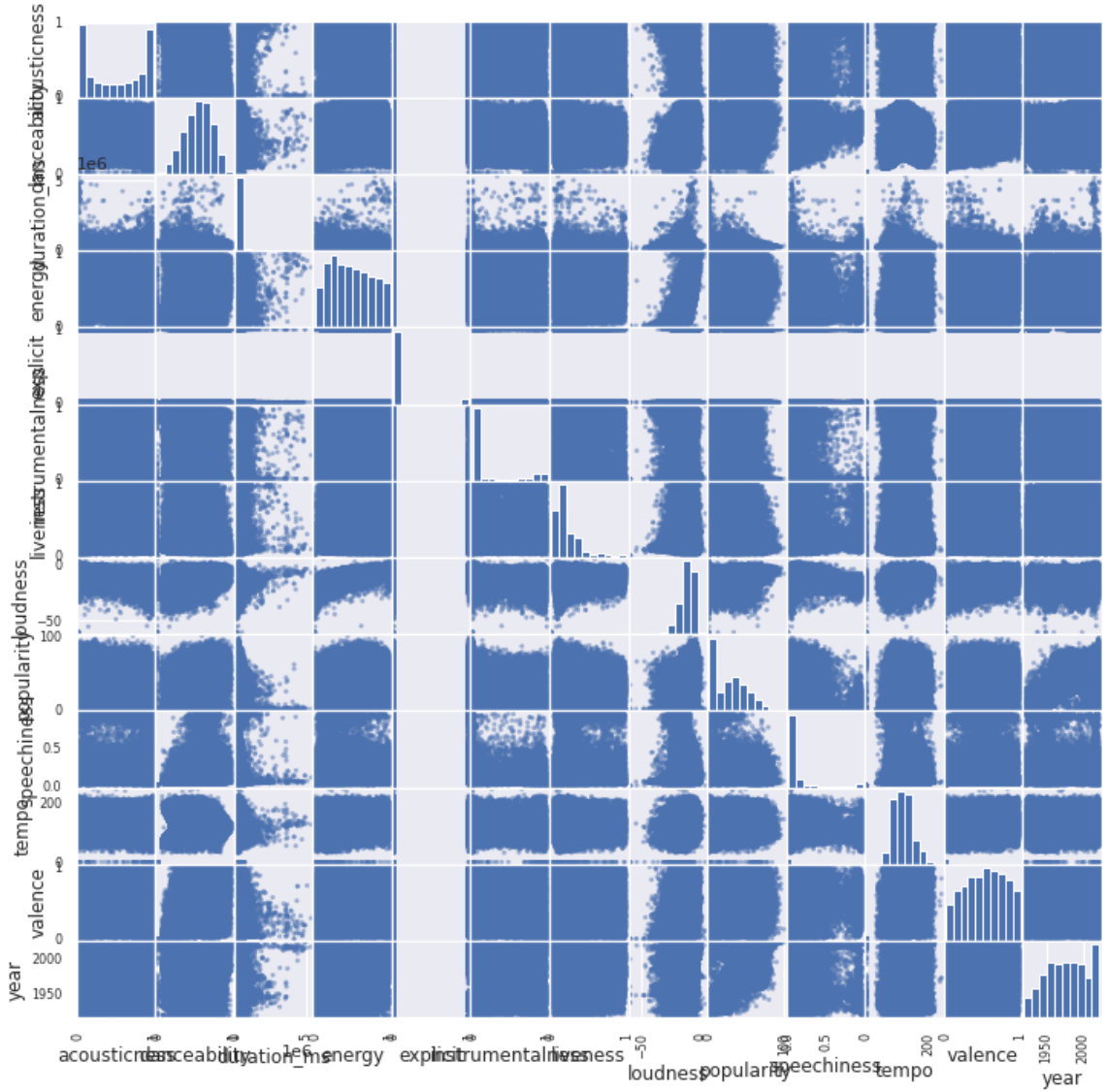
```
[103] datos['liveness'].min()    # el valor más bajo que aparece es de cero 0
```

```
0.0
```

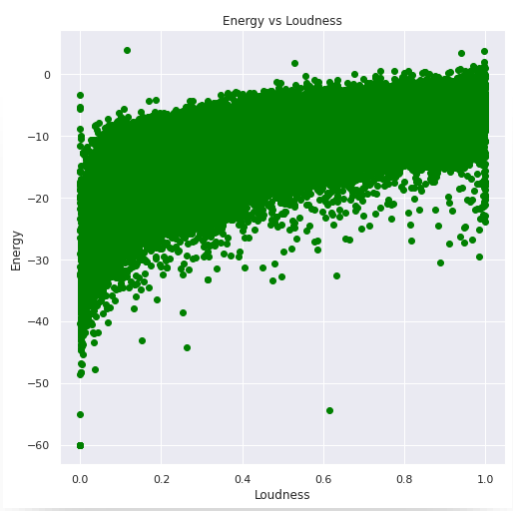
```
[104] datos['liveness'].max()  # el valor más alto es de 1
```

Gráficas y Correlaciones

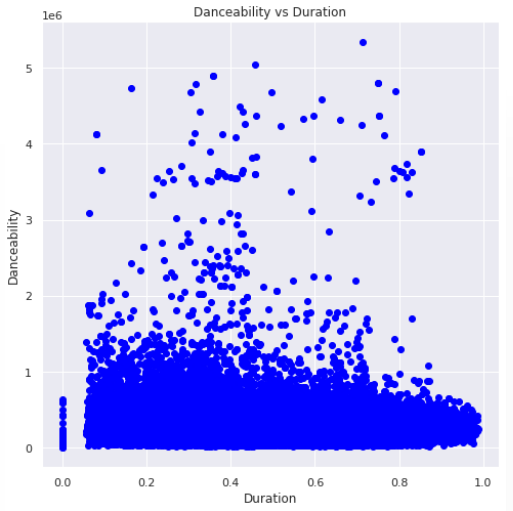
Grafica de tipo Scatter Matrix



Correlación positiva

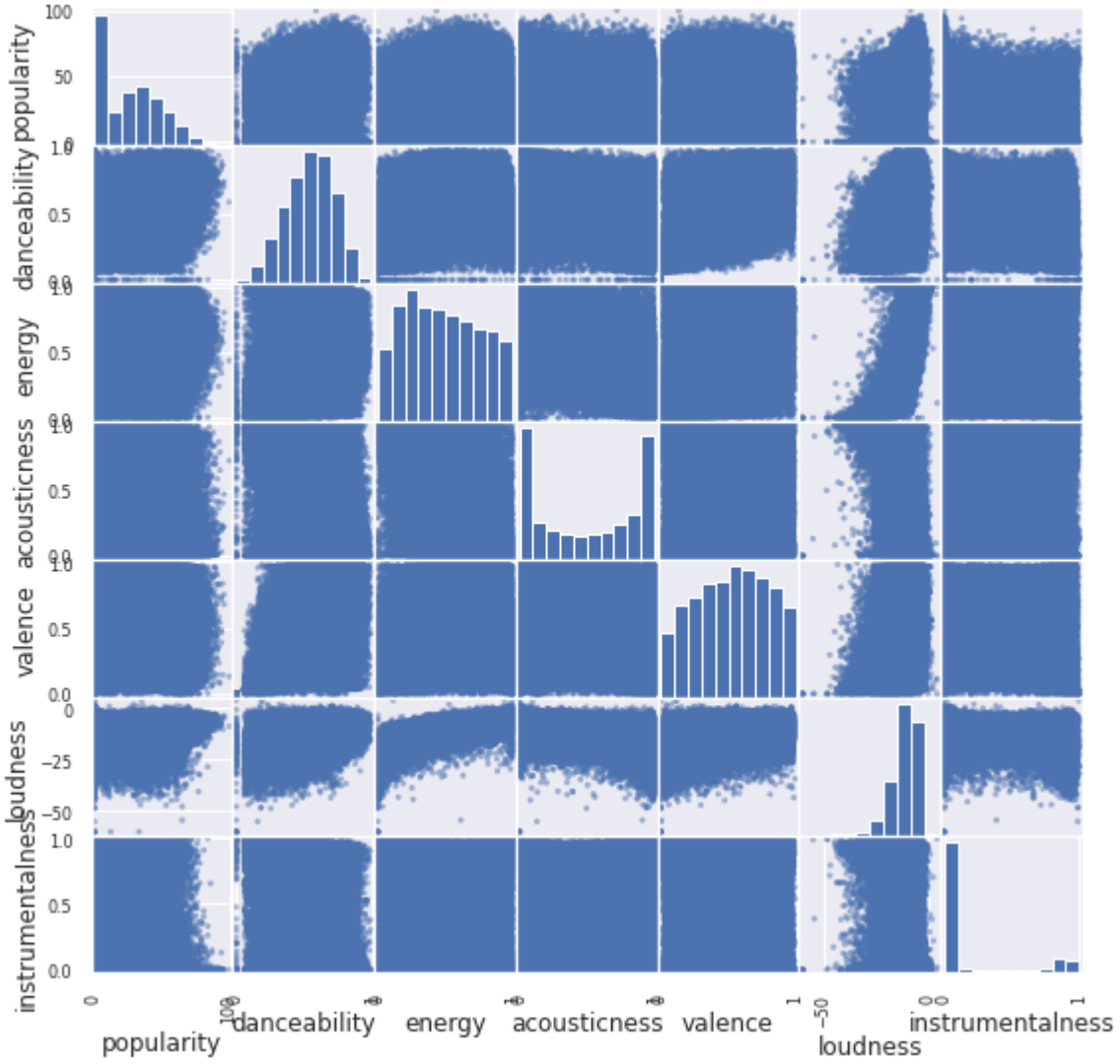


No correlación

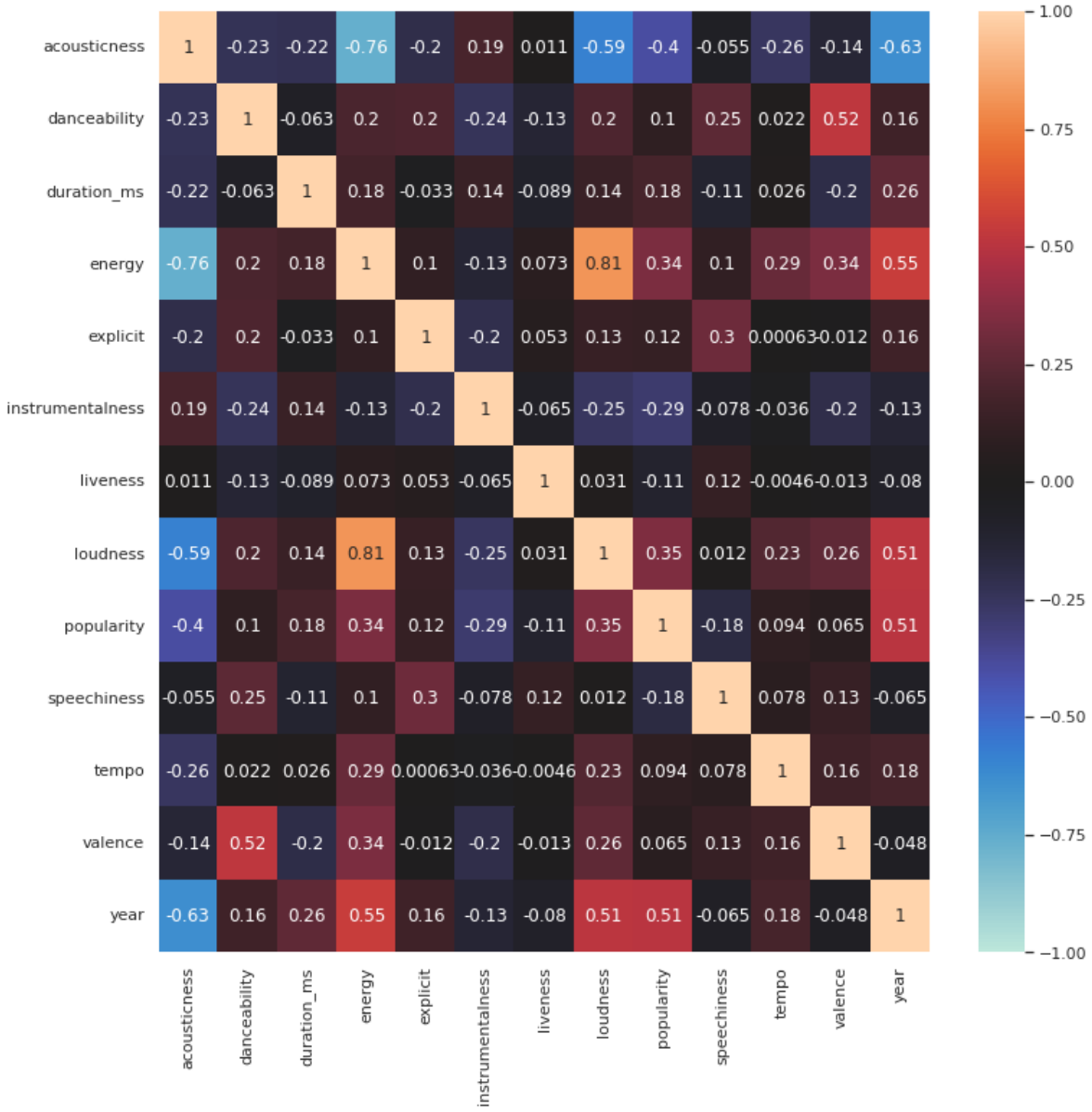


- Aquellos que presentan disperción en los puntos indica que no existe correlación.

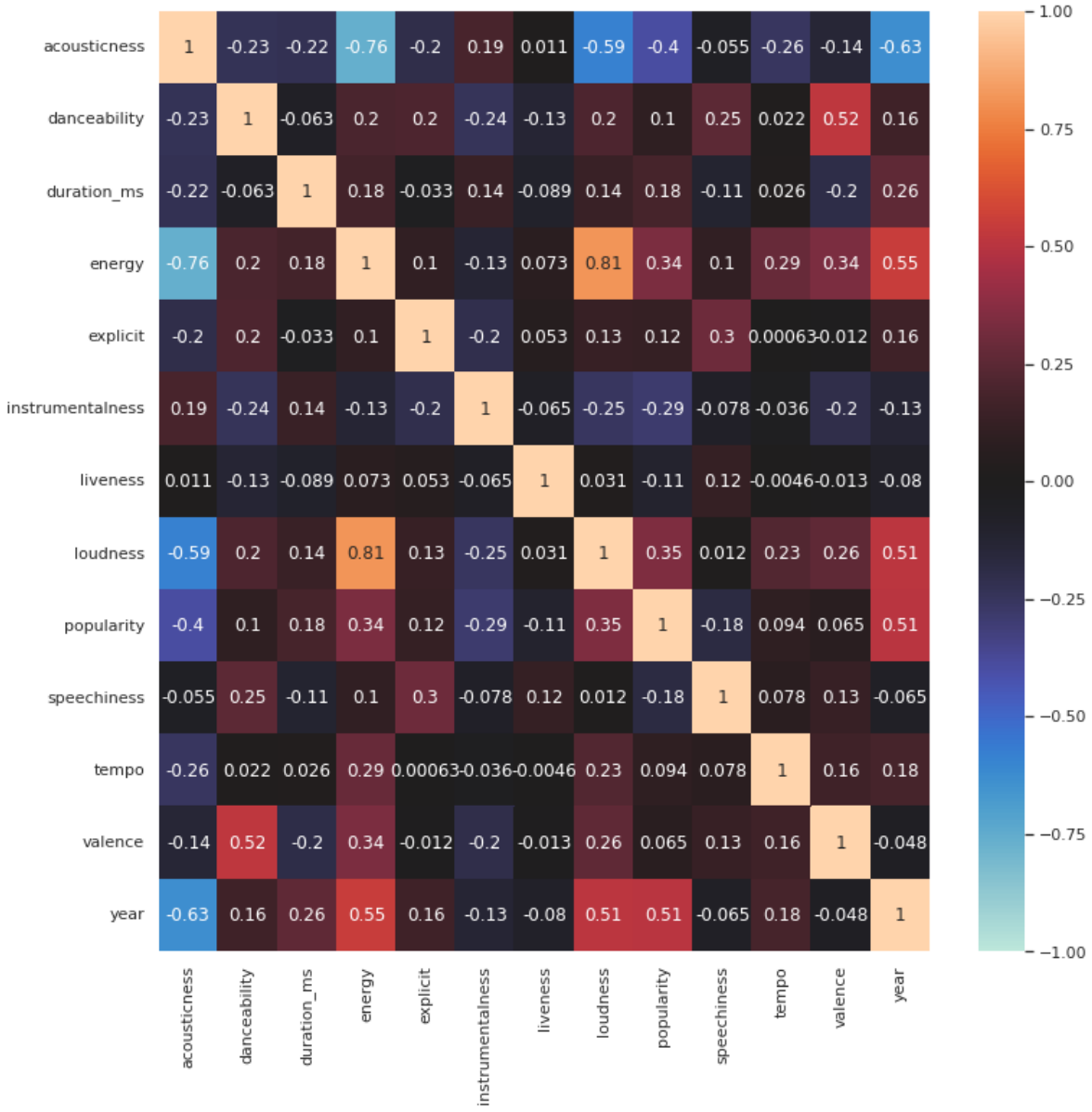
- Aquellos que presentan un grupo definido presentan correlación negativa o positiva



- Correlación positivafuerte entre loudness y energy
- Correlación positiva fuerte entre popularity y year
- Correlación Negativa fuerte entre acousticness y energy.



- Existe una relación negativa fuerte entre **acousticnes** y **energy** para un valor de -
  0.76 lo que indica que si aumenta una disminuye la otra
- Existe una relación negativa fuerte entre **acousticness** y ****Loudness**** para un valor de -
  0.59 lo que indica que si aumenta una disminuye la otra
- Existe una relación negativa fuerte entre **acousticness** y **year** para un valor de -
  0.63 lo que indica que si aumenta una disminuye la otra



- La caracteristica **explicit** presenta una leve relación con la caracteristica **speechiness** con un valor de 0.3, si incrementa una puede presentar un incremento en la otra
- La caracteristica presenta **explicit** una relación negativa leve con las caracteristicas **acousticness** y **instrumentalness** con un valor de -0.2 con ambas caracteristicas.
- la caracteristica **liveness** presenta poca o casi nula en las correlaciones con las demás caracteristicas. con la que presenta el valor mayor sería con **danceability** con un valor negativo de -0.13

- La caracteristica **loudness** presenta una correlación positiva alta con la caracteristica **energy** con un valor de 0.81 si aumenta una la otra tambien aumenta
- **popularity** presenta una correlación positiva alta con la caracteristica **year** por lo que si aumenta una la otra también aumentará. Además la carcteristica **popularity** presenta una correlación negativa con la caractistica **acousticness** con un valor de -0.4 si aumenta una la otra va a disminuir.

```
popularidad.head(20)
```

	artists	name	popularity	year
20062	Olivia Rodrigo	drivers license	100	2021
19862	24kGoldn, iann dior	Mood (feat. iann dior)	96	2020
19866	Ariana Grande	positions	96	2020
19886	Bad Bunny, Jhay Cortez	DÁKITI	95	2020
19976	KAROL G	BICHOTA	95	2020
19868	Ariana Grande	34+35	94	2020
19870	CJ	Whoopty	94	2020
19872	The Kid LAROI	WITHOUT YOU	94	2020
19876	Billie Eilish	Therefore I Am	94	2020
19928	Bad Bunny, ROSALÍA	LA NOCHE DE ANOCHE	94	2020
19900	Tate McRae	you broke me first	93	2020
19878	Pop Smoke	What You Know Bout Love	93	2020
39252	Tiësto	The Business	92	2020
76406	Boza	Hecha Pa' Mi	92	2020
19884	Lil Nas X	HOLIDAY	92	2020
19880	Cardi B, Megan Thee Stallion	WAP (feat. Megan Thee Stallion)	92	2020
19908	Justin Bieber, benny blanco	Lonely (with benny blanco)	92	2020
20068	Justin Bieber	Anyone	92	2021
19924	Shawn Mendes, Justin Bieber	Monster (Shawn Mendes & Justin Bieber)	91	2020
19864	SZA	Good Days	91	2020

Top 20 de las canciones más populares y sus Interpretes

```
[ ] artistas_canciones = datos2.iloc[:,[1,12,13,18]]
```

```
[ ] popularidad = artistas_canciones.sort_values('popularity',ascending=False)
```

```
[ ] popularidad.head(20)
```

```
text4 = popularidad.name.head(20).values
wordcloud = WordCloud().generate(str(text4))
plt.figure( figsize=(10,10), facecolor='k')
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

```
[224] acoustic.head(20)
```

	artists	name	acousticness	year
40396	Ignacio Corsini	Mentiras - Remasterizado	0.996	1930
23546	Frédéric Chopin, Claudio Arrau	24 Préludes, Op. 28: Prélude No. 12 in G-Sharp...	0.996	1941
78593	Ashok Kumar	Na Jane Kidhar Aaj Meri Nao	0.996	1941
78591	Khursheed	Toot Gayi Dali	0.996	1941
106466	Jacques Ibert, Hae Won Chang	Petite suite en 15 images: Berceuse aux etoile...	0.996	2000
143879	Robert Schumann, Sergei Rachmaninoff	Carnaval, Op. 9: 6. Florestan	0.996	1942
23609	Francisco Tárrega, Julio Martinez Oyanguren	Prelude No. 12	0.996	1941
143875	Robert Schumann, Sergei Rachmaninoff	Carnaval, Op. 9: 15. Pantalon et colombine	0.996	1942
1289	Bix Beiderbecke, The Wolverines	Tia Juana	0.996	1927
1288	Bix Beiderbecke, The Wolverines	Big Boy	0.996	1927
78540	Johann Sebastian Bach, Claudio Arrau	Goldberg Variations, BWV 988: Variation XXX - ...	0.996	1941
78516	Frédéric Chopin, Claudio Arrau	24 Préludes, Op. 28: Prélude No. 6 in B Minor	0.996	1941
78414	Rabindranath Tagore	Amare Ke Nibi Bhai	0.996	1940
1265	Bix Beiderbecke	In a Mist	0.996	1927
23514	Giorgos Papasideris, No. 6	Mixanikos	0.996	1940
143789	Elî Merdan	Mame	0.996	1940
78458	Markos Vamvakaris, Apostolos Xatzixristos	Mia omorfi melaxrini	0.996	1940
143749	Leyteris Melemenlis	Gianniotiko (paradosiako)	0.996	1940
23487	Dinendranath Tagore	Amar Milan Lagi Tumi	0.996	1940
23475	Milios	O Menoysis kai o Mpirmpilis	0.996	1940

Top 20 Canciones Según Acousticness y sus Interpretes

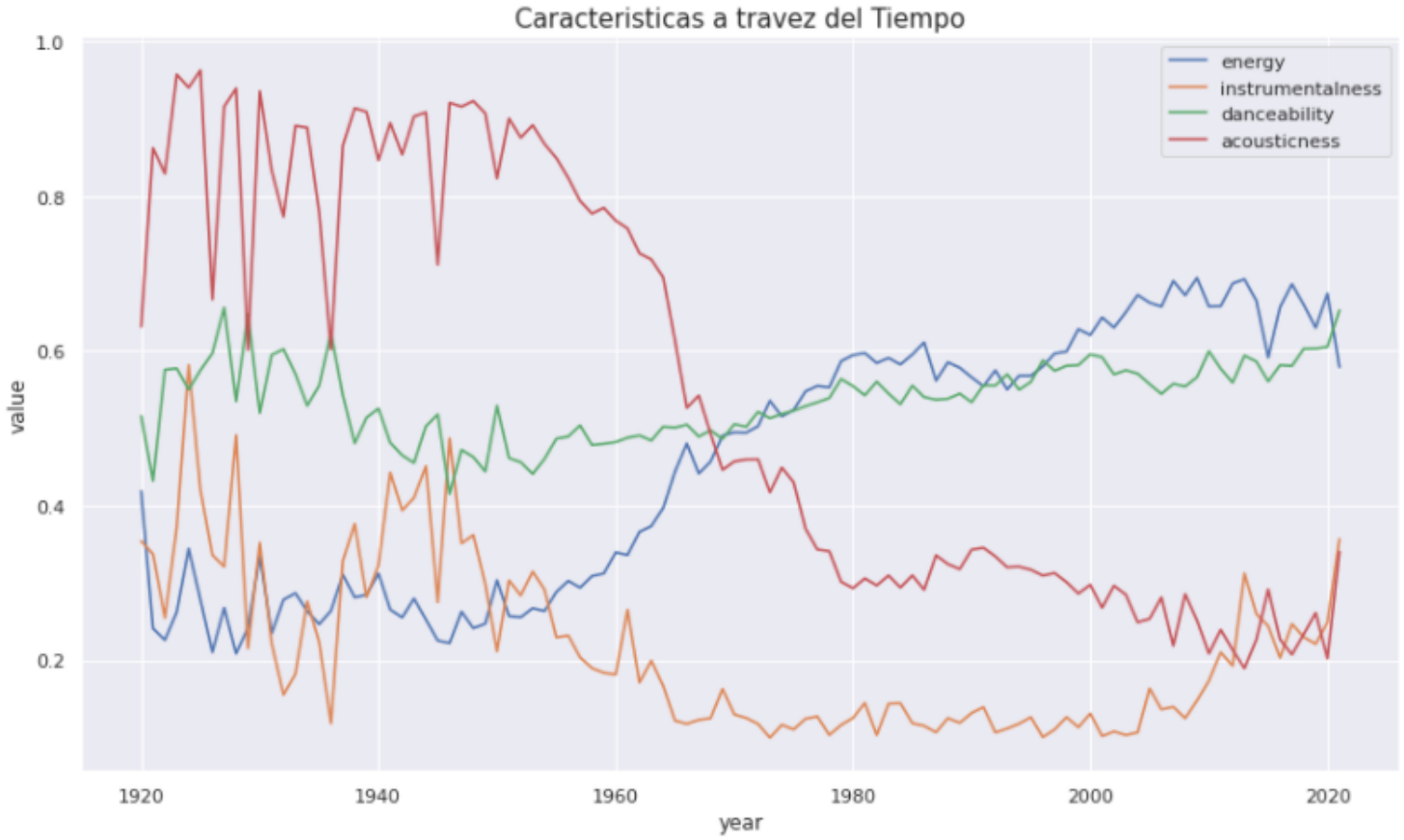
```
[222] artistas_canciones3 = datos2.iloc[:,[1,12,0,18]]
```

```
[223] acoustic = artistas_canciones3.sort_values('acousticness',ascending=False)
```

```
text7 = acoustic.name.head(20).values
wordcloud = WordCloud().generate(str(text7))
plt.figure( figsize=(10,10), facecolor='k')
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

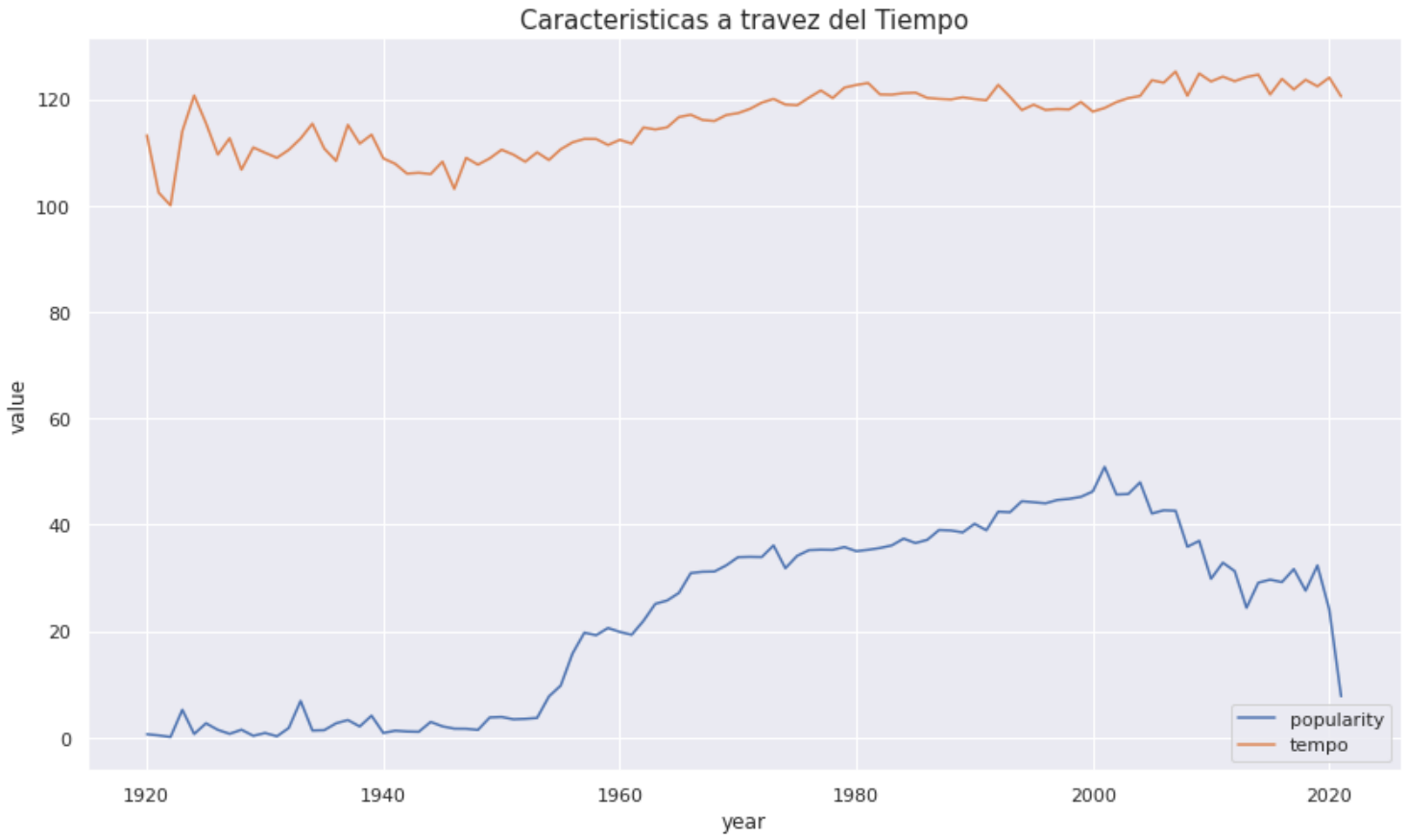
```
plt.figure(figsize=(14,8))
plt.title("Caracteristicas a travez del Tiempo", fontdict={"fontsize": 15})

lines = ['energy','instrumentalness','danceability','acousticness']

for line in lines:
    ax = sns.lineplot(x='year', y=line, data=year_avrg)
plt.ylabel("value")
plt.legend(lines)

# Según se aprecia en este gráfico los valores de Acousticness decaen con el pasar del tiempo, mientras que energy en su lugar
#incrementano con el pasar de los años , Danceability presenta fluctuaciones pero se recupera con el tiempo.
```

```
<matplotlib.legend.Legend at 0x7f3f548109d0>
```



- Comportamiento de energy, instrumentalness, danceability, acousticness a través del tiempo

- Según se aprecia en este gráfico los valores de **Acousticness** decaen con el pasar del tiempo, mientras que **energy** en su lugar se va incrementando.
- **Danceability** presenta fluctuaciones pero se recupera con el tiempo



Según se aprecia en este gráfico los valores en duración promedio de las canciones

para mantenerse con el pasar de los años, sin embargo el valor de la popularidad

sí se ve afectado por el paso del tiempo, las canciones más recientes presentan mayor popularidad

Modelo de Machine Learning

Modelo de Machine Learning

```
[ ]  datos1
```

	acousticness	danceability	duration_ms	energy	explicit	instrumentalness	liveness	loudness	popularity	speechiness	tempo	valence	year
0	0.991000	0.598	168333	0.224	0	0.000522	0.3790	-12.628	12	0.0936	149.976	0.6340	1920
1	0.643000	0.852	150200	0.517	0	0.026400	0.0809	-7.261	7	0.0534	86.889	0.9500	1920
2	0.993000	0.647	163827	0.186	0	0.000018	0.5190	-12.098	4	0.1740	97.600	0.6890	1920
3	0.000173	0.730	422087	0.798	0	0.801000	0.1280	-7.311	17	0.0425	127.997	0.0422	1920
4	0.295000	0.704	165224	0.707	1	0.000246	0.4020	-6.036	2	0.0768	122.076	0.2990	1920
...	...	...	...	...	...	...	...	...	...	...	...	...	...
174384	0.009170	0.792	147615	0.866	0	0.000060	0.1780	-5.089	0	0.0356	125.972	0.1860	2020
174385	0.795000	0.429	144720	0.211	0	0.000000	0.1960	-11.665	0	0.0360	94.710	0.2280	2021
174386	0.806000	0.671	218147	0.589	0	0.920000	0.1130	-12.393	0	0.0282	108.058	0.7140	2020
174387	0.920000	0.462	244000	0.240	1	0.000000	0.1130	-12.077	69	0.0377	171.319	0.3200	2021
174388	0.239000	0.677	197710	0.460	0	0.891000	0.2150	-12.237	0	0.0258	112.208	0.7470	2020

174389 rows × 13 columns

```
[ ]  X = datos1  # Renombrando variable para utilizarla en Scikit-Learn
```

```
[ ]  # Normalizando dataframe
     scaler = StandardScaler()
     X_std = scaler.fit_transform(X)
```

Componentes Principales como Optimizador de la cantidad de Variables Optimas

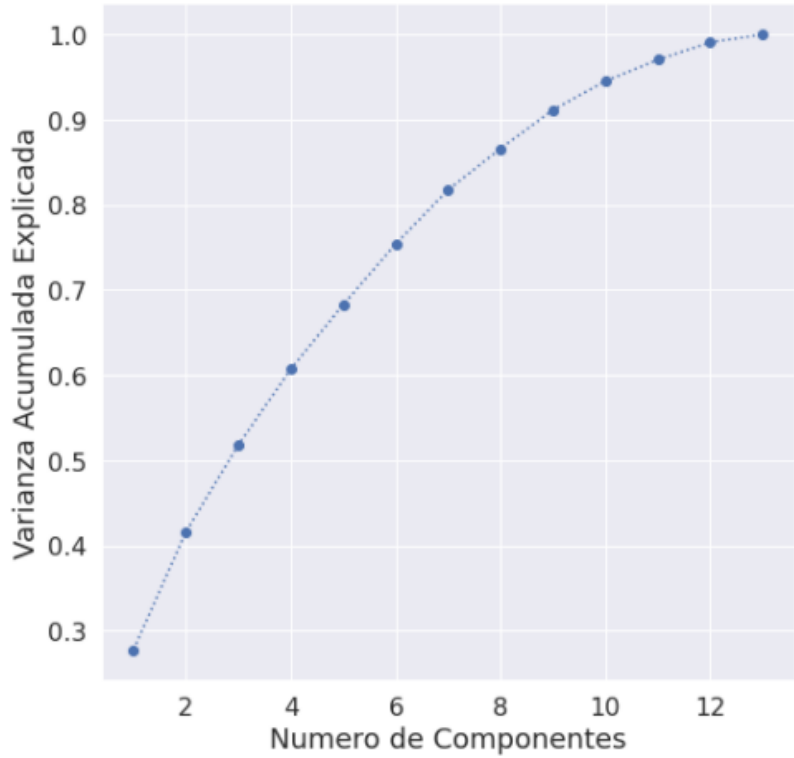
```
[ ]  # Importando PCA
     pca = PCA()
     pca.fit(X_std)
```

```
PCA(copy=True, iterated_power='auto', n_components=None, random_state=None,
    svd_solver='auto', tol=0.0, whiten=False)
```

```
[ ]  evr = pca.explained_variance_ratio_
     evr
```

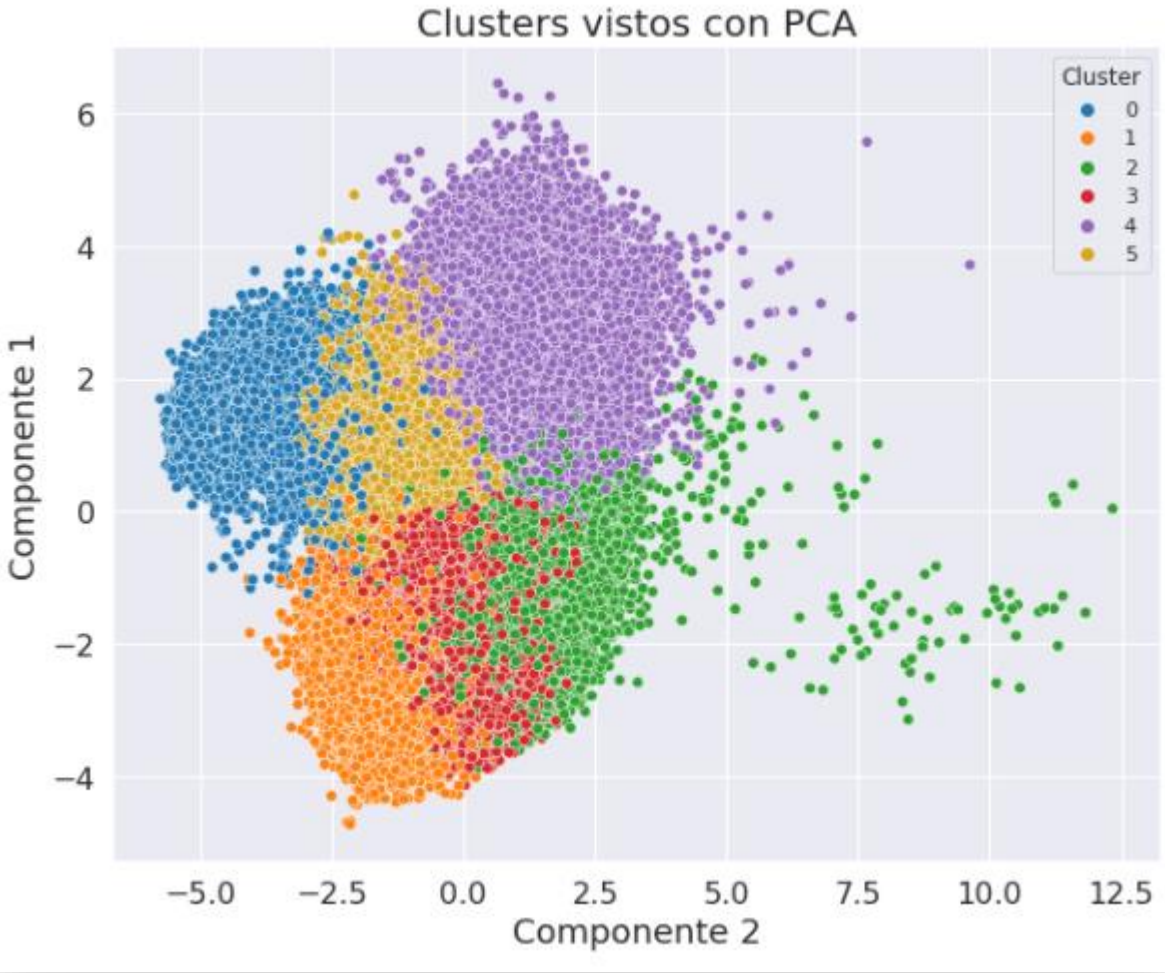
```
array([0.27784755, 0.13833818, 0.10201131, 0.08905998, 0.07564155,
       0.07117316, 0.06344042, 0.04854184, 0.04534896, 0.03380018,
       0.02501394, 0.02053265, 0.00925028])
```

```
[ ]  # Ploteando grafico de Componentes principales
     fig = plt.figure(figsize=(8,8))
     plt.plot(range(1, len(X.columns)+1), evr.cumsum(), marker='o', linestyle=':')
     plt.xlabel('Numero de Componentes', fontsize=18)
     plt.ylabel('Varianza Acumulada Explicada',fontsize=18)
     plt.xticks(fontsize=16)
     plt.yticks(fontsize=16)
     plt.show()
```




```
[ ]  # Iteracion para comprobar numero de componentes optimos a utilizar por su nivel de varianza

     for i, exp_var in enumerate(evr.cumsum()):
         if exp_var >= 0.8:
             n_comps = i + 1
             break
     print("Numero de Componentes Optimos:", n_comps)
     pca = PCA(n_components=n_comps)
     pca.fit(X_std)
     scores_pca = pca.transform(X_std)
```

```
Numero de Componentes Optimos: 7
```



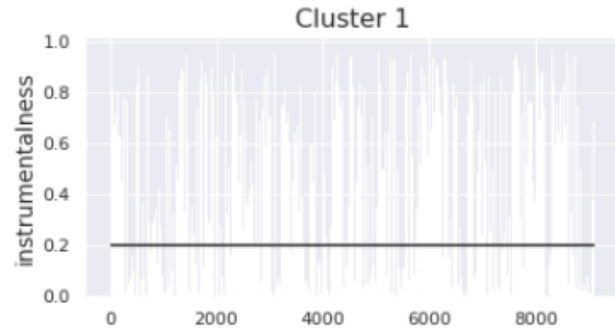
Se observan cluester bien definidos (el cluster 0 ,1, 4) pero a su vez se observan puntos muy fuera de los cluster.

Se debe proceder a realizar una revisión de outliers

```
# Visualizando caracteristicas generales de cada cluster

clusters = [1, 3, 5]
features = ["acousticness", "speechiness", "instrumentalness"]
#colors = ['tab:green', 'tab:olive', 'tab:cyan']
dim = len(clusters)

fig, axes = plt.subplots(dim, dim, figsize=(24, 12))
i = 0
test_cluster = data.loc[data['Cluster'] == clusters[0]]
for ax in (axes.flatten()):
    if i % dim == 0 and i != 0:
        test_cluster = data.loc[data['Cluster'] == clusters[i // dim]]
    col = features[i % dim]
    y = test_cluster[col]
    x = [i for i in range(len(y))]
    ax.bar(x, y) #colors[i//dim]
    ax.set_ylabel(col, fontsize=14)
    ax.set_title("Cluster " + str(clusters[i // dim]), fontsize=16)
    ax.hlines(np.mean(data[col]), 0, len(y))
    plt.subplots_adjust(wspace=.5, hspace=.5)
    i += 1
```

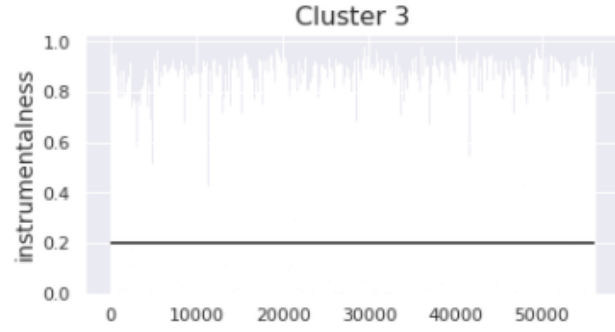

Cluster 1
acousticness
0.0
0.2
0.4
0.6
0.8
1.0
0
2000
4000
6000
8000

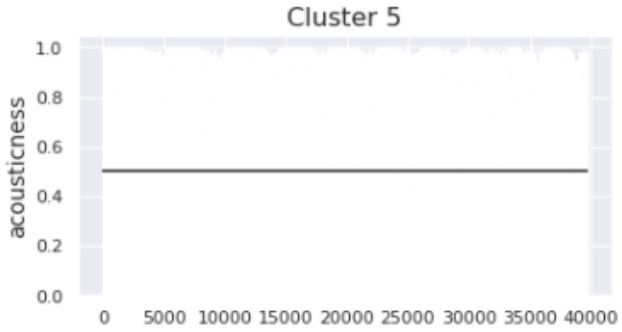

Cluster 1
speechiness
0.0
0.2
0.4
0.6
0.8
0
2000
4000
6000
8000

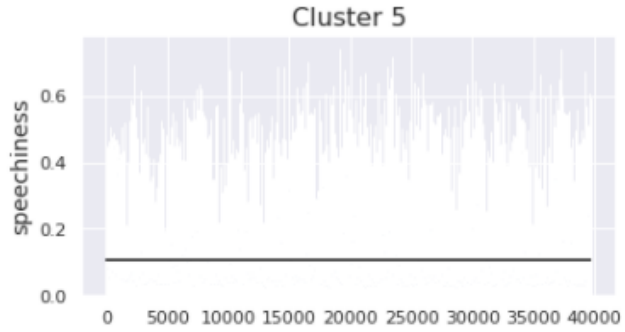

Cluster 1
instrumentalness
0.0
0.2
0.4
0.6
0.8
1.0
0
2000
4000
6000
8000

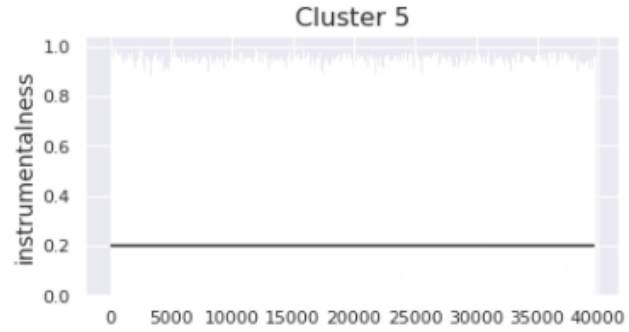

Cluster 3
acousticness
0.0
0.2
0.4
0.6
0.8
1.0
0
10000
20000
30000
40000
50000

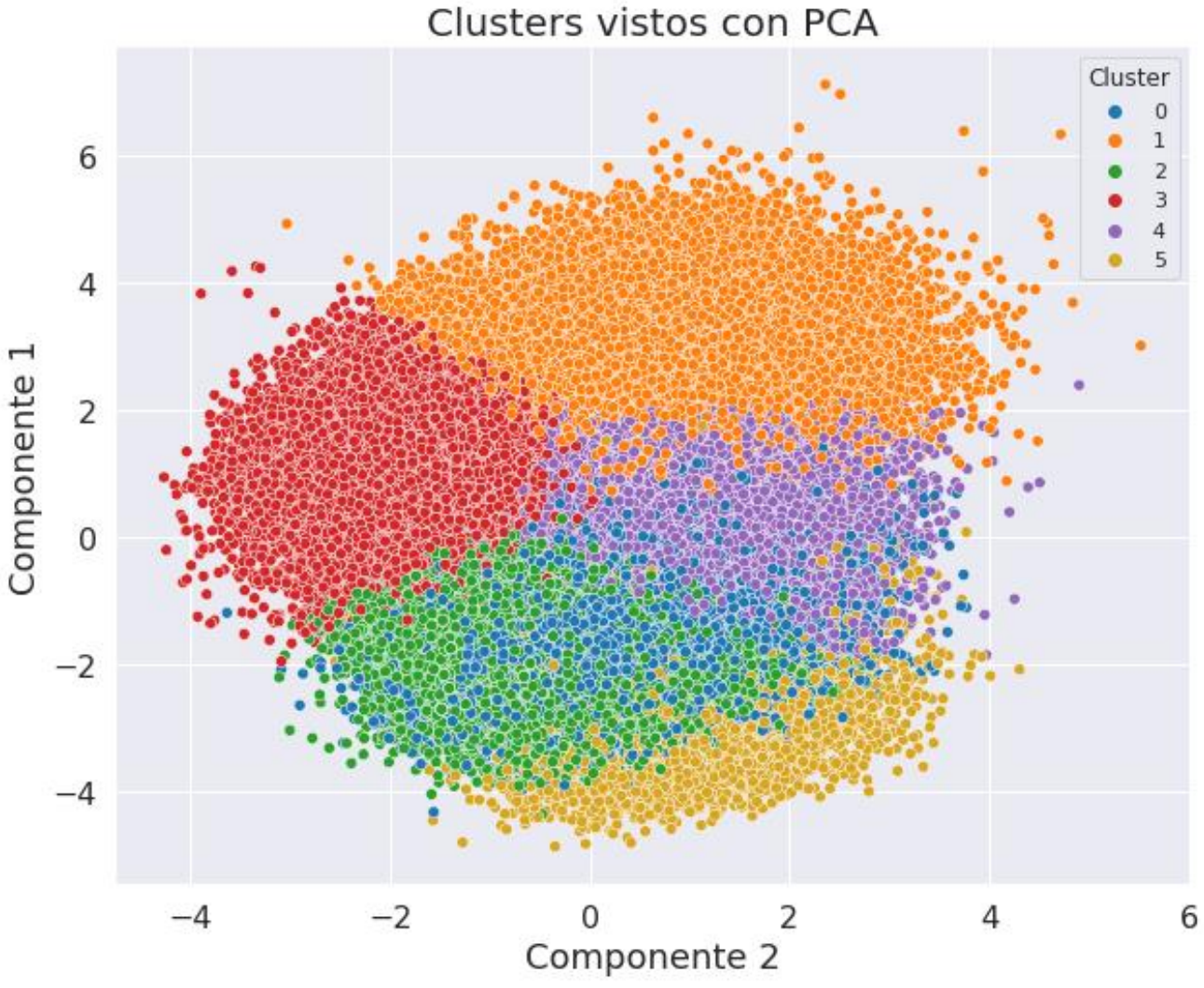

Cluster 3
speechiness
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
10000
20000
30000
40000
50000


Cluster 3
instrumentalness
0.0
0.2
0.4
0.6
0.8
1.0
0
10000
20000
30000
40000
50000


Cluster 5
acousticness
0.0
0.2
0.4
0.6
0.8
1.0
0
5000
10000
15000
20000
25000
30000
35000
40000


Cluster 5
speechiness
0.0
0.2
0.4
0.6
0
5000
10000
15000
20000
25000
30000
35000
40000

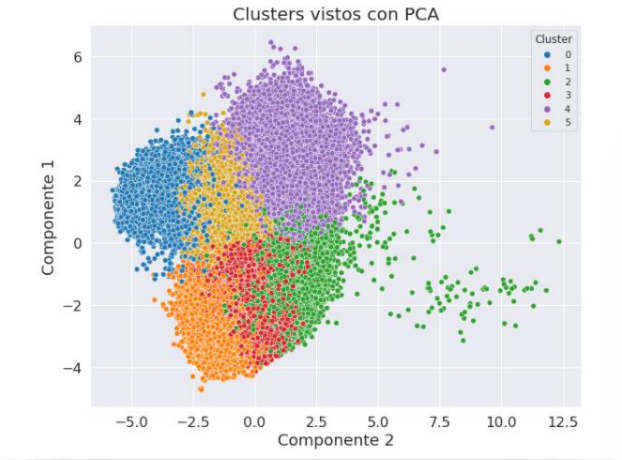

Cluster 5
instrumentalness
0.0
0.2
0.4
0.6
0.8
1.0
0
5000
10000
15000
20000
25000
30000
35000
40000

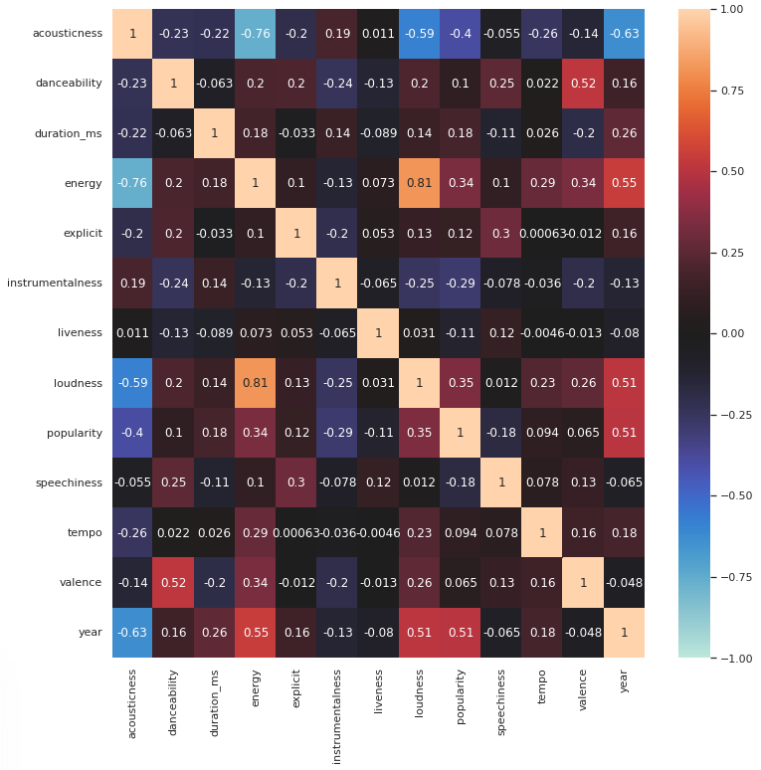


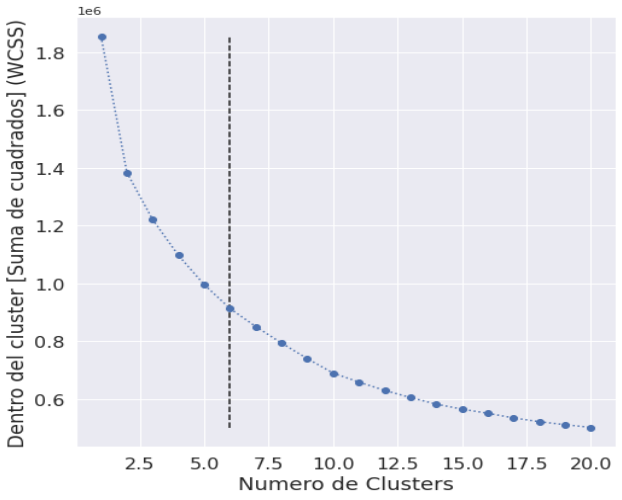
Clusters con Limpieza de Outliers

- Una vez aplicado una limpia en dos características de los valores outliers, se puede apreciar un cambio en lo que se refleja en los clusters.

Resumen de los Resultados







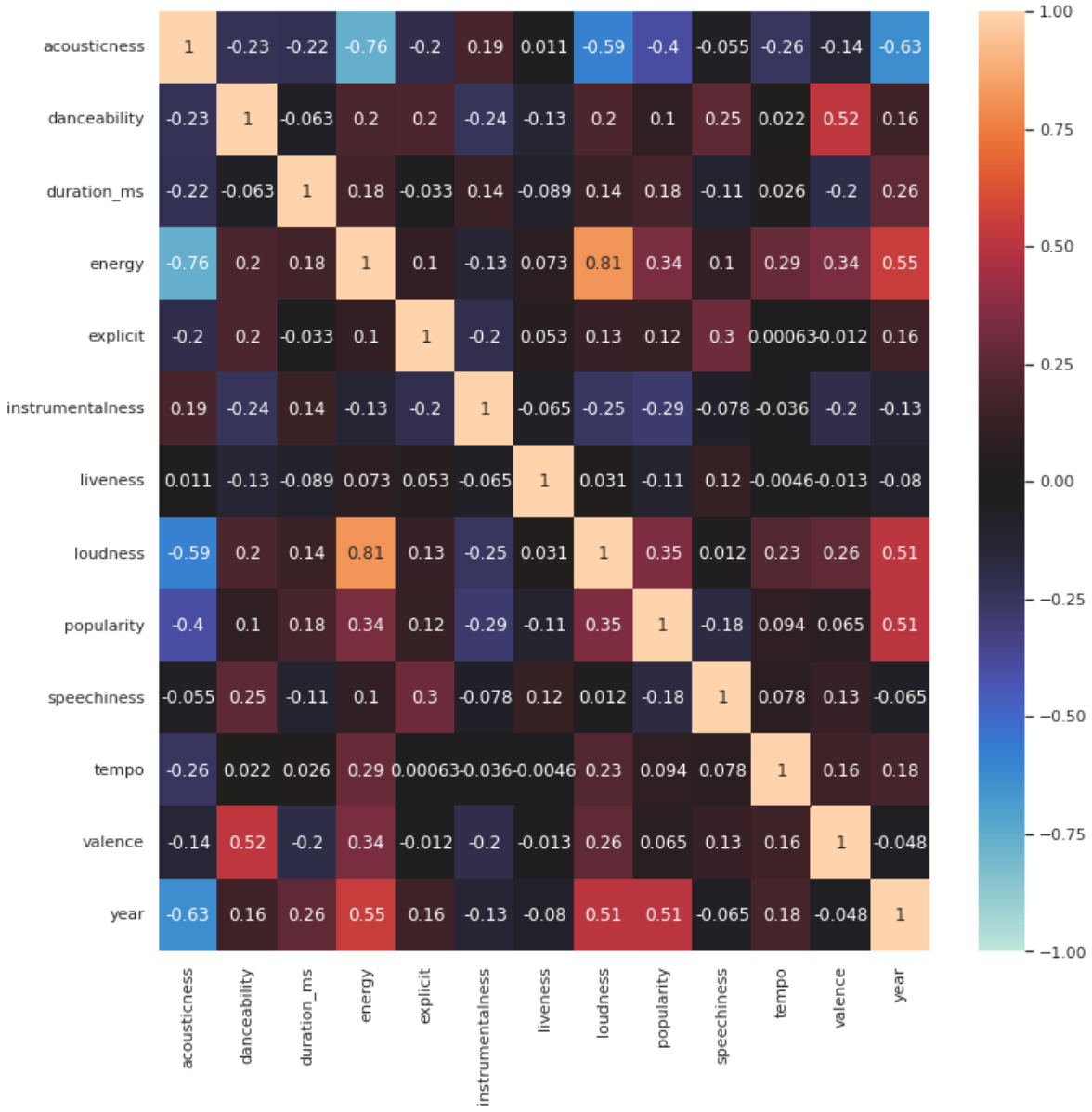
Resumen de los Resultados

- Songs dataset
- Top de Canciones
- Correlaciones

Resumen de Datos

```
[ ]  data.describe().T
```

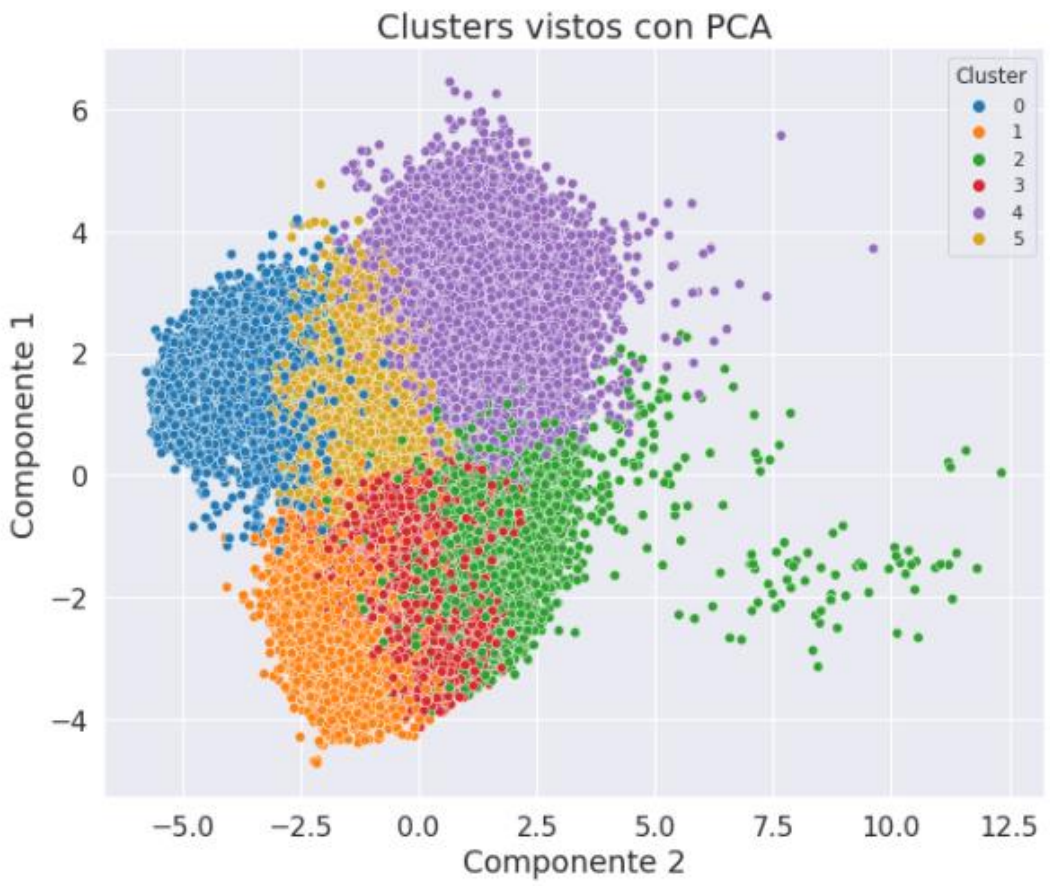
	count	mean	std	min	25%	50%	75%	max
acousticness	174389.0	0.499228	0.379936	0.0	0.0877	0.517000	0.8950	0.996
danceability	174389.0	0.536758	0.176025	0.0	0.4140	0.548000	0.6690	0.988
duration_ms	174389.0	232810.032026	148395.797680	4937.0	166133.0000	205787.000000	265720.0000	5338302.000
energy	174389.0	0.482721	0.272685	0.0	0.2490	0.465000	0.7110	1.000
explicit	174389.0	0.068135	0.251978	0.0	0.0000	0.000000	0.0000	1.000
instrumentalness	174389.0	0.197252	0.334574	0.0	0.0000	0.000524	0.2520	1.000
key	174389.0	5.205305	3.518292	0.0	2.0000	5.000000	8.0000	11.000
liveness	174389.0	0.211123	0.180493	0.0	0.0992	0.138000	0.2700	1.000
loudness	174389.0	-11.750865	5.691591	-60.0	-14.9080	-10.836000	-7.4990	3.855
mode	174389.0	0.702384	0.457211	0.0	0.0000	1.000000	1.0000	1.000
popularity	174389.0	25.693381	21.872740	0.0	1.0000	25.000000	42.0000	100.000
speechiness	174389.0	0.105729	0.182260	0.0	0.0352	0.045500	0.0763	0.971
tempo	174389.0	117.006500	30.254178	0.0	93.9310	115.816000	135.0110	243.507
valence	174389.0	0.524533	0.264477	0.0	0.3110	0.536000	0.7430	1.000
year	174389.0	1977.061764	26.907950	1920.0	1955.0000	1977.000000	1999.0000	2021.000
Cluster	174389.0	3.273578	1.327022	0.0	2.0000	3.000000	4.0000	5.000



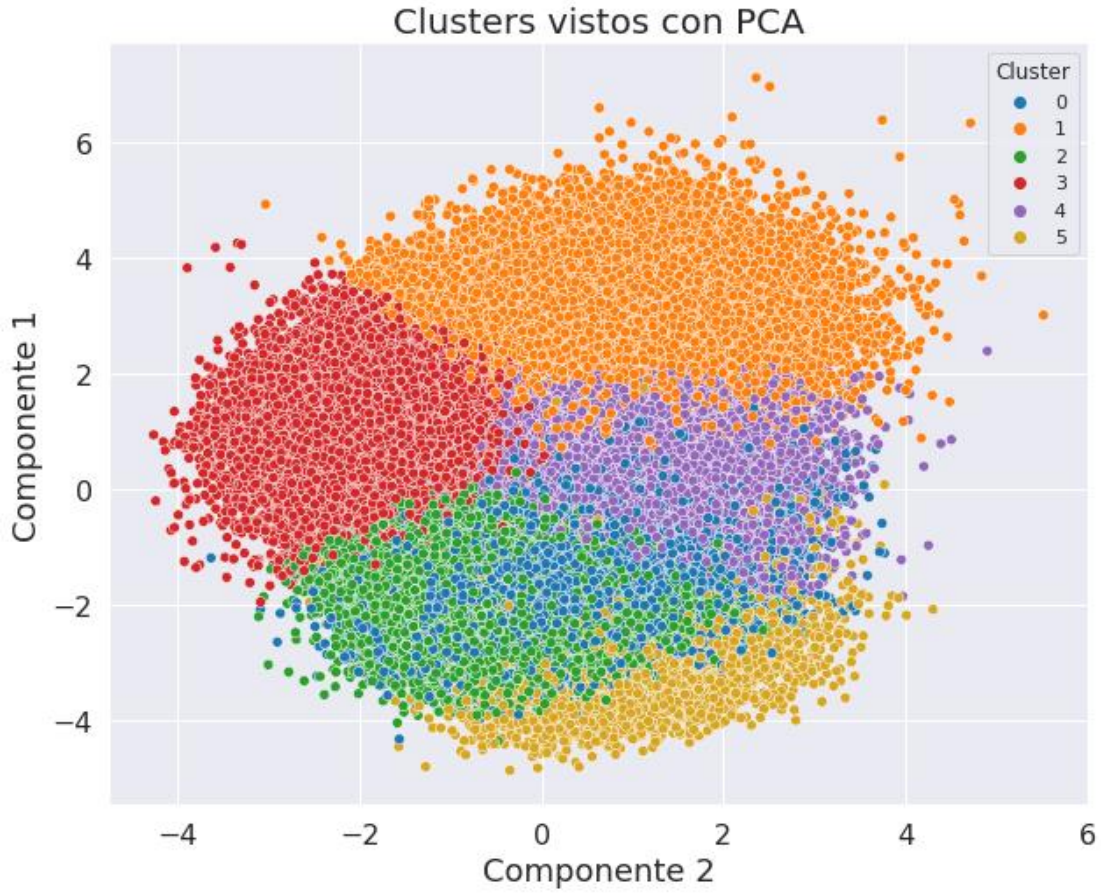
- La caracteristica **explicit** presenta una leve relación con la caracteristica **speechiness** con un valor de 0.3, si incrementa una puede presentar un incremento en la otra
- La caracteristica presenta **explicit** una relación negativa leve con las caracteristicas **acousticness** y **instrumentalness** con un valor de -0.2 con ambas caracteristicas.
- la caracteristica **liveness** presenta poca o casi nula en las correlaciones con las demás caracteristicas. con la que presenta el valor mayor sería con **danceability** con un valor negativo de -0.13

- La caracteristica **loudness** presenta una correlación positiva alta con la caracteristica **energy** con un valor de 0.81 si aumenta una la otra tambien aumenta
- **popularity** presenta una correlación positiva alta con la caracteristica **year** por lo que si aumenta una la otra también aumentará. Además la carcteristica **popularity** presenta una correlación negativa con la caractistica **acousticness** con un valor de -0.4 si aumenta una la otra va a disminuir.

Sin Tratamiento de Outliers

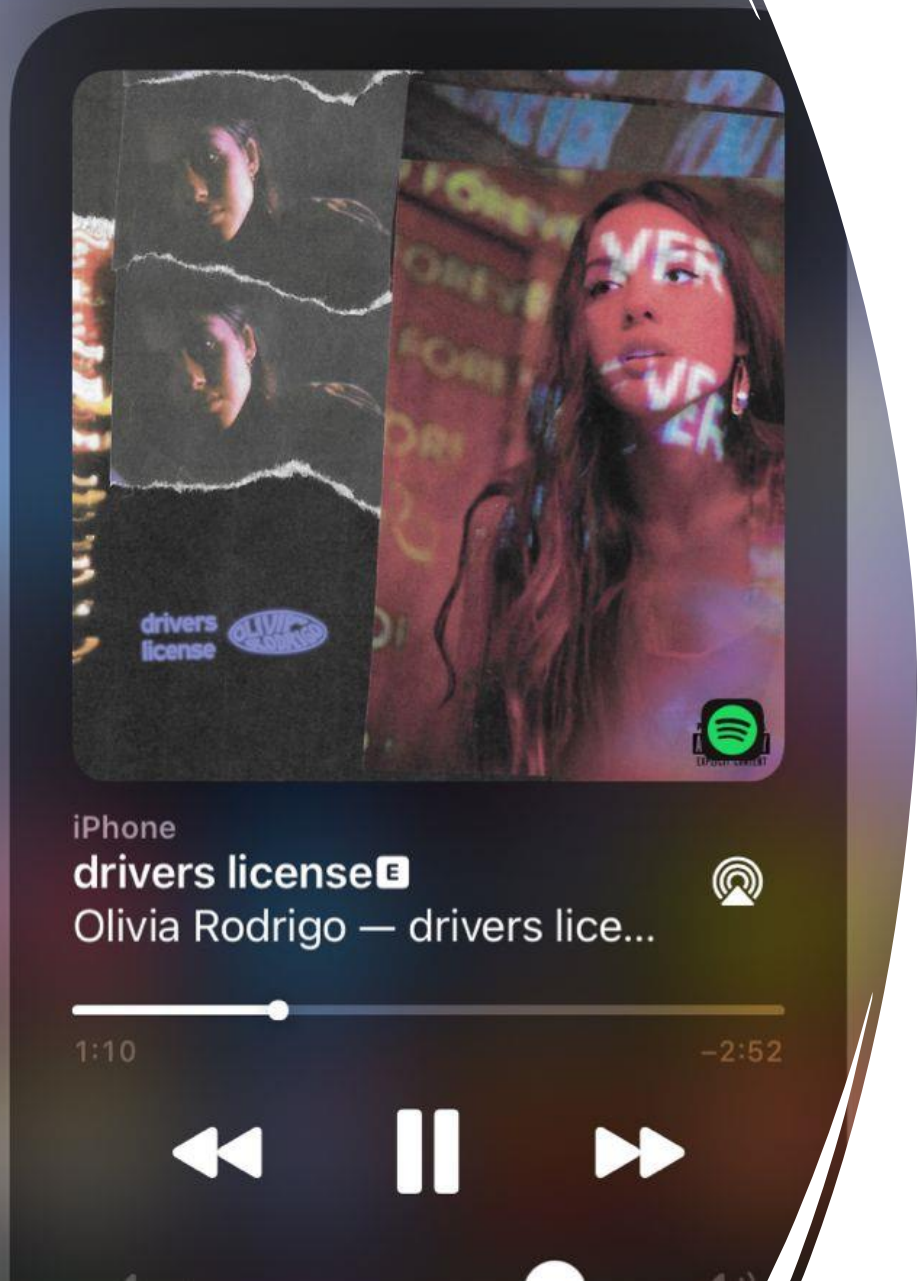


Con Tratamiento de Outliers



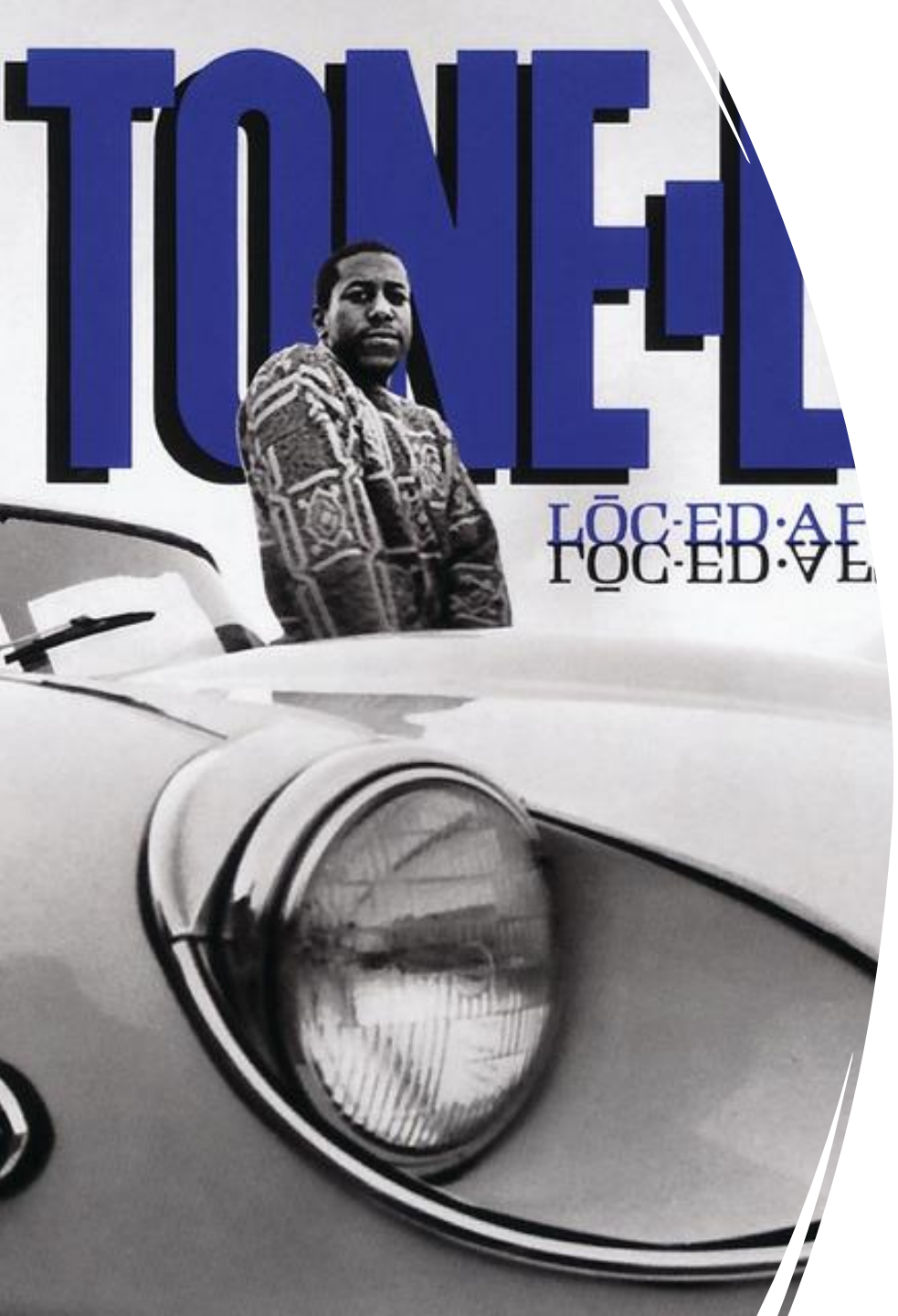
El artista que aparece como moda en el dataset

- Tadeusz Dołęga-Mostowicz fue un escritor, periodista y autor polaco de más de una docena de novelas populares. Por lo tanto el artista que aparece como moda en el dataset corresponde a una lista de reproducciones de relatos cortos, poesía y demás, muy distinto a lo que podría suponerse como un artista moderno.



La canción top según popularidad

drivers license



La canción top según Danceability

- Funky Cold Medina

Canción Top según característica Acousticness

—

- Mentiras. Año 1930

Songs dataset

Caracteristica a considerar a futuro

El dataset podría haber incluido una caracteristica de **genero musical**, con esto se lograría un agrupamiento de los datos según el genero musical. Esto podría permitir asignar relaciones entre las otras caracteristiscas a cada genero musical.

Recomendaciones Musicales

Se pueden realizar recomendaciones musicales o playlist por algunas caracteristicas que presenten las canciones que el usuario escuche, si por casualidad el usario tiende a escuchar musica actual pero con caracteristicas de acustics, entonces se le puede recomendar algunas acustic greathits por épocas, si el usuario se interesa más por caracteristicas con Danceability, entonces la App puede recomendarle canciones Mix Dance.