

documentación y guía de la demo

Contenido

• preparación del entorno y levantar Docker.....	3
• Distintos servicios	4
Capturas de cada servicio.....	4
• Prefect.....	4
• WEB	4
• API.....	5
• Guía de la demo.....	5
• Gráfico de géneros.....	7
• Gráfico de Popularidad.	8
• Recomendaciones en base a gustos.....	8

- preparación del entorno y levantar Docker

Lo primero para la demo es clonar o bajarse el repositorio:

https://github.com/hfernandezp-dev/Proyecto_de_big_data_e_inteligencia_artificial

Con la entrega de la demo viene el archivo privado .env ya que no se subirá al repositorio por seguridad, se debe copiar en la carpeta environment.

Una vez hecho este proceso y tener Docker en tu equipo si es en Windows debe ser Docker desktop siendo en Linux es instalar Docker como tal, una vez instalado se abre el proyecto en terminal y se ejecuta el siguiente comando “docker-compose up” mostrando lo siguiente:

```
PS C:\Users\pimie\Desktop\Proyecto_de_big_data_e_inteligencia_artificial-Fase4> docker-compose up
[+] Running 4/4
  ✓ Network proyecto_de_big_data_e_inteligencia_artificial-fase4_default      Created
  ✓ Container proyecto_de_big_data_e_inteligencia_artificial-fase4-prefect-1    Created
  ✓ Container proyecto_de_big_data_e_inteligencia_artificial-fase4-web-1       Created
  ✓ Container proyecto_de_big_data_e_inteligencia_artificial-fase4-api-1       Created
Attaching to api-1, prefect-1, web-1
web-1 | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
web-1 | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
web-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
web-1 | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
web-1 | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
web-1 | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
web-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
web-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
web-1 | /docker-entrypoint.sh: Configuration complete; ready for start up
web-1 | 2025/12/30 16:06:30 [notice] 1#1: using the "epoll" event method
web-1 | 2025/12/30 16:06:30 [notice] 1#1: nginx/1.29.4
web-1 | 2025/12/30 16:06:30 [notice] 1#1: built by gcc 15.2.0 (Alpine 15.2.0)
web-1 | 2025/12/30 16:06:30 [notice] 1#1: OS: Linux 6.6.87.2-microsoft-standard-WSL2
web-1 | 2025/12/30 16:06:30 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
```

Si es la primera vez que se hace tardara mas debido a la instalación de paquetes del archivo requirement en la carpeta environment. A continuación, muestro el resultado del comando:

```
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 30
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 31
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 32
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 33
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 34
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 35
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 36
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 37
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 38
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 39
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 40
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 41
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 42
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 43
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 44
web-1 | 2025/12/30 16:06:30 [notice] 1#1: start worker process 45
api-1 | [prefect-1] 2025/12/30 16:06:30 [INFO] Web server listening in these directories: [/app]
api-1 | [prefect-1] 2025/12/30 16:06:30 [INFO] Application running on https://0.0.0.0:8080 (Press CTRL+C to quit)
api-1 | [prefect-1] 2025/12/30 16:06:30 [INFO] Started reloader process [1] using StatReload
api-1 | [prefect-1] 2025/12/30 16:06:30 [INFO] Started server process [8]
api-1 | [prefect-1] 2025/12/30 16:06:30 [INFO] Waiting for application startup.
api-1 | [prefect-1] 2025/12/30 16:06:30 [INFO] Application startup complete.
prefect-1 |
prefect-1 |   _____
prefect-1 |  /     \ \
prefect-1 |  \     / \
prefect-1 |   \   /   \
prefect-1 |     \   /
prefect-1 |       \ /
prefect-1 |         |
prefect-1 | Configure Prefect to communicate with the server with:
prefect-1 |
prefect-1 |   prefect config set PREFECT_API_URL=http://0.0.0.0:4200/api
prefect-1 |   View the API reference documentation at http://0.0.0.0:4200/docs
prefect-1 |   Check out the dashboard at http://0.0.0.0:4200
prefect-1 |
prefect-1 | /usr/local/lib/python3.10/contextlib.py:142: SAWarning: Skipped unsupported reflection of expression-based index ix_flow_run__coalesce_start_time_expected_start_time_desc
prefect-1 |   next(self.gen)
prefect-1 | /usr/local/lib/python3.10/contextlib.py:142: SAWarning: Skipped unsupported reflection of expression-based index ix_flow_run__coalesce_start_time_expected_start_time_asc
prefect-1 |   next(self.gen)
```

View in Docker Desktop View Config Enable Watch

- Distintos servicios

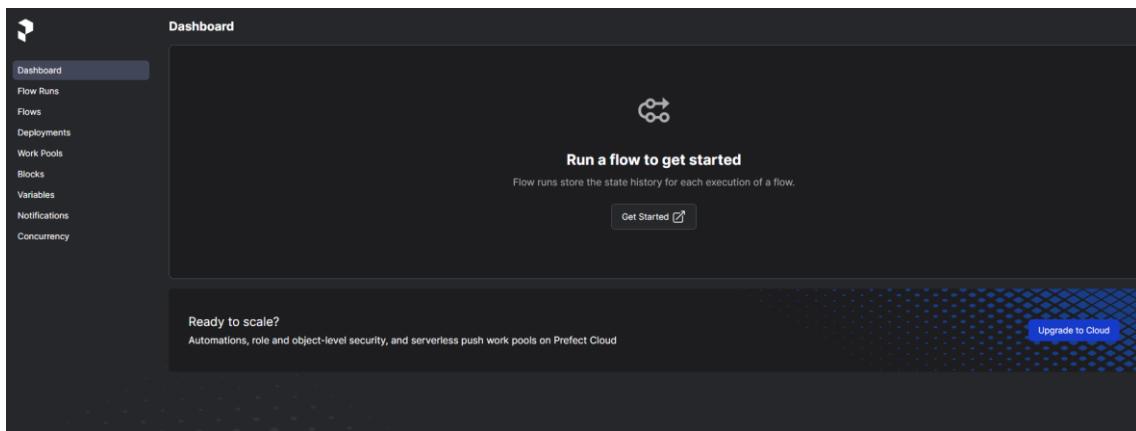
Una vez ejecutado el comando se levantarán 3 servicios, la API con puerto 8000:8000, la web con puerto 8080:80y el servidor de Prefect con puerto 4200:4200, debido a problemas con la demo Prefect falla en la ejecución periódica.

Rutas para acceder:

- Prefect: <http://localhost:4200/dashboard>
- Web: <http://localhost:8080>
- API: <http://localhost:8000> o <http://localhost:8000/docs>

Capturas de cada servicio.

- Prefect



- WEB



- API

API de Recomendación Musical 0.5.0 OAS 3.1

[/openapi.json](#)

API para recomendar canciones usando clustering KMeans

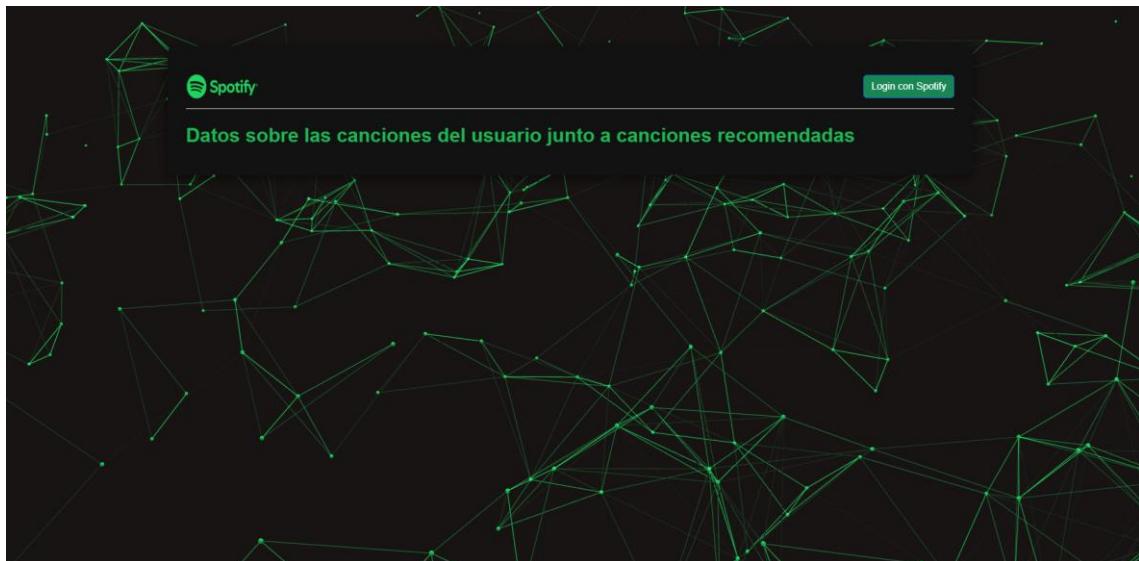
[Authorize](#)

default

<code>GET</code>	/login Login	
<code>GET</code>	/runflow ejecuta el flow una vez acabado	
<code>GET</code>	/callback Genera el accesoToken para consumir la API	
<code>GET</code>	/api/recent-tracks Recupera las ultimas 50 canciones	
<code>GET</code>	/api/generos Recupera los generos de los grupos	
<code>GET</code>	/api/conseguir_canciones Consigue las canciones	
<code>GET</code>	/obtener_datos_emul Simula la entrada de datos ya sea para un dashboard o un aplicacion	
<code>GET</code>	/enviar_datos_emul Simula el envio de datos ya sea para un dashboard o un aplicacion	
<code>POST</code>	/prediccion Predice una serie de 5 canciones del mismo cluster	
<code>GET</code>	/health Health	

- Guía de la demo

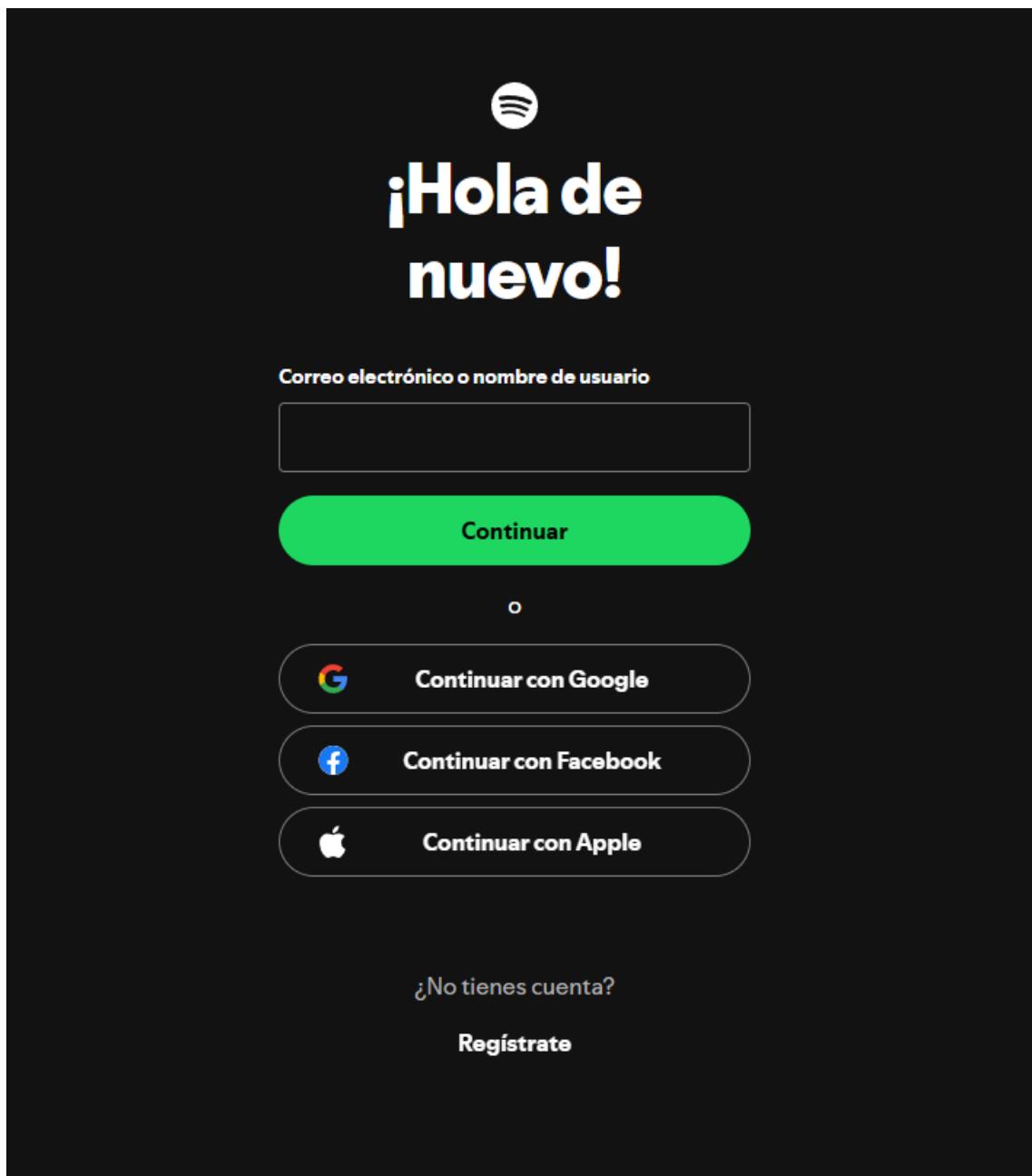
A continuación, se va a explicar el proceso de la demo y posibles fallas. Una vez iniciado el servicio usando Docker y entrando a la web con <http://localhost:8080>, lo primero que se verá es:



Esta todo oculto hasta que se empiece a cargar los gráficos y las recomendaciones, debes darle al botón de “Login con Spotify”.

Login con Spotify

Una vez pulsado se abrirá en otra pestaña la web de Spotify para iniciar sesión con tu cuenta.



En caso de que sea la primera vez que se hace el proceso, una vez iniciada la sesión te pedirá que confirmes si se permite el uso de datos, una vez aceptada se dirigira a una web que estará vacia pero con la URL. /callback

127.0.0.1/callback

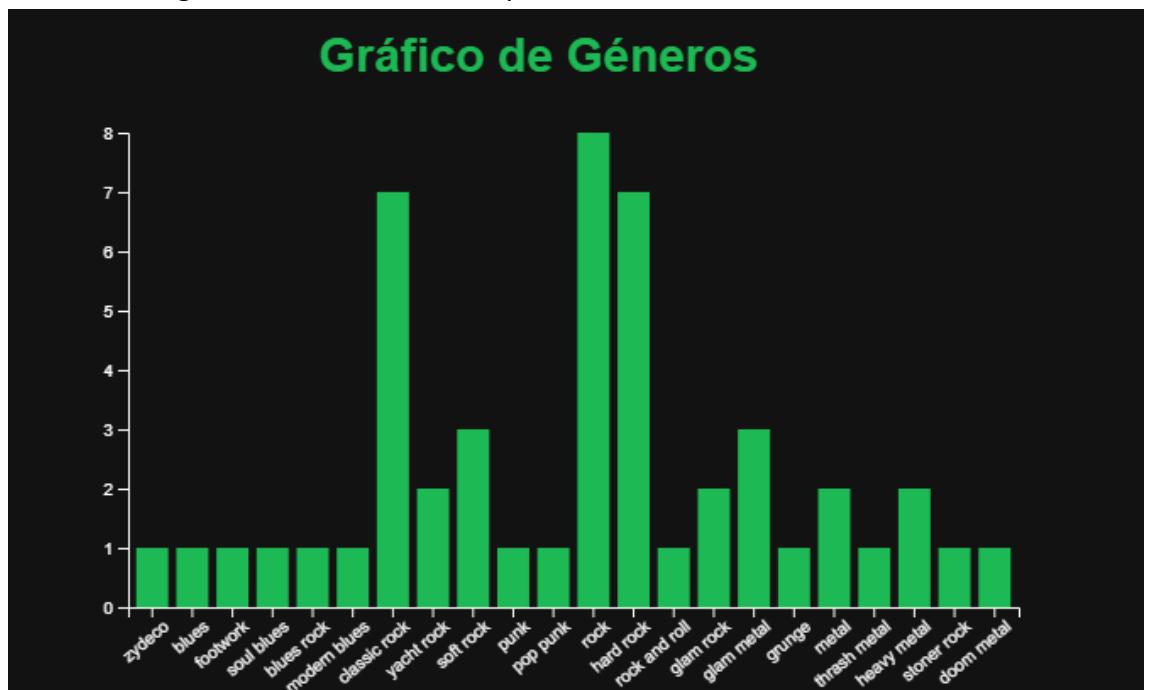
Dentro de la propia URL hay una parte con “code=xxxxx”, se deberá copiar ese código y cerrar esta pestaña, una vez se vuelva a la web se deberá poner el código en el campo que se ha habilitado a la izquierda del botón comentado anteriormente.



Una vez pulsado Enviar el token este se enviara al servicio API para así recuperar las ultimas 50 canciones que ha reproducido el usuario mostrando tres partes importantes.

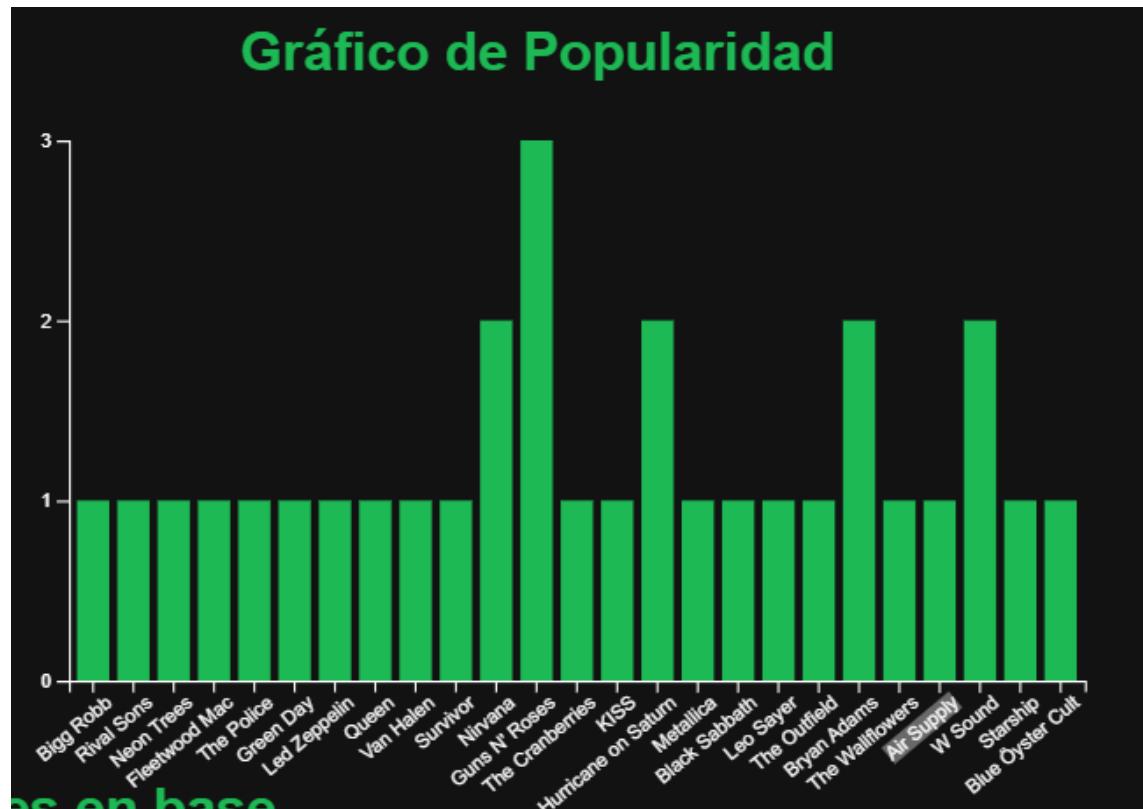
- Gráfico de géneros.

Muestra los géneros mas escuchados por el usuario.



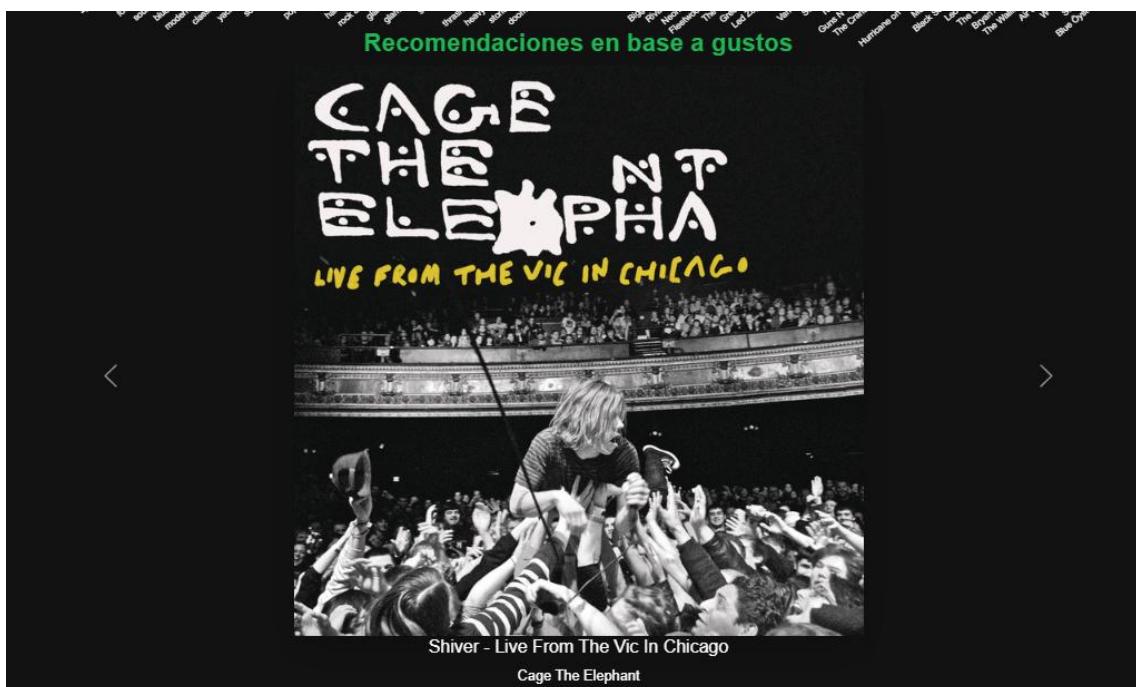
- Gráfico de Popularidad.

Muestra los grupos más escuchados.

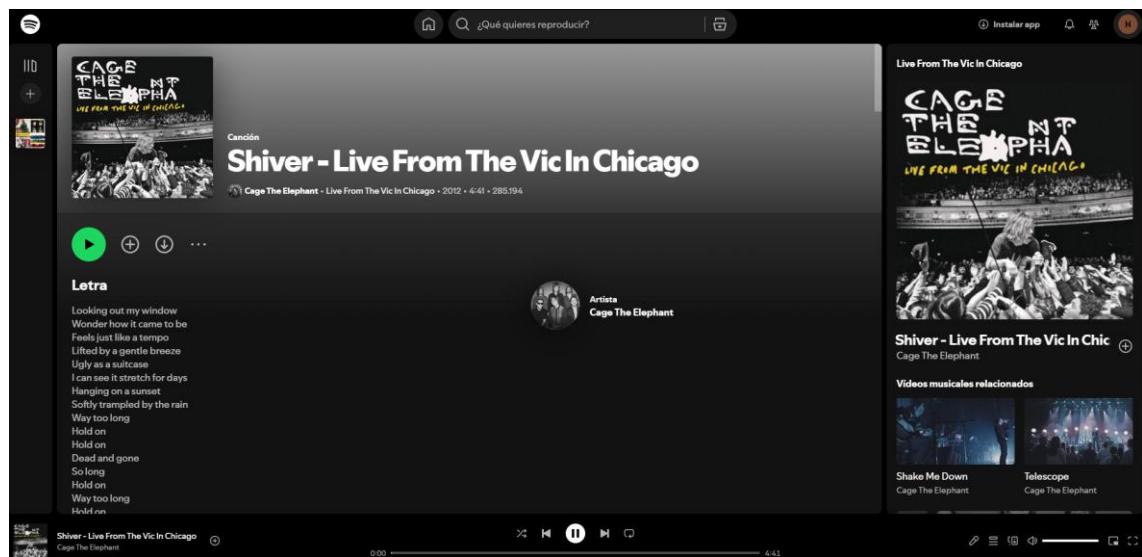


- Recomendaciones en base a gustos.

Muestra una serie de 5 canciones recomendadas por el modelo ya entrenado.



Una vez pulsado en las caratulas de las canciones se abrirá una pestaña en la web de Spotify.



Esto es lo ultimo en la api solo que una vez llegado a este ultimo punto si Spotify permitiese el entrenamiento de modelos con sus datos se recogerían los datos del usuario y se usarían para entrenar mas el modelo de recomendación.