

MUSICSTREAM

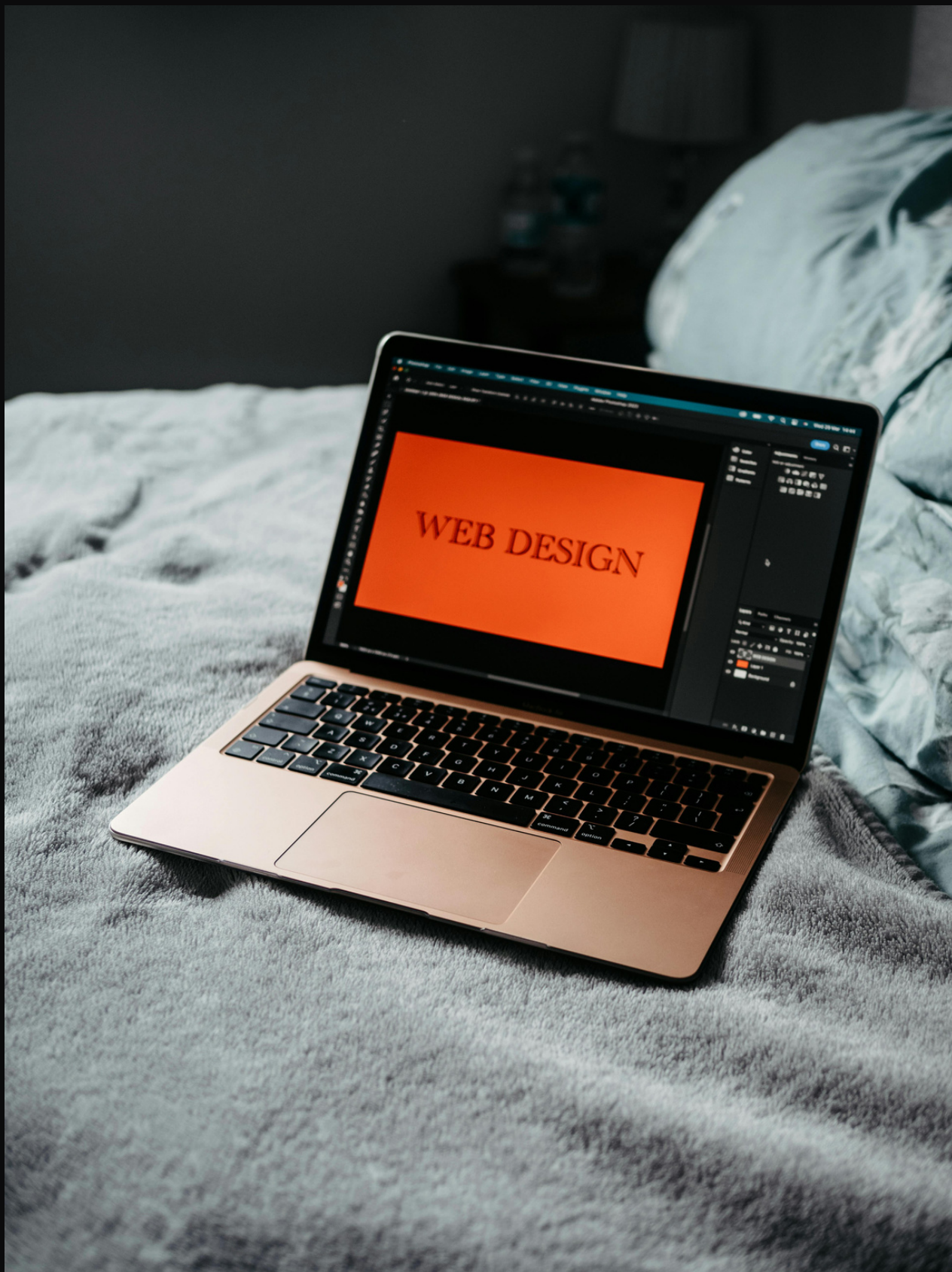
MUSICSTREAM

MUSICSTREAM

MUSICSTREAM

MUSIC DATA ON DEMAND, ANYTIME, ANYWHERE

MUSICSTREAM MUSICSTREAM MUSICSTREAM MUSICSTREAM MUSICSTREA



MusicStream

UN SERVICIO QUE PERMITE EL ANÁLISIS
Y DESCUBRIMIENTO DE MÚSICA,
IMPULSADA POR DATOS EN TIEMPO
REAL DE SPOTIFY Y LAST.FM.



Project Overview

Project Overview

Project Overview

Project Overview



NUESTRO PROPÓSITO

- Crear un servicio que permita explorar y descubrir nuevos patrones de escucha
- Analizar tendencias musicales
- Comparación de popularidad
- Insights sobre el comportamiento de los oyentes
- Unificar datos (canciones, artistas, popularidad, 'scrobbles')

“



SPOTIFY API

Provee datos estructurales: catálogo de canciones, artistas, álbumes, y métricas de popularidad oficiales.

“



LAST.FM API

Aporta datos de 'scrobbling': canciones más escuchadas, etiquetas de género y tendencias de los usuarios.

“



SQL (DESTINO)

Los datos se limpian y almacenan en una base de datos SQL relacional para consultas eficientes y análisis.

DE DATOS A RITMO. CONVERTIMOS CÓDIGO EN CANCIONES.

En MusicStream convertimos los datos de Spotify y Last.fm en inteligencia musical personalizada.

Ayudamos a nuestros clientes a conectar con su público mediante playlists creadas a partir de métricas reales y tendencias musicales auténticas.

- *Playlists inteligentes, no algoritmos aburridos.*
- *Datos reales + criterio humano = música con propósito.*
- *Tendencias y géneros emergentes, antes que nadie.*
- *Recomendaciones únicas, diseñadas para cada cliente.*



EXTRACCIÓN AUTOMATIZADA DESDE SPOTIFY API (PYTHON + SPOTIPY)



```
auth_manager = SpotifyClientCredentials(
    client_id='...', client_secret='...')
sp = spotipy.Spotify(auth_manager=auth_manager)
```

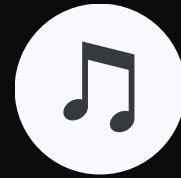
```
lista_canciones = [] # Lista general

for g in generos:
    genres_dic = {} # Contador por género
    for año in años:
        for o in myoffset:
            datos = sp.search(
                q=f"genre:{g}, year:{año}",
                type="track", limit=50, offset=o
            )
            tracks = datos.get('tracks', {}).get('items', [])

            for track in tracks:
                cancion = {
                    "id": track['id'],
                    "track_name": track['name'],
                    "artist_name": track['artists'][0]['name'],
                    "genre": g,
                    "year": año,
                    "album_type": track['album']['album_type'],
                    "release_date": track['album']['release_date'],
                    "popularity": track['popularity']
                }
                lista_canciones.append(cancion)
                genres_dic[track['id']] = cancion

print(f"Total canciones para {g}: {len(genres_dic)}")
```

Cada línea de código se convierte en una nota del análisis.



Desde Jupyter Notebook automatizamos la conexión con Spotify para obtener información de canciones, artistas y métricas de popularidad. Estos datos se limpian y estructuran para combinarse con Last.fm, generando un perfil musical completo por artista.



```
df = pd.DataFrame(lista_canciones) # Transformación en DataFrame
df.to_csv("Canciones_Spotify.csv", index=False) # Extracción de datos a CSV

df_spotify = pd.read_csv("Canciones_Spotify.csv") # Lectura de CSV
df_spotify.head()
```



	id	track_name	artist_name	year	genre	album_type	release_date	popularity
0	1WHNqqRWhJVZldCScFKtI5	Washington on Your Side	Leslie Odom Jr.	2015	soundtrack	album	2015-09-25	66
1	1DLfR4MOflYbV6v3xrmWa8	We Know	Lin-Manuel Miranda	2015	soundtrack	album	2015-09-25	66
2	2AtC6i0b8TjpjhWBZYLprX	Bonetrousle	Toby Fox	2015	soundtrack	album	2015-09-15	58
3	6oF8ueLn5hII4PRp17sxW6	That Would Be Enough	Phillipa Soo	2015	soundtrack	album	2015-09-25	67
4	46YSff2Rq1ZtN1YVvK5cwbZ	Brave Shine	Aimer	2015	soundtrack	album	2015-07-29	61



EXTRACCIÓN AUTOMATIZADA DESDE LASTFM API

- Se crea la función **get_artist_info** en Python para conectarse a la API de LastFM.
- Extrae información detallada sobre cada artista obtenido desde Spotify (lista de valores únicos).
- Realiza una solicitud a la API y procesa la respuesta en formato JSON.
- Obtiene datos como la biografía, oyentes, reproducciones y un artista similar.
- En caso de error, muestra un mensaje informativo; si es exitosa, devuelve la información en un diccionario listo para su uso o visualización.

```
df_spotify = pd.read_csv("Canciones_Spotify.csv") # Transformación en DataFrame del csv de Spotify.

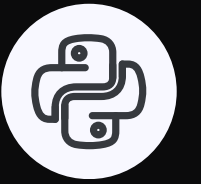
df_spotify['artist_name'].tolist() # Creación de la lista de artistas para buscar en la función get_artist_info.

unique_artists_spotify = df_spotify['artist_name'].unique().tolist() # Generación de valores únicos.

lista_artist_lastfm_NN = [] # Variable para almacenar toda la información de todos los artistas que no sean nulos.

for X in unique_artists_spotify:
    info = get_artist_info(X)
    if info is not None: # Solo agrega si tiene datos
        lista_artist_lastfm_NN.append(info)
    time.sleep(0.5)
```

Función **get_artist_info**:



```
def get_artist_info(artist_name):
    API_KEY = my_API_KEY
    BASE_URL = "https://ws.audioscrobbler.com/2.0/" # url de conexión a la API de LastFM.

    params = {
        "method": "artist.getinfo",
        "artist": artist_name,
        "api_key": API_KEY, # Parametros dados a la API para la extracción
        "format": "json", # en el endpoint "artist.getinfo".
        "autocorrect": 1,
        "lang": "en"
    }

    response = requests.get(BASE_URL, params=params)
    data = response.json()

    # Validar si hay error
    if "error" in data:
        print(f"❌ Error: {data['message']}")
        return None

    artist = data["artist"]
    listeners = artist["stats"]["listeners"]
    playcount = artist["stats"]["playcount"]
    similar_artists = [a["name"] for a in artist["similar"]["artist"]]
    similar_artist = similar_artists[0] if similar_artists else None
    # Información de cada canción según los parametros explícitos.

    bio = data.get("artist", {}).get("bio", {})
    summary = bio.get("summary", {})

    return {
        "artist_name": artist["name"],
        "biography": summary,
        "listeners": listeners,
        "playcount": playcount,
        "similar_artist": similar_artist
    } # Formato para mostrar la información.
```

MEJORA DE LA BBDD DE ARTISTAS OBTENIDA

Transformación de la lista de artistas únicos a DataFrame y CSV.

```
df_lastfm = pd.DataFrame(lista_artist_lastfm_NN) # Transformación en DataFrame.  
df_lastfm.to_csv("Artistas_LastFM.csv", index=False) # Conversión de LastFM a CSV.
```

	artist_name	biography	listeners	playcount	similar_artist
1188	Praiz	Praise Ugbede Adejo (born 8 March 1984), bette...	20002	118125	illBliss
1189	Irina Barros	<a href="https://www.last.fm/music/Irina+Barr...	4813	44915	Rui Orlando
1190	PAAX (Tulum)	<a href="https://www.last.fm/music/PAAX+(Tulu...	4002	449	Antaares
1191	Matt Sawyer	<a href="https://www.last.fm/music/Matt+Sawye...	1763	4411	Stereo Kulisse
1192	Cubita	Re...	5400	50332	Nuno Ribeiro
1193	Amaarae	Ama Serwah Genfi (born July 4, 1994), known pr...	938668	30555467	Rochelle Jordan



	artist_name	biography	listeners	playcount	similar_artist
1188	Praiz	Praise Ugbede Adejo (born 8 March 1984), bette...	20002	118125	illBliss
1189	Irina Barros	https://www.last.fm/music/Irina+Barros	4813	44915	Rui Orlando
1190	PAAX (Tulum)	https://www.last.fm/music/PAAX+(Tulum)	4002	449	Antaares
1191	Matt Sawyer	https://www.last.fm/music/Matt+Sawyer	1763	4411	Stereo Kulisse
1192	Cubita	https://www.last.fm/music/Cubita	5400	50332	Nuno Ribeiro
1193	Amaarae	Ama Serwah Genfi (born July 4, 1994), known pr...	938668	30555467	Rochelle Jordan



Función **clean_html_links**:



```
def clean_html_links(text):  
    if not isinstance(text, str):  
        return text  
  
    # Reemplaza <a href="URL">texto</a> → URL.  
    text = re.sub(r'<a href="(https?://[^"]+)"\.?>\.?</a>', r'\1', text)  
  
    # Quita cualquier otra etiqueta HTML que quede.  
    text = re.sub(r'<.*?>', '', text)  
  
    return text.strip()
```



Mediante la función **clean_html_links**, se crea un nuevo DataFrame con la bio más limpia:

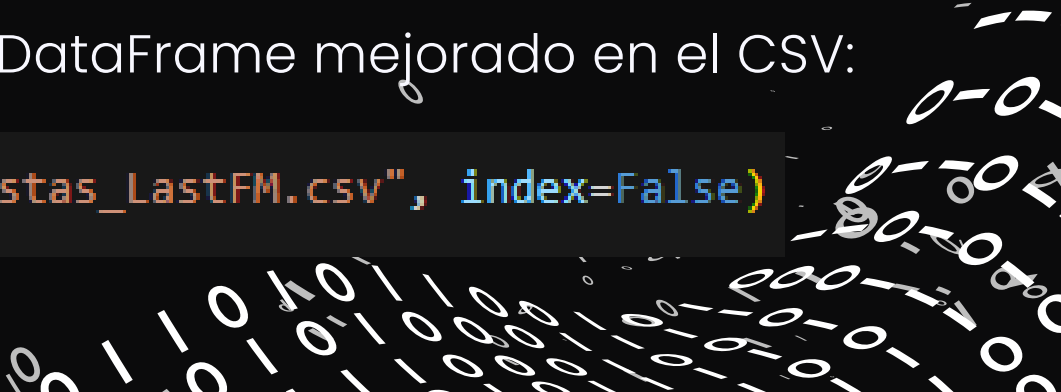
```
df_artists_lastfm = pd.read_csv("Artistas_LastFM.csv")  
  
for i, artist in enumerate(df_artists_lastfm["artist_name"], start=1):  
    biography_text = df_artists_lastfm.loc[i-1, "biography"]  
    cleaned_text = clean_html_links(biography_text) # Aquí se limpia el código HTML.  
    df_artists_lastfm.loc[i-1, "biography"] = cleaned_text # Se guarda el texto limpio en la misma celda.  
    print(f"{i}. ✅ {artist}")  
    time.sleep(0.25)
```



FINALMENTE se sobrescribe el DataFrame mejorado en el CSV:



```
df_artists_lastfm.to_csv("Artistas_LastFM.csv", index=False)
```



BASE DE DATOS EN MYSQL

Creación de la base de datos **musicstream**

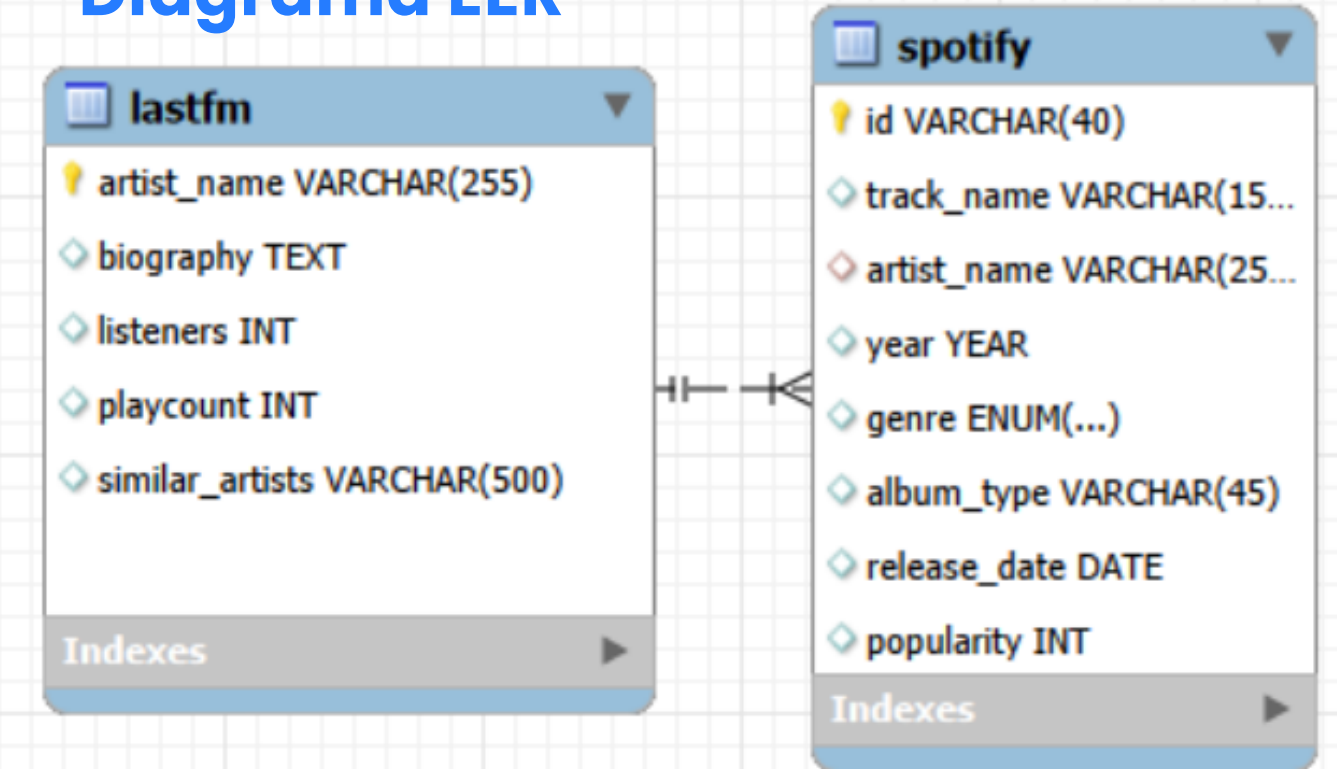


1. Crear el schema
2. Crear las tablas
3. Definir los tipos de datos
4. Visualizar el diagrama EER
5. Insertar datos a través de Python

```
1 • DROP SCHEMA IF EXISTS musicstream;
2
3 • CREATE SCHEMA musicstream;
4
5 • USE musicstream;
6
7 -- TABLE 1
8 • CREATE TABLE lastfm (
9     artist_name VARCHAR(255) PRIMARY KEY,
10    biography TEXT,
11    listeners INT,
12    playcount INT,
13    similar_artists VARCHAR(500)
14 );
```

```
17 -- TABLE 2
18 • CREATE TABLE spotify (
19     id VARCHAR(40) PRIMARY KEY,
20     track_name VARCHAR(150),
21     artist_name VARCHAR(255),
22     `year` YEAR,
23     genre ENUM('soundtrack', 'k-pop', 'afro', 'electronic'),
24     album_type VARCHAR(45),
25     release_date DATE,
26     popularity INT,
27     CONSTRAINT fk_spotify_lastfm
28         FOREIGN KEY (artist_name)
29         REFERENCES lastfm(artist_name)
30         ON DELETE CASCADE
31         ON UPDATE CASCADE
32 );
```

Diagrama EER



-- nuestra FK para lastfm
-- backticks para que se reconociera como identificador (comillas no)

BASE DE DATOS EN MYSQL

Inserción de los datos desde python

```
import mysql.connector
from mysql.connector import errorcode
import pandas as pd

# conectar con mysql -- cambiar PASSWORD! ;)
cnx = mysql.connector.connect(
    user='root',
    password='PASSWORD',
    host='127.0.0.1')

cursor = cnx.cursor()
cursor.execute("USE musicstream")
cnx.database

# importar .csv para tener un DataFrame
df_lastfm = pd.read_csv('Artistas_LastFM.csv')
df_spotify = pd.read_csv('Canciones_Spotify.csv')

df_spotify.head(73)
df_lastfm.tail(10)

# MySQL - insertar data en la tabla LastFM
sql_lastfm = """INSERT INTO lastfm (artist_name, biography,
    listeners, playcount, similar_artists)
    VALUES (%s, %s, %s, %s, %s)
    """

# MySQL - insertar data en la tabla Spotify (Unicos)
sql_spotify = """INSERT INTO spotify (id, track_name, artist_name,
    `year`, genre, album_type, release_date, popularity)
    VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
    """

cursor.executemany(sql_lastfm, df_lastfm.values.tolist())
cnx.close()
```



RETOS ENCONTRADOS SQL

- ⚠ **VALORES NULOS:** *datos con valores Nulos.*
- ✅ **SOLUCIÓN:** *convertir Nulos de NaN a None, para que SQL acepte los datos.*
- ⚠ **DUPLICADOS SPOTIFY:** *cargar los datos de Spotify en SQL- 61 IDs repetidos.*
- ✅ **SOLUCIÓN:** *código para que al insertar los datos, si encuentra un ID repetido no lo incluya y nos devuelva un mensaje de error con el ID repetido.*
- ⚠ **FECHAS INCOMPLETAS:** *datos sin mes y dia.*
- ✅ **SOLUCIÓN:** *convertir Nulos de NaN a None, para que SQL acepte los datos.*

*TOP ARTISTAS TOP CANCIONES
ES TOP ARTISTAS TOP CANCI
ONES TOP ARTISTAS TOP CAN
CIONES TOP ARTISTAS TOP C*

2015 - 2019

ELECTRONIC AFRO SOUNDTRACK K-POP

TOP 5 ARTISTAS
MÁS REPRODUCIDOS
DOS POR CADA
GENERO

ELECTRONIC

DRAKE
THE WEEKND
CHARLI XCX
GORILLAZ
FUTURE

AFRO

BEYONCÉ
METRO BOOMIN
CHRIS BROWN
MAJOR LAZER
JORJA SMITH

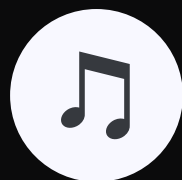
SOUNDTRACK

HANS ZIMMER
TOBY FOX
KELLY CLARKSON
LISA
JOHN WILLIAMS

K-POP

SUZY
DRAIN
HALO
J
SPACE COWBOY

TOP 5 CANCIONES POR GENERO



ELECTRONICA

WAITING FOR LOVE

2015-10-02

TOOK A PILL IN IBIZA

- SEEB REMIX

2016-05-06

**SUGAR (FEAT.
FRANCESCO YATES)**

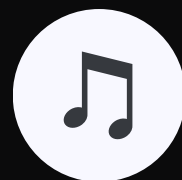
2015-09-25

ARE YOU WITH ME

2016-10-21

**MORE THAN YOU
KNOW**

2017-07-28



AFRO

**LOVE NWANTITI (AH
AH AH)**

2019-08-30

SHAKE BODY

2015-05-18

YE

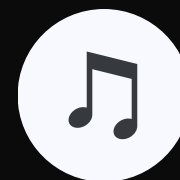
2018-01-26

MAD OVER YOU

2016-11-04

DUMEBI

2019-03-22



SOUNDTRACK

**TUJHE KITNA
CHAHNE LAGE (FROM
"KABIR SINGH")**

2019-05-31

REWRITE THE STARS

2017-11-17

WAIT FOR IT

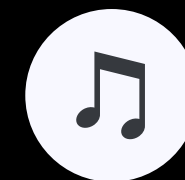
2015-09-25

A MILLION DREAMS

2017-12-08

**ALEXANDER
HAMILTON**

2015-09-25



K-POP

**EDEMBE
LY'OMUKYALA**

2015-11-12

**TYAR BHE KE AARE
REES**

2017-01-20

BE MY SPRING

2019-02-26

PUT U RAJ

2018-04-12

YEH MUJHE KYA HUA

2015-01-02

+ CANCIONES LANZADAS
+ CANCIONES LANZADAS
+ CANCIONES LANZADAS
+ CANCIONES LANZADAS



TOBY
FOX

57
SOUNDTRACK



BURNA
BOY

34
AFRO



AVICII

26
ELECTRONIC



YOONWON

5
K-POP

**EL GENERO
MUSICAL MÁS
REPRODUCIDO**

2015 – 2019



ELECTRONICA

ELECTRONICA

ELECTRONICA

RONICA



SIMILARES + RECOMENDADOS
SIMILARES + RECOMENDADOS
SIMILARES + RECOMENDADOS
SIMILARES + RECOMENDADOS



LENA RAINE

57

SOUNDTRACK



WIZKID

38

AFRO



KYGO

26

ELECTRONIC



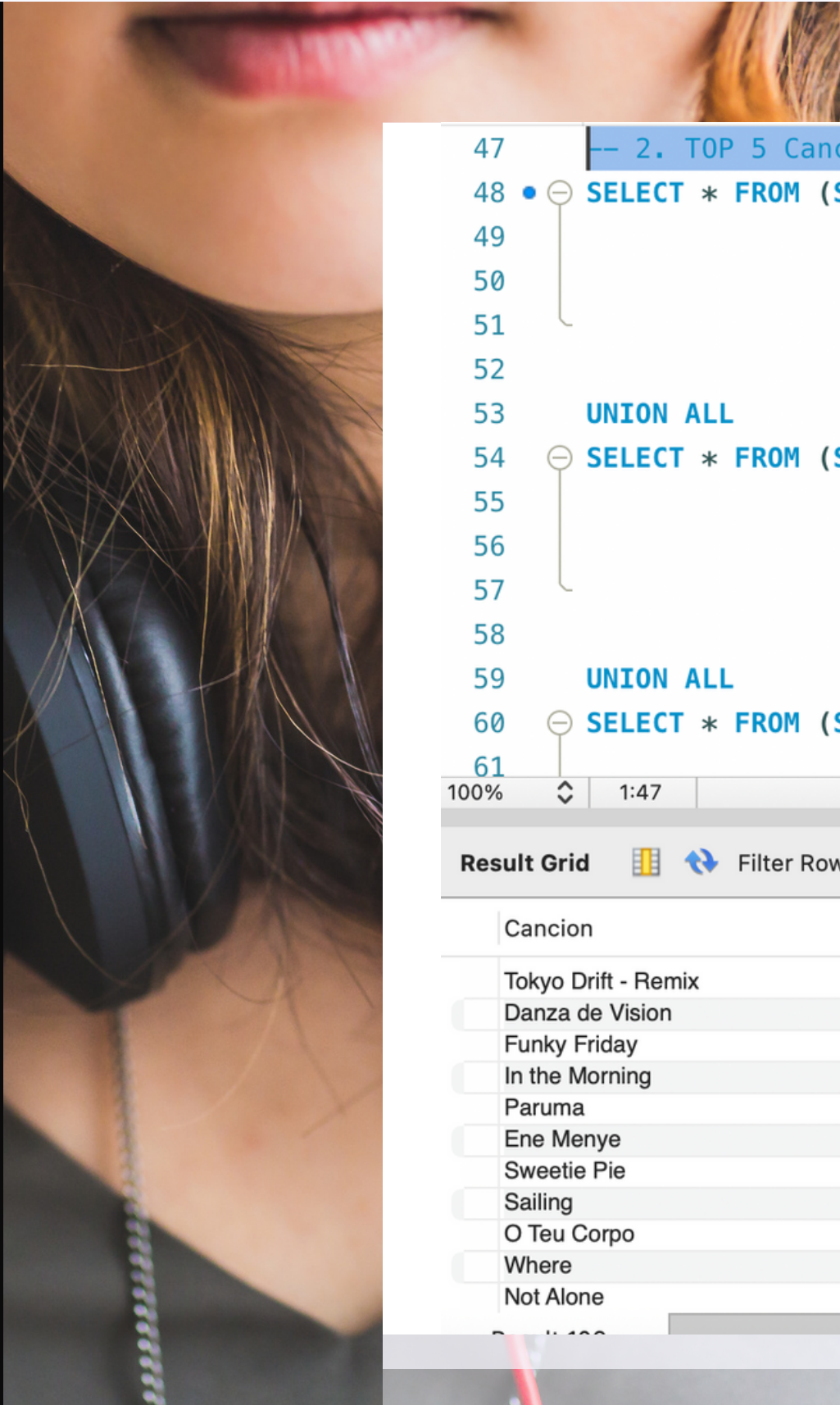
JUNG
CHANG
YONG

5

K-POP

EJEMPLO DE CONSULTA

TOP 5 Canciones más populares por genero y su fecha de lanzamiento



47 -- 2. TOP 5 Canciones más populares por genero y su fecha de lanzamiento (Spotify)

48 • SELECT * FROM (SELECT track_name AS Cancion, release_date AS Lanzamiento, genre AS Genero

49 FROM spotify

50 WHERE genre = 'electronic'

51 LIMIT 5) AS electronic

52

53 UNION ALL

54 • SELECT * FROM (SELECT track_name AS Cancion, release_date AS Lanzamiento, genre AS Genero

55 FROM spotify

56 WHERE genre = 'afro'

57 LIMIT 5) AS afro

58

59 UNION ALL

60 • SELECT * FROM (SELECT track_name AS Cancion, release_date AS Lanzamiento, genre AS Genero

61 FROM spotify

100% 1:47

Result Grid

Filter Rows: Search

Export:

Cancion	Lanzamiento	Genero
Tokyo Drift - Remix	2019-03-07	electronic
Danza de Vision	2017-01-23	electronic
Funky Friday	2018-10-04	electronic
In the Morning	2016-07-29	electronic
Paruma	2016-08-16	electronic
Ene Menye	2015-04-24	afro
Sweetie Pie	2017-11-22	afro
Sailing	2016-06-19	afro
O Teu Corpo	2015-07-20	afro
Where	2016-02-25	afro
Not Alone	2017-04-01	soundtrack

MUSICSTREAM MUSICSTREAM MUSICSTREAM MUSICSTREAM MUSICSTREAM

“

Próximos pasos

- AÑADIR MÁS FUENTES DE DATOS (EJ. MUSICBRAINZ).
- CREAR UN SISTEMA DE RECOMENDACIONES BASADO EN SQL.
- FORMALIZAR LA EXTRACCIÓN DE DATOS COMO UN SERVICIO DE API PARA CLIENTES.

MUSICSTREAM MUSICSTREAM MUSICSTREAM MUSICSTREAM MUSICSTREAM

MUSIC STREAM MUSIC STREAM
MUSIC STREAM MUSIC STREAM
MUSIC STREAM
MUSIC STREAM

CONTACT FORM

CLICK BELOW



EMAIL ADDRESS

hello@musicstream.com

PHONE NUMBER

+34 123456789

PLATAFORMA DEMO

<https://musicstream.com/>

Contact Us

GRACIAS

GRACIAS

GRACIAS

GRACIAS

GRACIAS POR VUESTRO TIEMPO