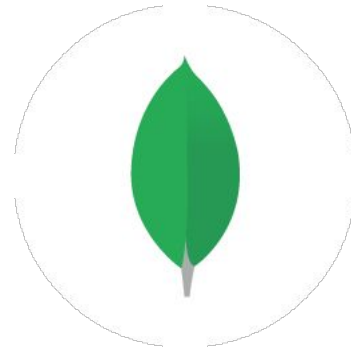


PROYECTO 1

Aplicación con una base de datos NoSQL:

Análisis de artistas en Spotify



Equipo: Top Tiers

Introducción

- Aproximadamente 7 de cada 10 (67%) adultos han utilizado plataformas de streaming para escuchar música durante el último mes.
- Spotify es la más popular a nivel mundial:
 - Más de 60 millones de canciones disponibles
 - 1.5 millones de podcasts
 - Aproximadamente 299 millones de usuarios activos mensualmente

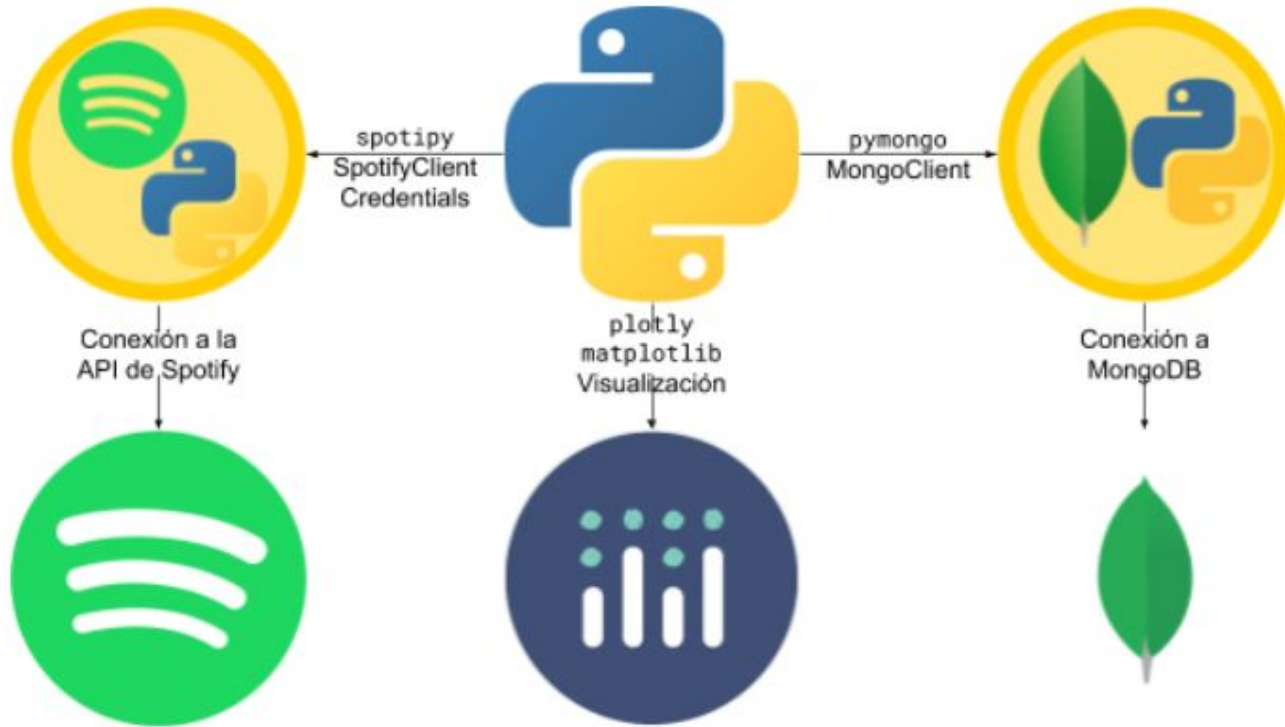


Solución

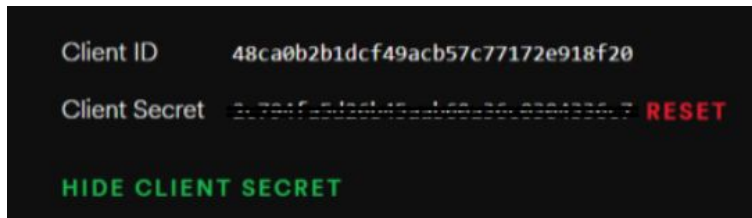
- URI (indicador uniforme de recursos) de Spotify:
 - Enlace que se puede encontrar en el menú compartir de cualquier pista, álbum o perfil de artista en Spotify.
- 81,323 artistas únicos y sus URIs

80110 unique values	81322 unique values
2:00 AM	spotify:artist:4tN3rZ7cChj4Wns2Wt2Nj6
2:15	spotify:artist:4Hs0m6VnKZtGh8W8GhdNu4
2:54	spotify:artist:3LsQK0RgMc8VEkQn66jFAQ
4:20	spotify:artist:5KCG0FDMDPzQpxcohGUnyH
6:30	spotify:artist:5abb1GojcZoe3zZBhJFBUi
6:33	spotify:artist:0oBPg2seHzVcAI0pdi10jj
10:32	spotify:artist:4CwN7lHe0BgAcyk7nNKSTJ
4:00 PM	spotify:artist:6n4LC99DYefuU6icCPk1zU
2	spotify:artist:6p0CEbvtPkErkz3884ikYF

Arquitectura del sistema



Funcionalidades principales



Se realizaron:

- Cinco consultas
- Cuatro gráficas interactivas
- Una plantilla de imágenes

```
{'external_urls': {'spotify': 'https://open.spotify.com/artist/1r4hJ1h58CWwUQe3MxPuau'},
'followers': {'href': None, 'total': 20894502},
'genres': ['reggaeton', 'reggaeton colombiano'],
'href': 'https://api.spotify.com/v1/artists/1r4hJ1h58CWwUQe3MxPuau',
'id': '1r4hJ1h58CWwUQe3MxPuau',
'images': [{'height': 640,
'url': 'https://i.scdn.co/image/0b6f5467bbde78796b33d8eed85157620b3080cc',
'width': 640},
{'height': 320,
'url': 'https://i.scdn.co/image/6265d3a759f85b133744209341c9bc87f75d919c',
'width': 320},
{'height': 160,
'url': 'https://i.scdn.co/image/bb1ae742982c6e4d90c8d40d1f860b1e3f89e7d2',
'width': 160}],
'name': 'Maluma',
'popularity': 95,
'type': 'artist',
'uri': 'spotify:artist:1r4hJ1h58CWwUQe3MxPuau'}
```

Herramientas nuevas utilizadas

— spotify —

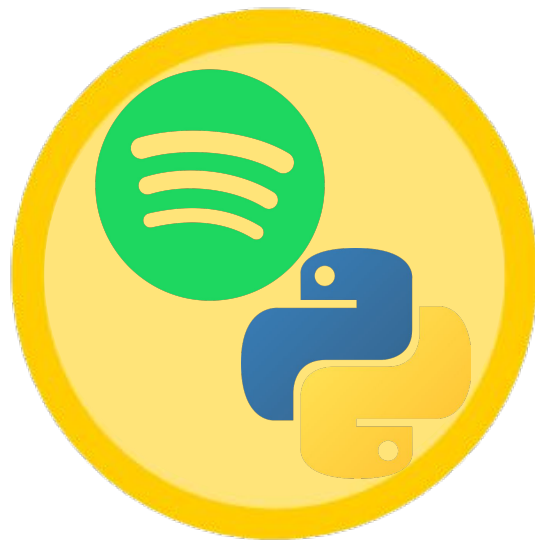
- Librería de Python para la API web de Spotify.
- Se obtiene acceso completo a todos los datos musicales.

Se puede instalar desde la terminal mediante el comando

```
pip install spotipy
```

o en Anaconda con

```
conda install -c jkroes spotipy
```



Herramientas nuevas utilizadas

— pandas —

- Su estructura de datos clave se llama DataFrame:
 - Permiten almacenar y manipular datos tabulares en filas de observaciones y columnas de variables.



Se puede instalar desde la terminal mediante el comando

```
pip install pandas
```

o en Anaconda con

```
conda install -c anaconda pandas
```

Herramientas nuevas utilizadas

— json —

- Librería estándar de Python que permite la codificación y decodificación de JSON
- El método `dump()` nos permitió convertir objetos de Python a JSON.

```
import json

# función para escribir en el archivo JSON
def write_json(data, filename='data_artists.json'):
    with open(filename, 'w') as f:
        json.dump(data, f, indent=4)

# consultamos y almacenamos la información por artista
with open('data_artists.json') as json_file:
    artists_uris_list = list(artists_uris['uri'])
    data = []

    for i in range(0, len(artists_uris_list), 50):
        x = sp.artists(artists_uris_list[i:i + 50])['artists']
        data = data + x

# Escribimos los datos en un archivo JSON
write_json(data)
```


Herramientas nuevas utilizadas

— plotly —



- Librería de Python capaz de crear gráficas interactivas de alta calidad.

Se puede instalar desde la terminal mediante el comando

```
pip install plotly==4.10.0
```

o en Anaconda con

```
conda install -c plotly  
plotly=4.10.0
```

Para usar en Jupyter Notebook, se deben instalar los paquetes notebook y ipywidgets mediante el comando

```
pip install "notebook>=5.3"  
"ipywidgets==7.5"
```

o en Anaconda con

```
conda install "notebook>=5.3"  
"ipywidgets=7.5"
```

Herramientas nuevas utilizadas

— matplotlib —

- Librería de Python para la generación de gráficos a partir de datos contenidos en listas o arrays.

Se puede instalar desde la terminal mediante el comando

```
pip install matplotlib
```

o en Anaconda con conda

```
install -c conda-forge matplotlib.
```



Herramientas nuevas utilizadas

— collections —

- Librería estándar de Python.
- De aquí se importa el objeto `Counter()`: se utiliza en la solución para contar los elementos de una lista.

```
from collections import Counter

# consultamos los generos de los artistas que rebasan un cierto umbral de popularidad
popularity_threshold = 80
lista_generos = mycol.find({ "popularity": { "$gt": popularity_threshold } }, {"genres":1, "_id":0})

# liampiamos un poco los datos
lista_generos = [e for genero in lista_generos for e in genero['genres']]
diccionario_generos = Counter(lista_generos)
```

Obtención y procesamiento de los datos

```
In [7]: artists_uris = pd.read_csv("artists_uris.csv", names=['name', 'uri'])
        print(artist_uris.shape)
        artists_uris.head()
```

(81323, 2)

Out[7]:

	name	uri
0	1:43	spotify:artist:39EHxSQAIAWusRqSI9xoyF
1	2:00 AM	spotify:artist:4tN3rZ7cChj4Wns2Wt2Nj6
2	2:15	spotify:artist:4HsOm6VnKZtGh8W8GhdNu4
3	2:54	spotify:artist:3LsQKoRgMc8VEkQn66jfAQ
4	4:20	spotify:artist:5KCG0FDMDPzQpxcohGUnyH

```
import json

# función para escribir en el archivo JSON
def write_json(data, filename='data_artists.json'):
    with open(filename, 'w') as f:
        json.dump(data, f, indent=4)

# consultamos y almacenamos la información por artista
with open('data_artists.json') as json_file:
    artists_uris_list = list(artist_uris['uri'])
    data = []

    for i in range(0, len(artist_uris_list), 50):
        x = sp.artists(artist_uris_list[i:i + 50])['artists']
        data = data + x

# Escribimos los datos en un archivo JSON
write_json(data)
```

Almacenamiento de los datos

```
from pymongo import MongoClient as Connection

connection = Connection('localhost',27017)
mydb = connection["spotify_data"]
mycol = mydb["artistas"]
mycol.drop() # borramos la colección por si ya existe
mycol = mydb["artistas"]

with open('data_artists.json') as json_file:
    artist_data = json.load(json_file)

# insertamos datos
mycol.insert_many(artist_data)
```

```
{
  "_id": ObjectId("5f7560430ea6758368bffc2b"),
  "external_urls": Object {
    "spotify": "https://open.spotify.com/artist/3WrFJ7ztbogyGnTHbHJf12"
  },
  "followers": Object {
    "href": null,
    "total": 17387526
  },
  "genres": Array [
    0: "beatlesque"
    1: "british invasion"
    2: "classic rock"
    3: "merseybeat"
    4: "psychedelic rock"
    5: "rock"
  ],
  "href": "https://api.spotify.com/v1/artists/3WrFJ7ztbogyGnTHbHJf12",
  "id": "3WrFJ7ztbogyGnTHbHJf12",
  "images": Array [
    0: Object {
      "height": 640
      "url": "https://i.scdn.co/image/6b2a709752ef9c7aaf0d270344157f6cd2e0f1a7"
      "width": 640
    },
    1: Object {
      "height": 320
      "url": "https://i.scdn.co/image/1047bf172446f2a815a99ab0a0395099d621be51"
      "width": 320
    },
    2: Object {
      "height": 160
      "url": "https://i.scdn.co/image/0561b59a91a5e904ad2d192747715688d5f05012"
      "width": 160
    }
  ],
  "name": "The Beatles",
  "popularity": 89,
  "type": "artist",
  "uri": "spotify:artist:3WrFJ7ztbogyGnTHbHJf12"
}
```

Estructura de las “tablas” de la BD

Llave	Tipo de valor	Descripción
external_urls	Un external_urls object	URL externas conocidas de este artista.
followers	Un followers object	Información sobre los seguidores del artista.
genres	Arreglo de strings	Una lista de los géneros con los que está asociado el artista. Por ejemplo: "Prog Rock", "Post-Grunge". (Si aún no está clasificado, el arreglo está vacío).
href	string	Un link al Web API endpoint que proporciona todos los detalles del artista.
id	string	El ID del artista en Spotify.
images	Arreglo de image object	Imágenes del artista en varios tamaños, empezando por la más amplia.
name	string	El nombre del artista
popularity	int	La popularidad del artista. El valor estará entre 0 y 100, siendo 100 el más popular. La popularidad del artista se calcula a partir de la popularidad de todas las pistas del artista.
type	string	El tipo de objeto: "artist"
uri	string	El URI de Spotify para el artista.

Resultados — Consultas

```
# Cantidad total de documentos en la colección de artistas  
cantidad_documentos = mycol.count_documents({})  
print(f"La cantidad de documentos en la colección de artistas es de: {cantidad_documentos}")
```

La cantidad de documentos en la colección de artistas es de: 81323

```
# Cantidad de artistas con popularidad menor a un cierto umbral  
popularity_threshold = 30  
cantidad = mycol.count_documents({ "popularity": { "$lt": popularity_threshold } })  
print(f"La cantidad de artistas con menos de {popularity_threshold} puntos de popularidad es de: {cantidad}")
```

La cantidad de artistas con menos de 30 puntos de popularidad es de: 36346

Resultados — Consultas

```
genres = ['pop', 'rock', 'classical', 'rap', 'hip hop']  
  
# Cantidad total de documentos que contienen diversos géneros  
for genre in genres:  
    cantidad = mycol.count_documents({ "genres": genre})  
    print(f"La cantidad de artistas con el género {genre} es de: {cantidad}")
```

```
La cantidad de artistas con el género pop es de: 557  
La cantidad de artistas con el género rock es de: 581  
La cantidad de artistas con el género classical es de: 210  
La cantidad de artistas con el género rap es de: 464  
La cantidad de artistas con el género hip hop es de: 478
```


Resultados — Consultas

```
from collections import Counter

# consultamos los generos de los artistas que rebasan un cierto umbral de popularidad
popularity_threshold = 80
lista_generos = mycol.find({ "popularity": { "$gt": popularity_threshold } }, {"genres":1, "_id":0})

# liampiamos un poco los datos
lista_generos = [e for genero in lista_generos for e in genero['genres']]
diccionario_generos = Counter(lista_generos)

import plotly.express as px

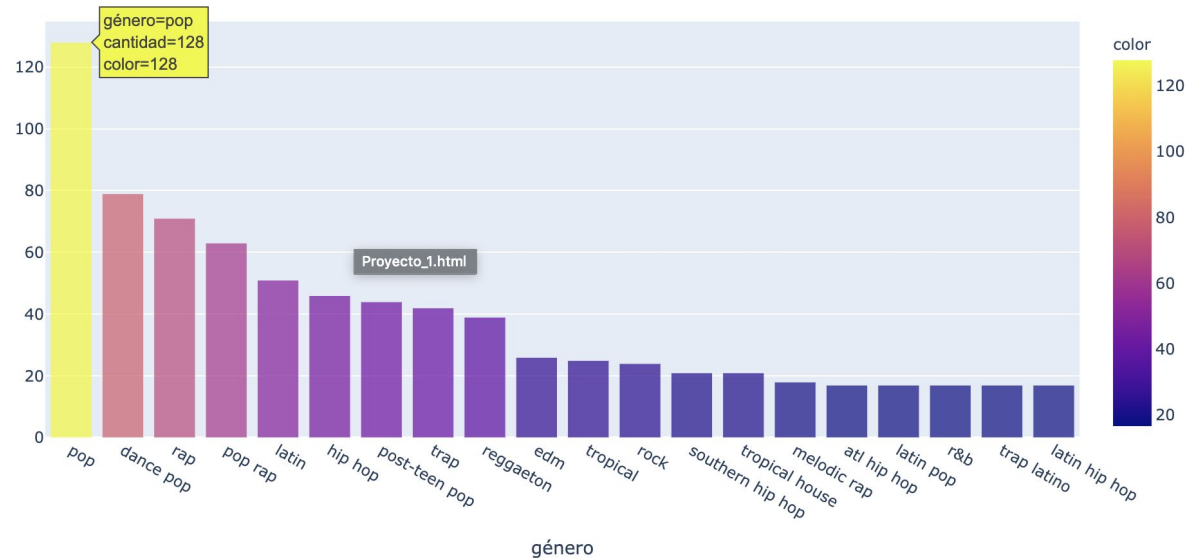
genres_threshold = 20

nombre_genero = [e[0] for e in diccionario_generos.most_common()[:genres_threshold]]
valor_genero = [e[1] for e in diccionario_generos.most_common()[:genres_threshold]]

fig = px.bar(x=nombre_genero,
             y=valor_genero,
             labels = {"x":"género","y":"cantidad"},
             title="Géneros más comunes en los artistas más populares",
             color=valor_genero,
             opacity=0.7
            )
fig.show()
```

Resultados — Gráficos

Géneros más comunes en los artistas más populares



Resultados — Consultas

```
# consultamos los generos de los artistas que no rebasan un cierto umbral de popularidad
popularity_threshold = 10
lista_generos = mycol.find({ "popularity": { "$lt": popularity_threshold } }, {"genres":1, "_id":0})

# limpiamos los datos
lista_generos = [e for genero in lista_generos for e in genero['genres']]
diccionario_generos = Counter(lista_generos)

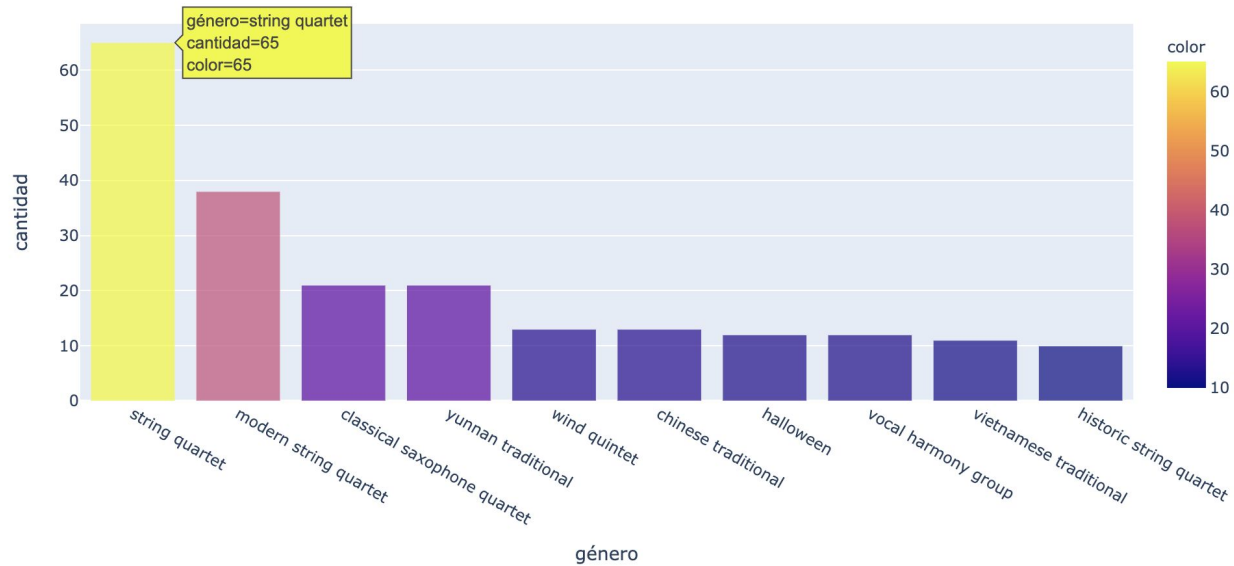
# graficamos
genres_threshold = 10

nombre_genero = [e[0] for e in diccionario_generos.most_common()[:genres_threshold]]
valor_genero = [e[1] for e in diccionario_generos.most_common()[:genres_threshold]]

fig = px.bar(x=nombre_genero,
             y=valor_genero,
             labels = {"x":"género", "y":"cantidad"},
             title="Géneros más comunes en los artistas menos populares",
             color=valor_genero,
             opacity=0.7
            )
fig.show()
```

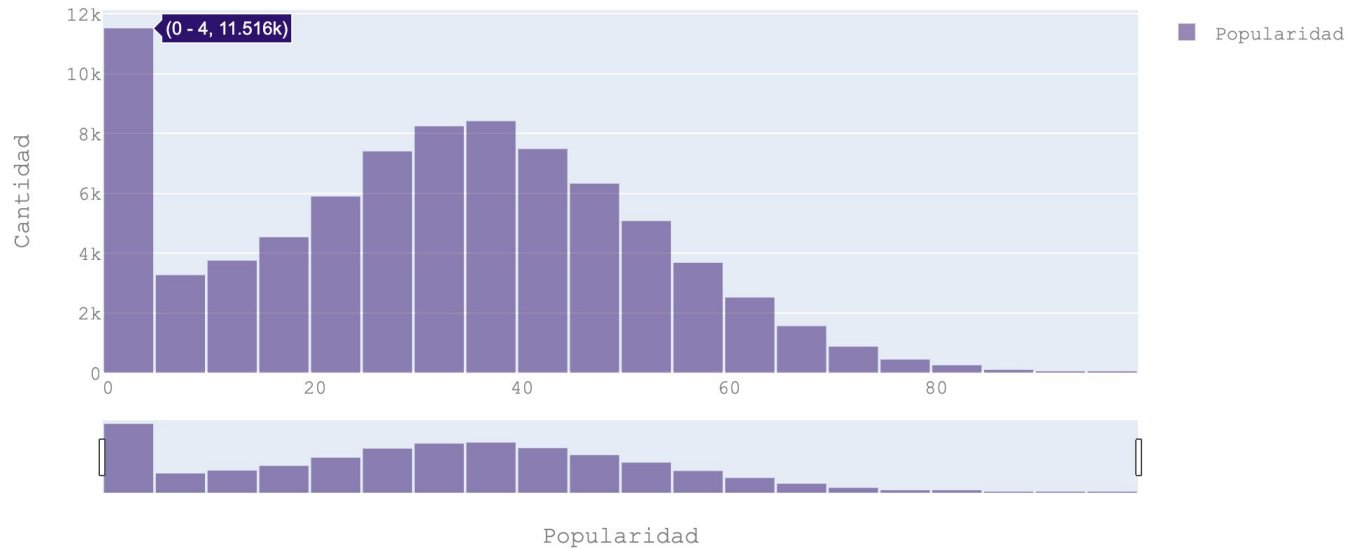
Resultados — Gráficos

Géneros más comunes en los artistas menos populares

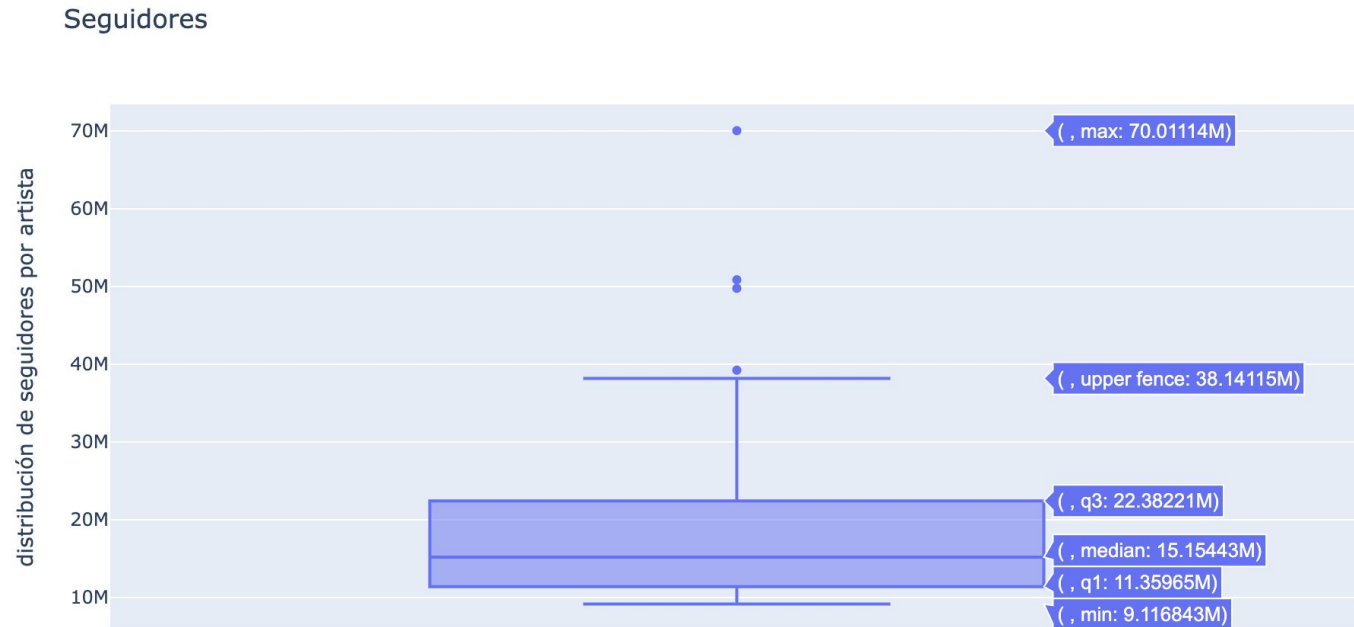


Resultados — Gráficos

Popularidad de artistas



Resultados — Gráficos

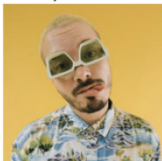


Resultados — Gráficos

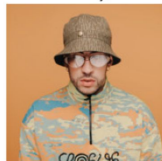
Drake, 99



J Balvin, 98



Bad Bunny, 98



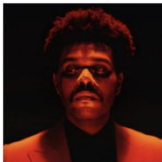
BTS, 97



Ozuna, 96



The Weeknd, 96



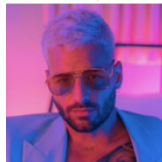
Taylor Swift, 96



Travis Scott, 96



Maluma, 95



DaBaby, 95



Lil Baby, 95



Post Malone, 95



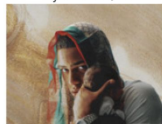
Ariana Grande, 94



Anuel AA, 94



Myke Towers, 94



Justin Bieber, 94



Mejoras

- Utilizar la autenticación de Spotify para conseguir más datos.
- Crear más colecciones en MongoDB para almacenar más datos.

Conclusiones

- Tendencias actuales en géneros y artistas en la música de streaming.
- Ahorro de tiempo al no tener que definir múltiples tablas y relaciones.
- Velocidad en las consultas.
- Nuevas herramientas.

Bibliografía

- [1] Global Web Index, "Infographic: Music Streaming Around the World", *Global Web Index*, 2020. [Online]. Available: <https://www.globalwebindex.com/reports/music-streaming-around-the-world>. [Accessed: 30- Sep- 2020].
- [2] Spotify AB, "Company Info", *Spotify*, 2020. [Online]. Available: <https://newsroom.spotify.com/company-info/#:~:text=Today%2C%20Spotify%20is%20the%20world's,138m%20subscribers%2C%20across%2092%20markets>. [Accessed: 30- Sep- 2020].
- [3] Spotify AB, "What's a Spotify URI?", *Spotify Community*, 2018. [Online]. Available: <https://community.spotify.com/t5/Spotify-Answers/What-s-a-Spotify-URI/ta-p/919201>. [Accessed: 30- Sep- 2020].
- [4] E. Call, "Spotify Artists", *Kaggle*, 2017. [Online]. Available: <https://www.kaggle.com/ehcall/spotify-artists>. [Accessed: 30- Sep 2020].

Bibliografía

- [5] P. Lamere, "Welcome to Spotipy! — spotipy 2.0 documentation", *Spotipy*, 2014. [Online]. Available: <https://spotipy.readthedocs.io/en/2.16.0/>. [Accessed: 30- Sep- 2020].
- [6] Spotify AB, "Platform Documentation", *Spotify for Developers*, 2020. [Online]. Available: <https://developer.spotify.com/documentation/>. [Accessed: 30- Sep- 2020].
- [7] MongoDB, Inc., "PyMongo 3.11.0 Documentation", *Pymongo*, 2020. [Online]. Available: <https://pymongo.readthedocs.io/en/stable/>. [Accessed: 30- Sep- 2020].
- [8] Plotly Technologies Inc., "Plotly: The front-end for ML and data science models", *Plotly*, 2020. [Online]. Available: <https://plotly.com/>. [Accessed: 30- Sep- 2020].
- [9] The Matplotlib development team, "Matplotlib 3.3.2 documentation", *Matplotlib*, 2020. [Online]. Available: <https://matplotlib.org/>. [Accessed: 30- Sep- 2020].

Bibliografía

[10] learnpython.org, "Pandas Basics", *Learn Python - Free Interactive Python Tutorial*. [Online]. Available: https://www.learnpython.org/en/Pandas_Basics#:~:text=Pandas%20is%20a%20high%2Dlevel,observations%20and%20columns%20of%20variables. [Accessed: 30- Sep- 2020].

[11] Python Software Foundation, "JSON encoder and decoder", *Python 3.8.6 documentation*, 2020. [Online]. Available: <https://docs.python.org/3/library/json.html>. [Accessed: 30- Sep- 2020].

[12] Python Software Foundation, "8.3. collections — Python 2.7.18 documentation", *The Python Standard Library*, 2020. [Online]. Available: <https://docs.python.org/2/library/collections.html>. [Accessed: 30- Sep- 2020].

[13] Spotify AB, "Get an Artist", *Spotify for Developers*, 2020. [Online]. Available: <https://developer.spotify.com/documentation/web-api/reference/artists/get-artist/>. [Accessed: 30- Sep- 2020].