UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS

Materia:
Programación para ciencia de datos

Tema:

Proyecto Final - Análisis de canciones populares en spotify - top 1000

Profesor:
Raul Casillas Figueroa

Alumna:
Diana Cecilia Beltran Garcia

Grupo
440

Ensenada, Baja California a 03 de junio del 2025

Índice

I. Introducción.

Un archivo CSV (valores separados por comas) es un archivo de texto que almacena datos en forma de tabla, donde cada línea representa una fila de datos y cada campo de esa fila está separado por una coma. Los archivos CSV son sencillos y fáciles de usar, lo que los convierte en una opción popular para intercambiar datos entre distintos programas, bases de datos y hojas de cálculo.

Los datos de un archivo CSV se representan como filas y columnas. Las filas están separadas por caracteres de nueva línea y las columnas por comas, lo que le da el nombre de “Valores separados por comas”.

En muchos archivos CSV la primera línea se utiliza como encabezado y contiene los nombres de las columnas. Esto ayuda a los usuarios a comprender el significado de cada columna.

En esta práctica, trabajaremos con el archivo spotify_top_1000_tracks.csv, que contiene información sobre las 1000 canciones más reproducidas en Spotify. A partir de este dataset exploraremos diversas características musicales como la popularidad, duración, artistas, álbumes y tendencias a lo largo del tiempo,

II. Contexto

El análisis se realizó sobre el archivo spotify_top_1000_tracks.csv, que contiene información de 1000 canciones y está estructurado en 8 columnas con datos relevantes para estudiar la popularidad y características musicales en Spotify.

- track_name: Título de la canción.
- artist: Nombre(s) del(los) intérprete(s).
- álbum: Álbum donde fue lanzada originalmente la canción.
- release_date: Fecha de lanzamiento en formato DD-MM-AAAA.
- popularity: Puntuación de popularidad que va de 0 a 100, calculada según reproducciones y actividad en Spotify.
- spotify_url: Enlace directo para acceder a la canción en Spotify.
- id: Identificador único de la canción dentro de la plataforma.
- duration_min: Duración de la pista expresada en minutos (convertida desde milisegundos).

III. Objetivo

Analizar el comportamiento y las características de las canciones más populares en Spotify, utilizando el dataset spotify_top_1000_tracks.csv, con el fin de descubrir patrones relacionados con la popularidad, los artistas, los álbumes y las tendencias musicales.

IV. Herramientas

Las herramientas que se utilizaron para realizar el proyecto final :

- Archivo csv - spotify_top_1000_tracks.csv, extraído de Kaggle
- Visual Studio Code
- Librerías de python (Pandas, Matplotlib y Seaborn)
- Red inalámbrica
- Lenguaje de programación Python
- Dispositivo compatible con los archivos utilizados

V. Desarrollo

Para el análisis se utilizaron las siguientes librerías:

- Pandas: Para la manipulación y limpieza de datos.
- Matplotlib y Seaborn: Para la visualización gráfica de los resultados.

Primero, empleé el método read_csv() de pandas para leer el archivo spotify_top_1000_tracks.csv y convertirlo en un DataFrame. El archivo fue obtenido de la fuente: https://www.kaggle.com/datasets/kunalgp/top-1000-most-played-spotify-songs-of-all-time

```
#%%
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

"""
        CARGAMOS NUESTRO ARCHIVO CSV A UN DATAFRAME CON EL METODO READ  DE PANDAS


"""
spotify_df = pd.read_csv('spotify_top_1000_tracks.csv')

"""
        LIMPIEZA DE NUESTROS DATOS
"""

spotify_df.drop(columns=['id', 'spotify_url'], inplace=True) # Eliminamos las columnas 'id' y 'spotify_url'
```

Durante la limpieza y preparación de los datos, eliminé columnas innecesarias para centrar el análisis en las variables relevantes. Además, convertí la columna release_date a un formato datetime con pandas, y la desglosé en tres nuevas columnas: día, mes y año, para facilitar el análisis temporal.

```
"""
        FORMATO DATETIME
"""
# Convertimos la fecha en formato datetime
spotify_df['release_date'] = pd.to_datetime(spotify_df['release_date'], errors='coerce')

# Creamos las nuevas columnas , year, month, day
spotify_df['year'] = spotify_df['release_date'].dt.year
spotify_df['month'] = spotify_df['release_date'].dt.month
spotify_df['day'] = spotify_df['release_date'].dt.day

# Convertimos  las columnas a enteros, para que se vea mas estetico
spotify_df['year'] = spotify_df['year'].astype('Int64')
spotify_df['month'] = spotify_df['month'].astype('Int64')
spotify_df['day'] = spotify_df['day'].astype('Int64')
```

Para enfocar el análisis y obtener conclusiones claras, planteé las siguientes preguntas de investigación que guiaron el estudio del dataset.

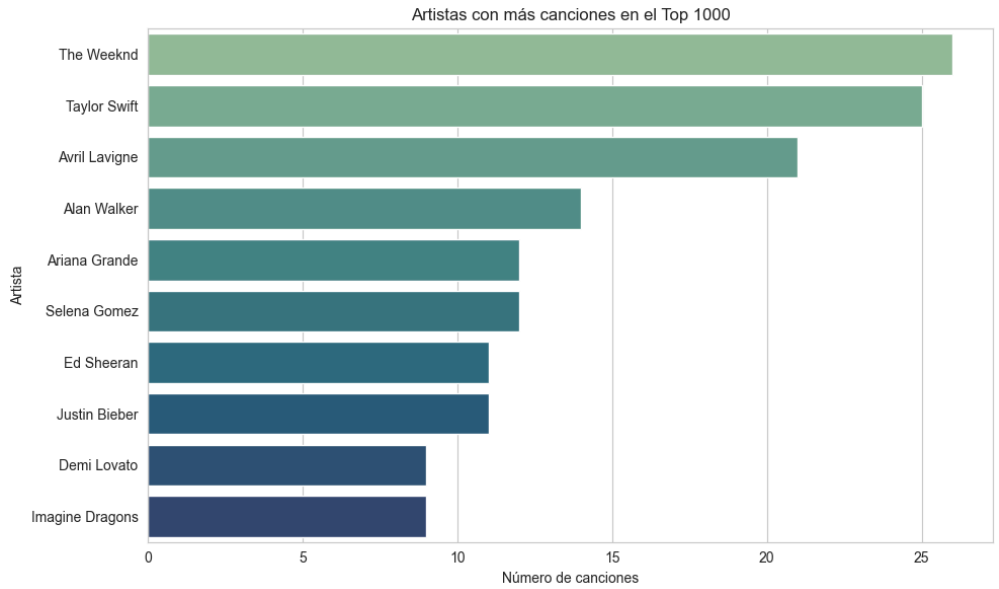
1. **¿Cuáles son los artistas con más canciones en el top 1000?**

Objetivo específico: Identificar y analizar a los artistas que tienen una mayor presencia dentro del top 1000 de canciones más populares en Spotify. Esto permitirá evaluar la influencia que han tenido ciertos artistas en la plataforma.

Código:

```
top_artistas = spotify_df['artist'].value_counts().head(10)
sns.set_style("whitegrid")
plt.figure(figsize=(10,6))
sns.barplot(x=top_artistas.values, y=top_artistas.index, palette="crest")
plt.title("Artistas con más canciones en el Top 1000")
plt.xlabel("Número de canciones")
plt.ylabel("Artista")
plt.tight_layout()
plt.show()
```

Gráfica:



En la gráfica se puede observar que el artista con el mayor número de canciones presentes en el dataset es The Weeknd, con un total de 26 canciones incluidas en el top 1000. Le sigue de cerca Taylor Swift, con 25 canciones en el ranking. En tercer lugar se encuentra Avril Lavigne, con 21 canciones dentro del listado. Este resultado nos muestra el notable dominio de estos artistas en el top 1000.

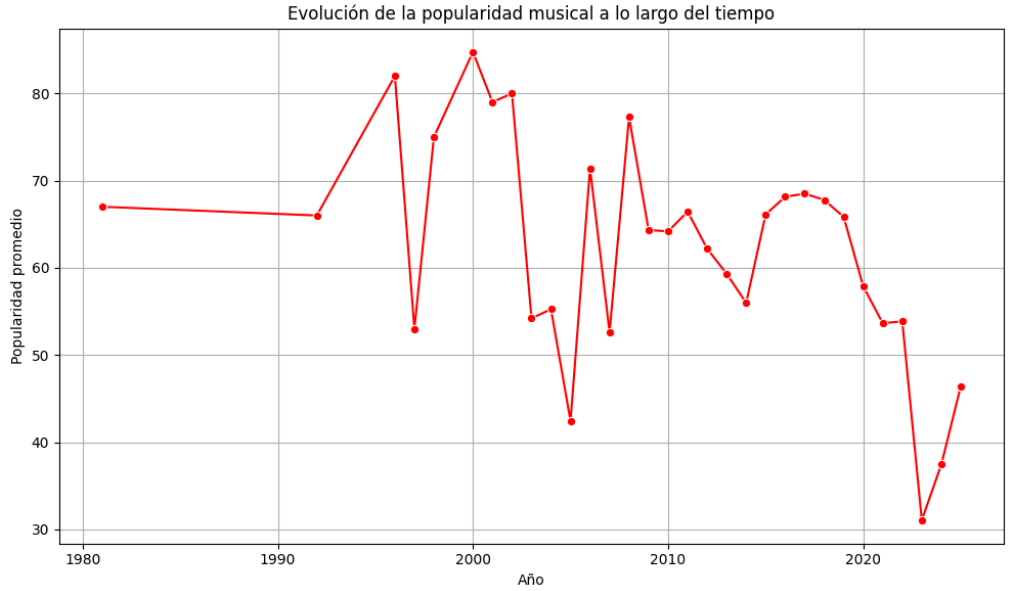
2. ¿Cómo ha cambiado la popularidad de la música a lo largo del tiempo?

Objetivo específico: Analizar cómo ha evolucionado la popularidad de la música a lo largo del tiempo, observando si las canciones más presentes en el top 1000 se concentran en ciertos años.

Código:

```
# Agrupar por año  y con el metodo mean de Pandas calculamos la popularidad promedio

popularidad_por_año = spotify_df.groupby('year')['popularity'].mean().reset_index()

# Ordenarmos los datos por año
popularidad_por_año = popularidad_por_año.sort_values('year')

# Gráfica de línea
plt.figure(figsize=(10,6))
sns.lineplot(data=popularidad_por_año, x='year', y='popularity', marker='o', color='red')
plt.title("Evolución de la popularidad musical a lo largo del tiempo")
plt.xlabel("Año")
plt.ylabel("Popularidad promedio")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Gráfica:



La gráfica muestra que, entre los años 1980 y 1990, la popularidad promedio de las canciones dentro del top 1000 se mantiene alrededor del 68. Con el pico de popularidad de 85 en el año 2000, y se puede ver que a partir del año 2010 hasta 2020, se aprecia una tendencia baja con una disminución en la popularidad promedio que oscila entre 31 y 60.

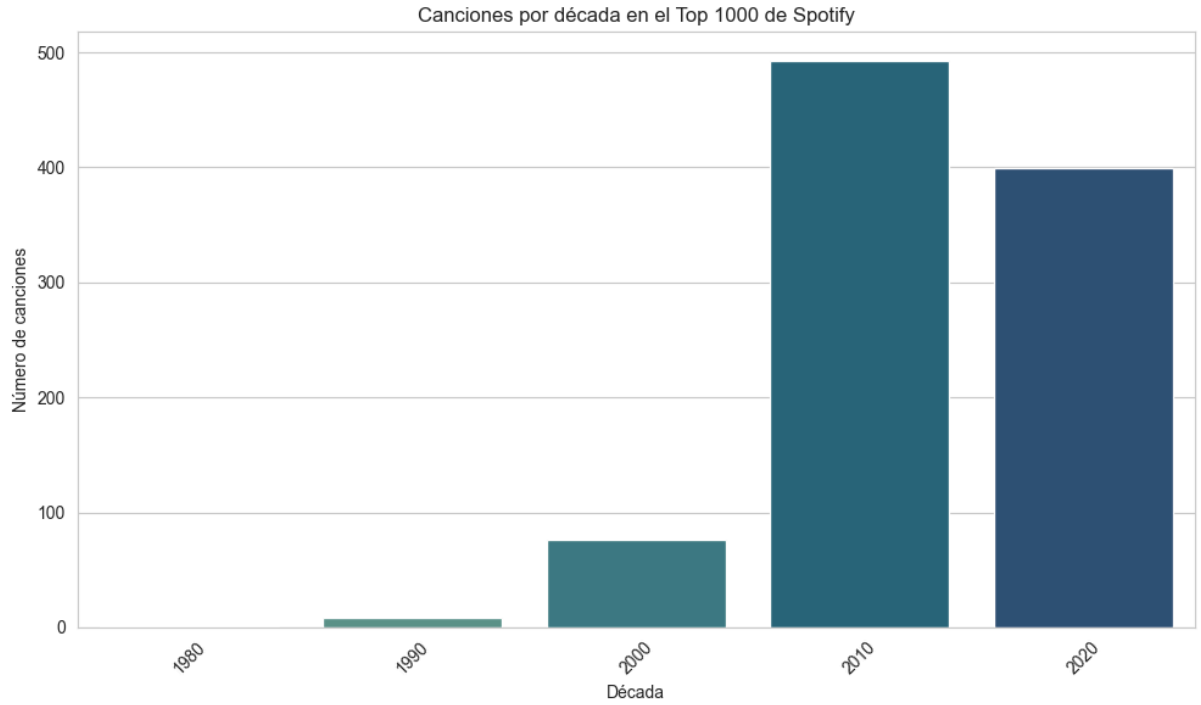
3. ¿Qué décadas están mejor representadas en el Top 1000?

Objetivo específico: Identificar cuales décadas tienen mayor representación en el top 1000, con el fin de determinar qué épocas han sido más exitosas en términos de cantidad de canciones presentes en el top.

Código:

```
#  Creamos una nueva columna  de decada

spotify_df['decada'] = (spotify_df['year'] // 10) * 10

# Contamos las canciones que hay por decada y las ordenamos cronologicamente

canciones_por_decada = spotify_df['decada'].value_counts().sort_index()

#print (canciones_por_decada)       # Imprime el numero total de canciones que hay por decada

# Convertirmos los datos que obtuvimos anteriormente a un DataFrame para graficar, de esta manera es mas sencillo graficar.
#
canciones_por_decada_df = canciones_por_decada.reset_index()   # Creamos nuevos indices
canciones_por_decada_df.columns = ['Decada', 'Cantidad de Canciones']  # Nombramos los indices
#print(canciones_por_decada_df)     # Imprime el indice, la decada y la cantidad de canciones que hay por decada

#Gráfica de barras
plt.figure(figsize=(10,6))
sns.set_style("whitegrid")
sns.barplot(data=canciones_por_decada_df, x='Decada', y='Cantidad de Canciones', palette='crest')
plt.title("Canciones por década en el Top 1000 de Spotify")
plt.xlabel("Década")
plt.ylabel("Número de canciones")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Gráfica:



Se observa que la década de 2010 es la más presente en el top 100, con cerca de 500 canciones. Le sigue la década 2020, con 400 canciones, mientras que la década de 200 solo cuenta con alrededor de 80 canciones.

4. ¿Existe una duración ideal para una canción popular?

Objetivo específico: Analizar si existe una duración promedio o ideal entre las canciones en el top 1000, con el fin de identificar si las canciones tienden a tener una duración similar.

Código:

```
sns.set_style("whitegrid")  # Estilo de la cuadricula
plt.figure(figsize=(10,6))
sns.scatterplot(data=spotify_df, x='duration_min', y='popularity', alpha=0.6)
plt.title('Relación entre duración de la canción y popularidad')
plt.xlabel('Duración (minutos)')
plt.ylabel('Popularidad')
plt.show()

#   Nos muestra cual es la cancion de duracion maxima
max_duration = spotify_df['duration_min'].max()
cancion_mas_larga = spotify_df[spotify_df['duration_min'] == max_duration]
#print(cancion_mas_larga[['artist','track_name', 'duration_min', 'popularity']])

#  Nos muestra cual es la cancion con una duracion minima
min_duration = spotify_df['duration_min'].min()
cancion_mas_larga = spotify_df[spotify_df['duration_min'] == min_duration]
#print(cancion_mas_larga[['artist','track_name', 'duration_min', 'popularity']])

# Mostrar las 10 canciones con mayor duración
top_10_largas = spotify_df.sort_values(by='duration_min', ascending=False).head(10)
print(top_10_largas[['artist', 'track_name', 'duration_min', 'popularity']])
```

Gráfica:

La respuesta es sí, con la gráfica de dispersión podemos observar que, la mayoría de las canciones más populares (con popularidad mayor a 60) tienen una duración de entre 2 y 4 minutos. Esta concentración sugiere que las canciones dentro de este rango tienden a tener mayor éxito.

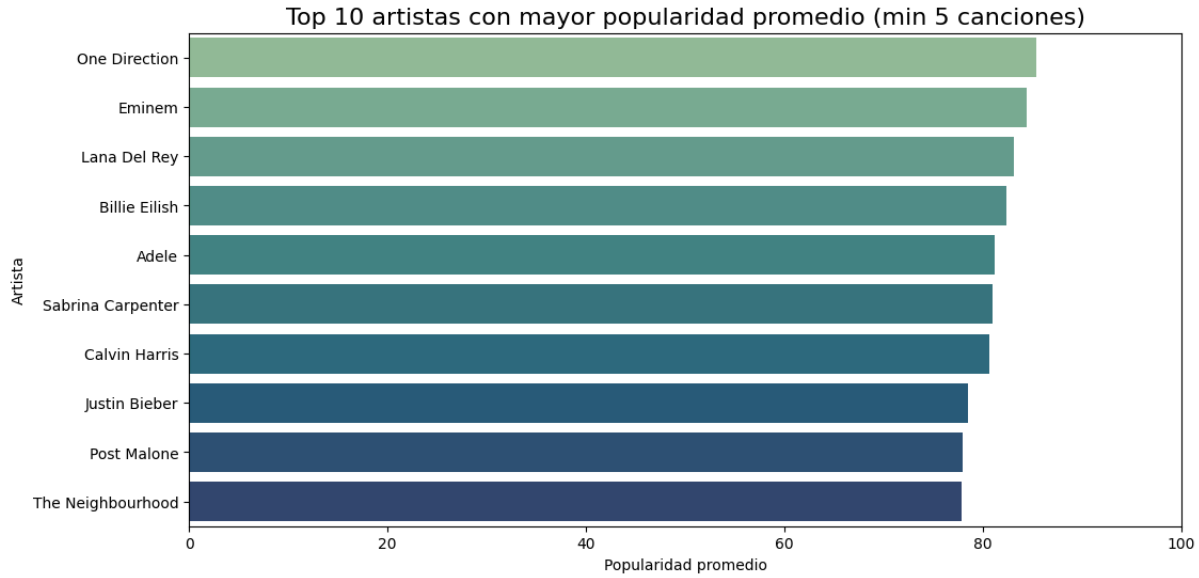
5. **¿Qué artistas tienen la mayor popularidad promedio en sus canciones?**

Objetivo específico: Identificar qué artistas no solo tienen presencia en el top 1000, sino que además destacan por tener una alta popularidad promedio en sus canciones.

Código:

```
# Número mínimo de canciones para filtrar
min_canciones = 5

# Agrupamos por artista, calcular cantidad y popularidad promedio

artistas_pop = spotify_df.groupby('artist').agg(
    canciones=('track_name', 'count'),
    popularidad_promedio=('popularity', 'mean')
).reset_index()

# Filtramos los  artistas con al menos min_canciones canciones
artistas_filtrados = artistas_pop[artistas_pop['canciones'] >= min_canciones]

# Ordenamos  por popularidad promedio descendente y tomar top 10
top10_artistas = artistas_filtrados.sort_values('popularidad_promedio', ascending=False).head(10)

# Gráfica de barras

plt.figure(figsize=(12,6))
sns.barplot(data=top10_artistas, x='popularidad_promedio', y='artist', palette="crest")
plt.title(f'Top 10 artistas con mayor popularidad promedio (min {min_canciones} canciones)', fontsize=16)
plt.xlabel('Popularidad promedio')
plt.ylabel('Artista')
plt.xlim(0, 100)  # popularidad está en rango 0-100
plt.show()
```

Gráfica:



En la gráfica de barras se observa que, los artistas con mayor popularidad promedio (con al menos cinco canciones) son One Direction, Eminem, Lana Del Rey, Billie Eilish, Adele entre otros.

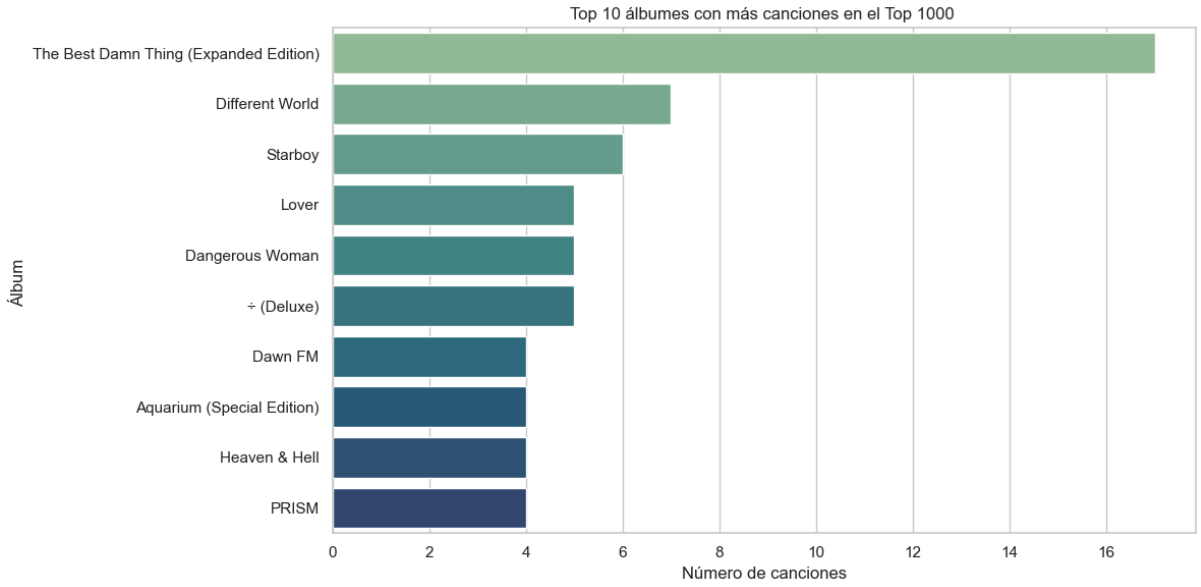
6. ¿Cuáles son los álbumes con más canciones en el top 1000?

Objetivo específico: Identificar cuales álbumes contienen la mayor cantidad de canciones dentro del top 1000, con el fin de destacar aquellos álbumes que han logrado un impacto en términos de popularidad y presencia en el top.

Código:

```
# Agrupamos correctamente por álbum y artista
albumes_artistas = spotify_df.groupby(['album', 'artist']).size().reset_index(name='num_canciones')

# Ordenamos  por número de canciones y tomar el top 10
# Utilizamos el by para indicar que columna deseo ordenar
top10_albumes = albumes_artistas.sort_values(by='num_canciones', ascending=False).head(10)

# Grafica de barras
sns.set_theme(style="whitegrid")  # Estilo cuadricula
plt.figure(figsize=(12,6))
sns.barplot(data=top10_albumes, x='num_canciones', y='album', palette='crest')
plt.title('Top 10 álbumes con más canciones en el Top 1000 ')
plt.xlabel('Número de canciones')
plt.ylabel('Álbum ')
plt.tight_layout()
plt.show()
```

Gráfica:



Con la gráfica de barras se observa que en el primer lugar está el álbum The Best Damn Thing con 18 canciones en el top 1000. Le sigue Different World, con 7 canciones, y en tercer lugar se encuentra Starboy, con 6 canciones en el ranking.

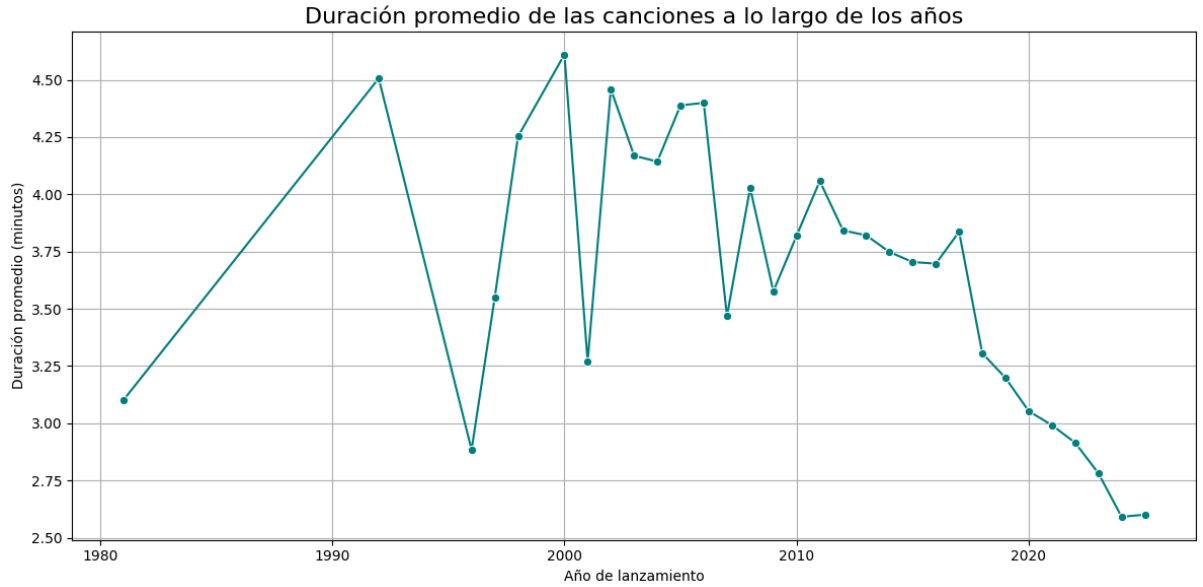
7. **¿Cómo ha cambiado la duración de las canciones a lo largo de los años?**

Objetivo específico: Analizar cómo ha evolucionado la duración promedio de las canciones a lo largo del tiempo, con el fin de detectar si existe una tendencia hacia canciones más cortas o más largas en distintas épocas.

Código:

```
# Agrupamos  por año y calculamos la  duración promedio
duracion_por_ano = spotify_df.groupby('year')['duration_min'].mean().reset_index()

# Grafica de linea
plt.figure(figsize=(12,6))
sns.lineplot(data=duracion_por_ano, x='year', y='duration_min', marker='o', color='teal')
plt.title('Duración promedio de las canciones a lo largo de los años', fontsize=16)
plt.xlabel('Año de lanzamiento')
plt.ylabel('Duración promedio (minutos)')
plt.grid(True)
plt.tight_layout()
plt.show()
```

Grafica de linea :



La gráfica muestra que la duración promedio de las canciones ha variado por década. En los años 80 rondaba los 3 minutos, en los 90 aumentó a más de 4.5 minutos, y en los 2000 comenzó a reducirse. Entre 2010 y 2020, la tendencia se inclinó hacia canciones más cortas, llegando a menos de 3 minutos. Esto muestra una evolución en la estructura musical con el paso del tiempo.

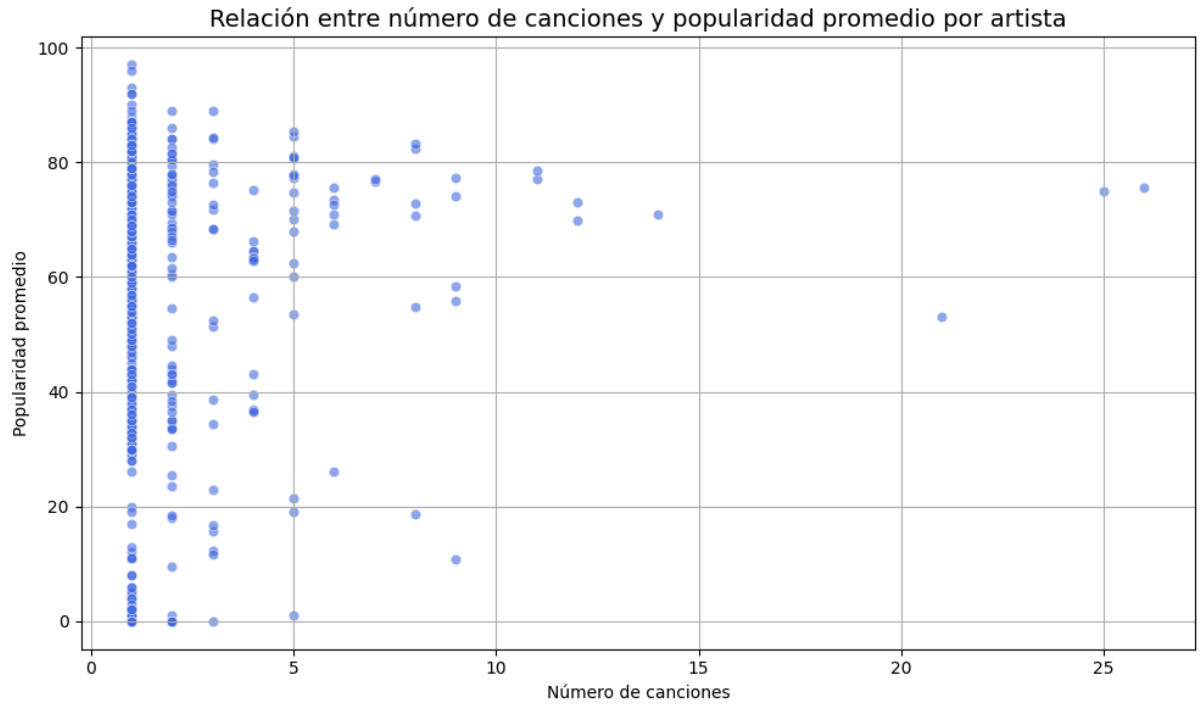
8. ¿Hay relación entre el número de canciones de un artista y su popularidad promedio?

Objetivo específico: Analizar si existe una relación entre la cantidad de canciones que un artista tiene en el top 1000 y su popularidad promedio.

Código:

```
# Agrupamos  por artista y calculamos
artistas_stats = spotify_df.groupby('artist').agg(
    num_canciones=('track_name', 'count'),
    popularidad_promedio=('popularity', 'mean')
).reset_index()

# Grafica de dispersion
plt.figure(figsize=(10,6))
sns.scatterplot(data=artistas_stats, x='num_canciones', y='popularidad_promedio', alpha=0.6, color='royalblue')
plt.title('Relación entre número de canciones y popularidad promedio por artista', fontsize=14)
plt.xlabel('Número de canciones')
plt.ylabel('Popularidad promedio')
plt.grid(True)
plt.tight_layout()
plt.show()
```

Gráfica de dispersión:



La respuesta es no ya que en la gráfica no existe una relación clara entre el número de canciones en el top 1000 y la popularidad promedio. La distribución es dispersa: algunos artistas con muchas canciones tienen una popularidad baja, mientras que otros con pocas canciones logran promedios altos. Esto indica que tener más canciones no garantiza mayor popularidad promedio.

VI. Conclusión

El análisis del dataset spotify_top_1000_tracks.csv permitió identificar patrones relevantes sobre la música popular en Spotify. Se observó que la popularidad de las canciones no depende únicamente de la cantidad de temas que un artista tenga en el ranking, sino también de otros factores como la duración y la época de lanzamiento.

Además, se detectaron tendencias claras en la duración de las canciones a lo largo del tiempo, mostrando una evolución hacia composiciones más cortas en los últimos años.

VII. Referencia

Top 1000 Most Played Spotify Songs of All Time. (2025b, abril 12). Kaggle. https://www.kaggle.com/datasets/kunalgp/top-1000-most-played-spotify-songs-of-all-time