25 de noviembre de 2024

Proyecto en equipo avance 02

Instituto Politécnico Nacional.
Escuela Superior de Cómputo.
Licenciatura en ciencia de datos.

Nombre de la materia: Desarrollo de Aplicaciones para el Análisis de Datos
Grupo: 4AV1
Profesora: Sandra Luz Morales Guitron

Lopez Mendez Emiliano
Leonardo Hinostroza Loera

25 de noviembre de 2024

INDICE

25 de noviembre de 2024

INTRODUCCION

El proyecto busca desarrollar un sistema de recomendación musical que, basado en las tres canciones favoritas del usuario, sugiera cinco canciones adicionales con características similares. Para ello, se aplican técnicas de **ciencia de datos**, como la extracción de características musicales con librosa y la implementación del algoritmo **K-Nearest Neighbors (KNN)** para el modelamiento.

Este proyecto sigue la metodología **CRISP-DM**, que estructura el desarrollo en fases como la comprensión del problema, la preparación de datos y la evaluación del modelo. Además, se utiliza **web scraping** para enriquecer los datos obtenidos con información adicional desde plataformas como Last.fm. El resultado será un sistema capaz de mejorar la experiencia del usuario en plataformas de streaming mediante recomendaciones musicales personalizadas.

DESAROLLO

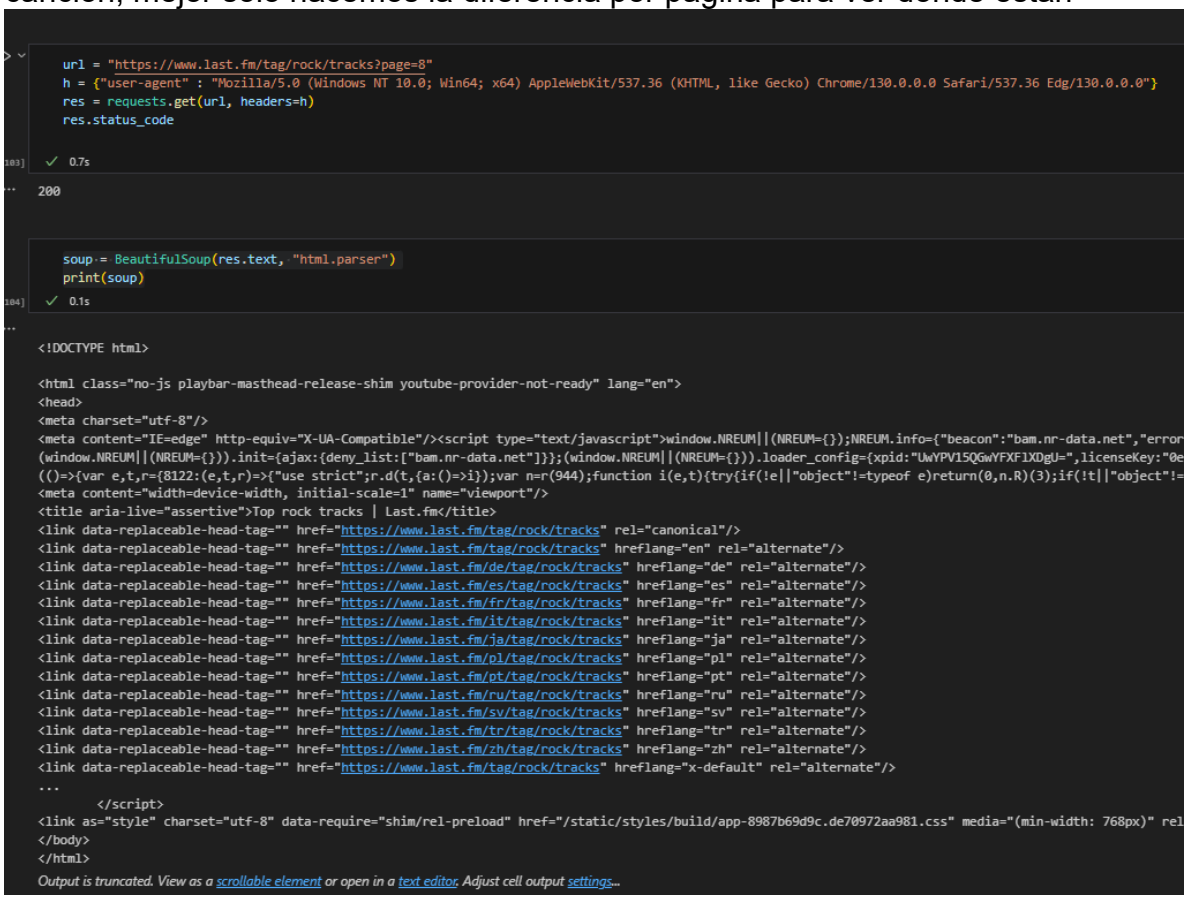
En este código, realizamos un **web scraping** para extraer información de canciones y artistas de la página específica de Last.fm, y posteriormente almacenamos esta información en un formato estructurado mediante un **DataFrame** de pandas. A continuación, se describe brevemente el flujo del proceso:

1. **Obtención de los Datos (Web Scraping):**
    - Utilizamos la librería requests para realizar una solicitud HTTP y acceder al contenido HTML de la página web de Last.fm.
    - Analizamos el contenido HTML con BeautifulSoup para identificar y extraer los elementos correspondientes a las canciones y artistas.
2. **Procesamiento de los Datos:**
    - Los nombres de las canciones y los artistas se extraen utilizando selectores específicos (class='chartlist-name' y class='chartlist-artist').
    - Guardamos cada canción y su artista en una lista de diccionarios, donde cada diccionario representa una fila con las claves "Canción" y "Artista".
3. **Estructuración de los Datos:**
    - Convertimos la lista de diccionarios en un **DataFrame** de pandas, lo que permite manipular los datos de forma más fácil y ordenada.

25 de noviembre de 2024

1. Obtención de los datos

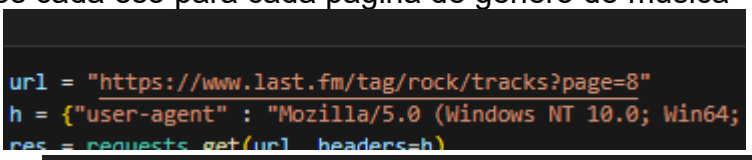
Para cada pagina, necesitábamos buscar en donde se guardan las y los nombres de la canción y del artista, como en los tags no viene el genero de la canción, mejor solo hacemos la diferencia por pagina para ver donde están



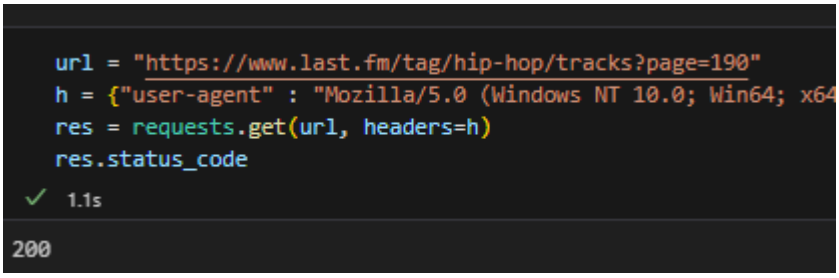
Primero obtenemos el callback del sitio web, 200, significa que si funciona y obtenemos el html.parser

Hicimos cada eso para cada pagina de genero de música

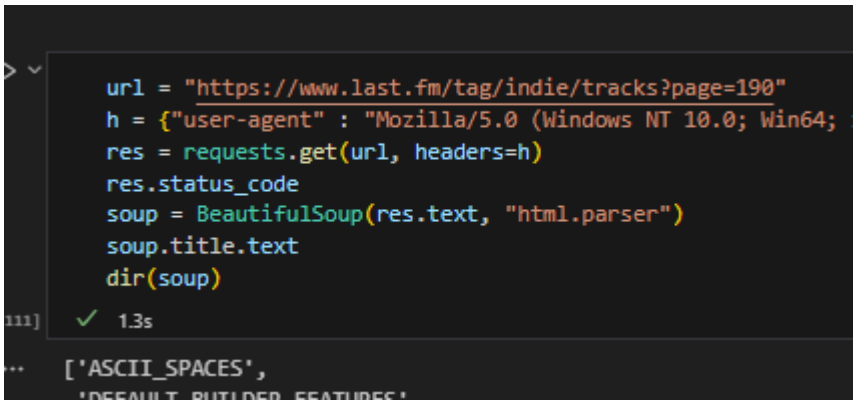
Rock:



Hip-hop:



Indie:

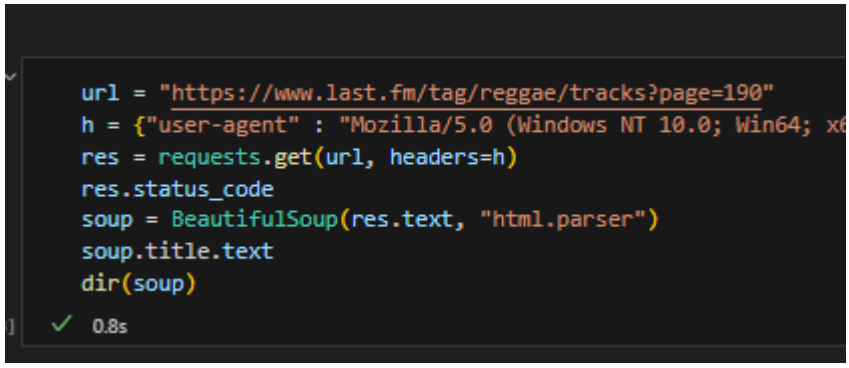


```
url = "https://www.last.fm/tag/indie/tracks?page=190"
h = {"user-agent" : "Mozilla/5.0 (Windows NT 10.0; Win64;
res = requests.get(url, headers=h)
res.status_code
soup = BeautifulSoup(res.text, "html.parser")
soup.title.text
dir(soup)
```

✓ 1.3s

```
['ASCII_SPACES',
```

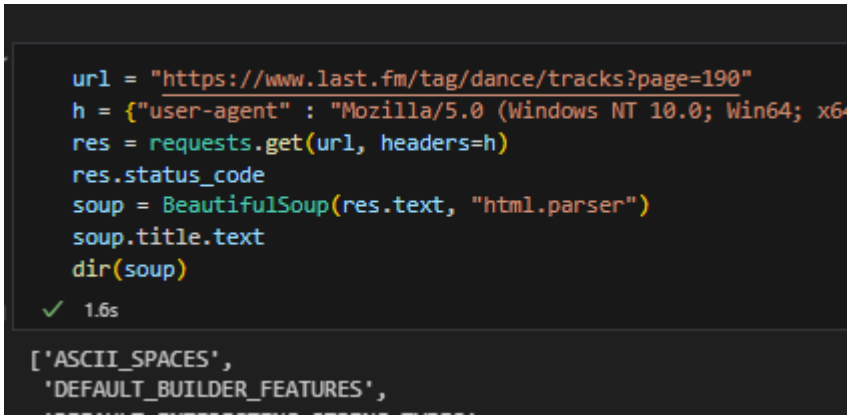
Reggae:



```
url = "https://www.last.fm/tag/reggae/tracks?page=190"
h = {"user-agent" : "Mozilla/5.0 (Windows NT 10.0; Win64; x6
res = requests.get(url, headers=h)
res.status_code
soup = BeautifulSoup(res.text, "html.parser")
soup.title.text
dir(soup)
```

✓ 0.8s

Dance:



```
url = "https://www.last.fm/tag/dance/tracks?page=190"
h = {"user-agent" : "Mozilla/5.0 (Windows NT 10.0; Win64; x6
res = requests.get(url, headers=h)
res.status_code
soup = BeautifulSoup(res.text, "html.parser")
soup.title.text
dir(soup)
```

✓ 1.6s

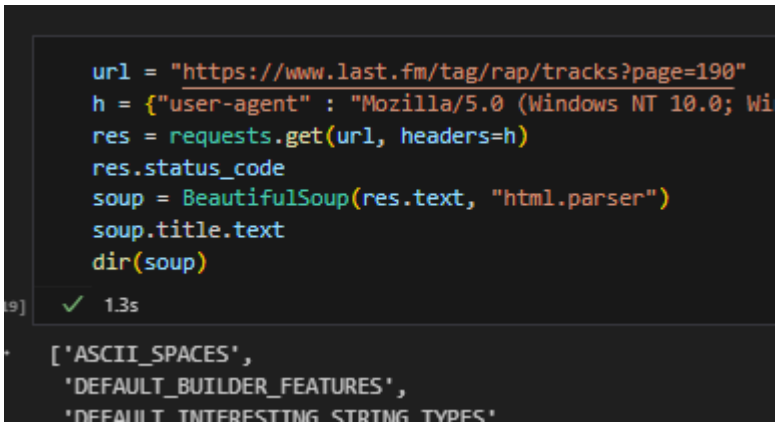
```
['ASCII_SPACES',
 'DEFAULT_BUILDER_FEATURES',
```

Rnb:

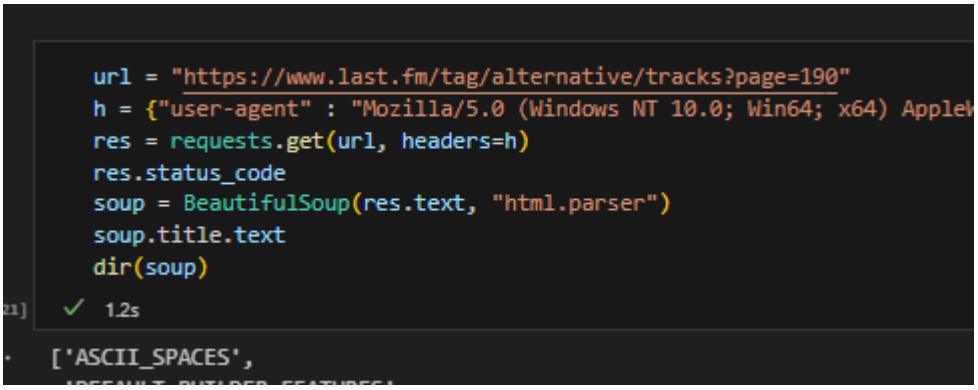
```
url = "https://www.last.fm/tag/rnb/tracks?page=190"
h = {"user-agent" : "Mozilla/5.0 (Windows NT 10.0; Win64;
res = requests.get(url, headers=h)
res.status_code
soup = BeautifulSoup(res.text, "html.parser")
soup.title.text
dir(soup)
```

✓ 0.9s

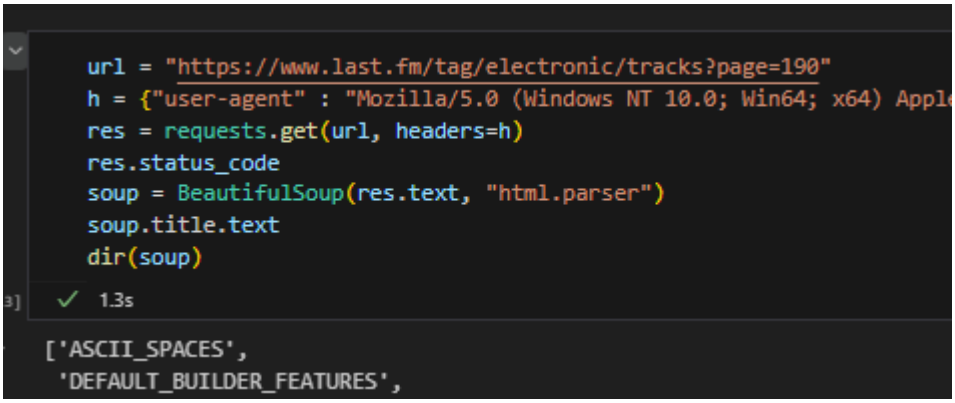
Rap:



Alt:



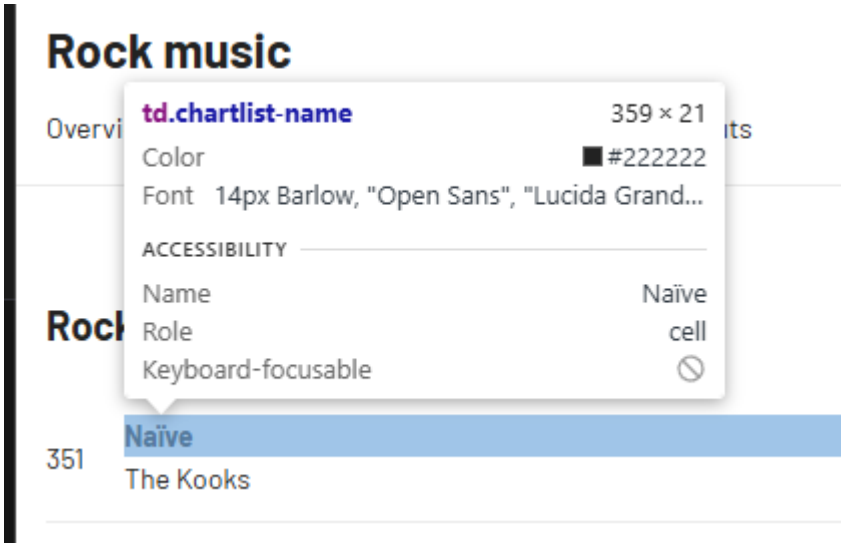
Electronica:



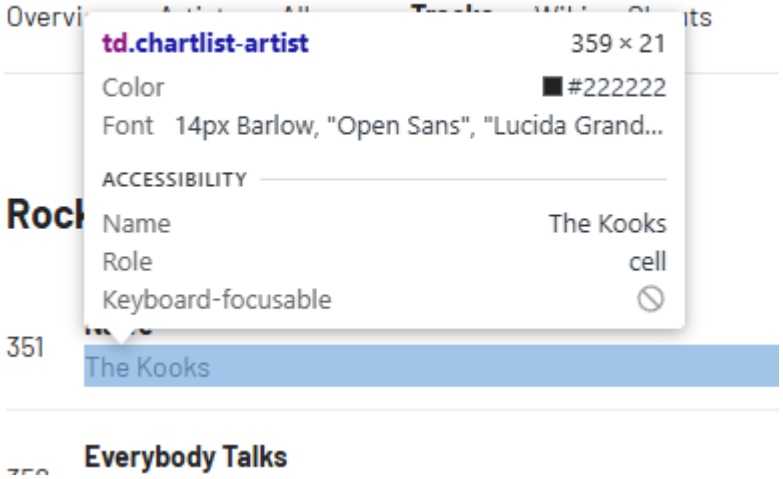
2. **Procesamiento**

   Luego necesitamos ver las etiquetas para cada canción

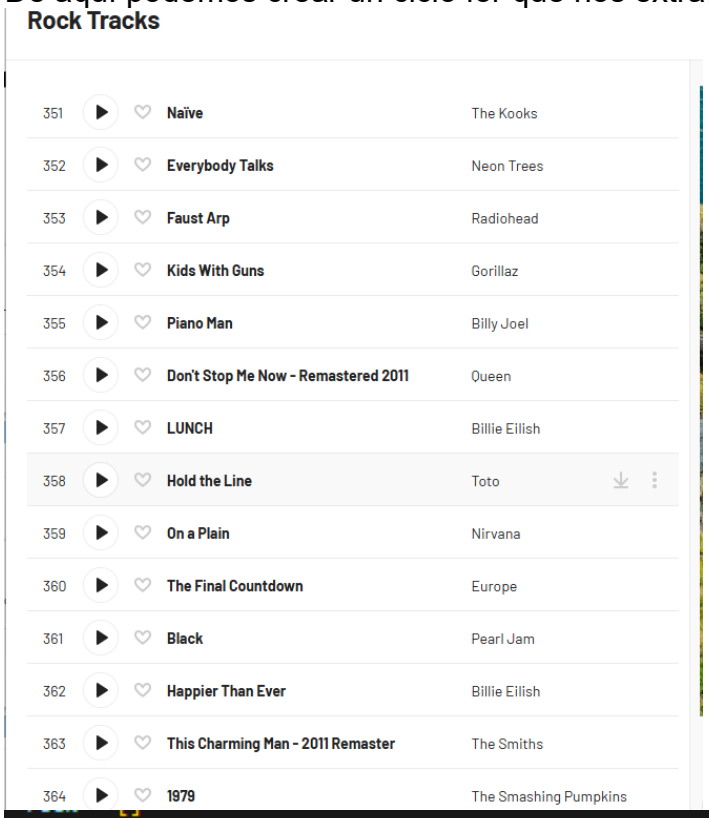
El nombre esta en chartlist-name



El autor esta en chartlist-artist



De aquí podemos crear un ciclo for que nos extrae las canciones de la pagina




```
tracks = soup.find_all('tr', class_='chartlist-row')
for track in tracks:
        # Extraer el nombre de la canción
        song = track.find('td', class_='chartlist-name').find('a').text.strip()

        # Extraer el nombre del artista
        artist = track.find('td', class_='chartlist-artist').find('a').text.strip()

        # Agregar la canción y el artista a la lista
        rock.append({'Canción': song, 'Artista': artist})
```

Extrae, la canción y el nombre del artista, con chartlist-row, se obtiene todas las canciones de esa pagina web y termina hasta que ya no encuentre mas canciones. Hacemos eso para cada genero musical.

3. Estructura

Para mas facilidad de manejar los datos, los manejaremos como dataframes, entonces por cada genero hacemos una lista del genero: por ejemplo rock

```
rock = []
```

y en el ciclo for, se añade a la lista con append usando pandas

```
# Agregar la canción y el artista a la lista
rock.append({'Canción': song, 'Artista': artist})
```

Finalmente convertimos la lista en un dataframe y lo mostramos

```
            rock.append({'Cancion': song, 'Artista': artist})

rock_df = pd.DataFrame(rock)
display(rock_df)
```

✓ 0.0s

	Canción	Artista
0	Naïve	The Kooks
1	Everybody Talks	Neon Trees
2	Faust Arp	Radiohead
3	Kids With Guns	Gorillaz
4	Piano Man	Billy Joel
5	Don't Stop Me Now - Remastered 2011	Queen
6	LUNCH	Billie Eilish
7	Hold the Line	Toto
8	On a Plain	Nirvana
9	The Final Countdown	Europe
10	Black	Pearl Jam
11	Happier Than Ever	Billie Eilish
12	This Charming Man - 2011 Remaster	The Smiths
13	1979	The Smashing Pumpkins
14	Down Under	Men at Work

Así podemos manejar mejor el uso de las canciones

```
hip_hop_df = pd.DataFrame(hip_hop)
display(hip_hop_df)
```

[10] ✓ 0.0s

	Canción	Artista
0	I Need a Girl	P. Diddy
1	A Little Of This	Grand Puba
2	Zucker (feat. Vanessa Mason)	Peter Fox
3	Natural Green	Blazo
4	Werdz From The Ghetto Child	Gang Starr
5	It Was Whatever	Shlohmo
6	Mittrom	Mach-Hommy
7	Knockout	Yung Gravy
8	LA chA TA	f(x)
9	Devil in a New Dress [Feat. Rick Ross]	Kanye West
10	Tyler	yxngxr1
11	Galvanize (feat Q-Tip)	The Chemical Brothers

```
indie_df = pd.DataFrame(indie)
display(indie_df)
```

[12] ✓ 0.0s

	Canción	Artista
0	Personal	Stars
1	She's A Jar	Wilco
2	Darkseid	Grimes
3	New Round	Beck
4	Headache	Girls
5	3 AM	HAIM
6	Oh My God	Ida Maria
7	Made by Maid	Laura Marling

```
reggae_df = pd.DataFrame(reggae)
display(reggae_df)
```

114] ✓ 0.0s

	Canción	Artista
0	Spy Fi Die	Bounty Killer
1	Natural Music	Bob Marley
2	Vara nu dorm	Connect-R
3	Love You	Jah Cure
4	What Can I Say	The Tartans
5	K.S. Karkonosze	Leniwiec
6	Listen To DJ's	Long Beach Dub All Stars
7	1-2-3 Version	Augustus Pablo
8	Soulful I	The Upsetters
9	Neurostamos	Cheb Balowski
10	How Many People	Ayo

```
dance_df = pd.DataFrame(dance)
display(dance_df)
```

116] ✓ 0.0s

	Canción	Artista
0	Chronicles of a Fallen Love	The Bloody Beetroots
1	Dynomite	Ima Robot
2	Girls	Sugababes
3	Bobblehead	Christina Aguilera
4	Set Fire to the Rain (Moto Blanco edit)	Adele
5	Nothing Left To Lose	Everything But the Girl
6	Lucky Star (featuring Dizzee Rascal)	Basement Jaxx
7	I Wanna Be Software	Grimes
8	Touch My Body	Slayyyter
9	Goin' Crazy	Ashley Tisdale
10	Feel Good Hit of the Fall	!!!

```
        rnb.append({'Canción': song, 'Artista': artist})

rnb_df = pd.DataFrame(rnb)
display(rnb_df)
```

18] ✓ 0.0s

	Canción	Artista
0	Drunk	Tweet
1	Have You Ever Loved Somebody	Freddie Jackson
2	Airplane	Magdalena Bay
3	Girl Don't Take Your Love From Me	Michael Jackson
4	Stuttering	Fefe Dobson
5	Live For (feat. Drake)	The Weeknd
6	Party Boy	Boris

```
        rap.append({'Canción': song, 'Artista': artist})

rap_df = pd.DataFrame(rap)
display(rap_df)
```

120] ✓ 0.0s

	Canción	Artista
0	Runaway Love	Ludacris
1	Gone	TobyMac
2	123	Smokepurpp
3	Hate	JAY-Z
4	Hold My Hand - Duet with Akon	Michael Jackson
5	Dear Sirs	El-P
6	Winter Blues	JJ DOOM
7	Not Tonight	Lil' Kim
8	2 Vaults (feat. Lil Yachty)	Tee Grizzley

```
        alt.append({'Canción': song, 'Artista': artist})

alt_df = pd.DataFrame(alt)
display(alt_df)
```

122] ✓ 0.0s

	Canción	Artista
0	Elevator Music	Beck
1	Morning Theft	Jeff Buckley
2	Extra Kings	The Avalanches
3	LIFE AFTER SALEM	Lil Nas X
4	Elephant Stone	The Stone Roses
5	Pretty	The Weeknd
6	Don't Leave	Snakehips

```
elc_df = pd.DataFrame(elc)
display(elc_df)
```

124] ✓ 0.0s

	Canción	Artista
0	Red-Eye	The Album Leaf
1	Lithium	Labyrinth Ear
2	Dark Lady	DJ Food
3	Black Jesus † Amen Fashion	Lady Gaga
4	Who Gon Stop Me	JAY-Z & Kanye West
5	In the Summer	Crystal Fighters
6	Chaos Lives in Everything	Korn
7	Cish Cash	Basement Jaxx

Se hizo para cada genero de musica

CONCLUSIONES

El desarrollo de este proyecto permitió implementar técnicas fundamentales de ciencia de datos para extraer y organizar información musical de manera automatizada. A través del uso de web scraping, se logró recolectar datos detallados de canciones y artistas desde múltiples páginas de Last.fm, estructurándolos eficientemente en DataFrames de pandas. Este proceso no solo facilitó el manejo de grandes volúmenes de información, sino que también sentó las bases para un análisis más profundo y personalizado.

Además, el proyecto reforzó el uso de herramientas clave como requests para realizar solicitudes HTTP, BeautifulSoup para analizar contenido HTML y pandas

para organizar y almacenar los datos. Esta metodología asegura que el sistema de recomendación musical pueda aprovechar datos fiables y estructurados para ofrecer resultados precisos y relevantes.

El trabajo realizado es un ejemplo práctico del impacto que tienen las técnicas de programación y análisis de datos en la generación de soluciones personalizadas. Este avance representa un paso importante hacia la construcción de un sistema de recomendación musical robusto y funcional, destacando el potencial de la automatización y modelado de datos en aplicaciones del mundo real.