

DOCUMENTACIÓN DE PROYECTO INTEGRADOR DE APRENDIZAJE

Programación básica

Extracción De Datos, Limpieza De Datos Y Estructura De Datos

Perla Cristal López Álvarez - 2225483 | Raquel Guadalupe Gloria Vázquez - 2011258

Grupo: 073

La API seleccionada fue: **Spotify**

Definición del Problema

Análisis de Coachella 2025, centrado en los artistas, así como sus seguidores, géneros musicales y nivel de popularidad de cada uno. Haciendo una comparación entre cada artista por día y las estadísticas generales de cada día. A continuación, se muestra el póster de Coachella 2025 que servirá de base para recolectar los artistas que se presentaron.



Extracción de Datos

En la documentación de la primera entrega se explica los requisitos que se necesita para poder realizar una conexión exitosa a la API de Spotify, De tal manera que podamos obtener los datos que necesitamos para resolver el problema planteado.

Así que a continuación se explicará de manera breve todas las acciones necesarias (líneas de código) para poder hacer la extracción de datos.

1. Se definió una función llama a datos `_artista`, la cual recibe como parámetros Coachella e `i`, donde Coachella es una lista de listas, cada una de las listas que conforman la lista Coachella son los IDs de los artistas que se presentaron en cada día del festival. Mientras que `i` se toma como un contador.

```
def datos_artistas (coachella, i):
```

2. Dentro de la función, se define un ciclo for anidado, el primero es para cada día en Coachella donde recorrerá la lista Coachella, y el segundo es para cada artista en día, en donde recorrerá cada lista de días que se encuentra en la lista Coachella.

```
    for dia in coachella:
        for artista in dia:
```

Dentro del segundo for, se realizarán las siguientes acciones:

3. Se declara una variable llamada `url_artista`, la cual es la liga o link a la que se va a consultar los datos que se necesita, que este caso son los artistas que se presentaron en cada día del festival.

```
url_artista = f"https://api.spotify.com/v1/artists/{artista}"
```

4. Se declara un bloque try para verificar que, en el siguiente bloque de código, al momento de ejecutarlo no ocurra algún error, sobre todo en la conexión de la API.

```
    try:
        datos = reque
```

5. Dentro de bloque try, se declara una variable llamada `datos` con la cual haremos una petición tipo get para así poder realizar la conexión al api de Spotify con la URL proporcionada anteriormente. Además de que con `headers=acceso`: se envía información de autenticación (por ejemplo, el token de acceso a la API).

```
        datos = requests.get(url_artista,headers = acceso)
```

ES IMPORTANTE QUE PARA REALIZAR ESTA ACCION ESTE IMPORTADO EL MODULO REQUESTS

6. Después realizamos una pequeña validación de estatus, para conocer si se hizo la conexión correctamente y de no ser así saber el tipo de estatus que tiene.

```
        datos.raise_for_status()
```

7. Si la solicitud fue exitosa, se declara una variable llamada info en convierte la respuesta (que está en formato JSON) a un diccionario de Python con la información del artista.

```
#Extracción de datos
info = datos.json()
```

Verificación de Datos

Después del último paso de la extracción de datos es necesario verificarlos, siempre dependiendo del contexto del problema. En este caso verificaremos que los datos sean ciertos.

La siguiente línea de código es un condicional, el cual verifica que si el campo genres (géneros musicales) del artista este vacío, en caso de que lo esté se va a asignar el valor desconocido para evitar que esté en blanco.

```
if len(info["genres"]) == 0:
    info["genres"] = "desconocido"
```

Ya validado fuera del condicional, se utilizarán las expresiones regulares para buscar una o más letras, números o guiones bajos por lo que se crea un patrón el cual debemos compilar.

```
patron_id = re.compile(r'\w+')

```

ES IMPORTANTE QUE PARA REALIZAR ESTA ACCION ESTE IMPORTADO EL MODULO RE

Se define mo1 el cual se aplica la expresión regular al campo id del artista E intentará hacer coincidir el patrón que ya se definió desde el inicio del string.

```
mo1 = re.match(patron_id, info["id"])
```

Pero en caso de que no encuentre coincidencias, el siguiente condicional mandará un mensaje de error con el nombre del artista.

```
if not mo1:
    print(f"Error con el artista: {info["name"]} en su id")

```

Limpieza de Datos

Para la parte de limpieza de datos se necesita organizar y estructurar la información que nos proporciona la API, qué son los artistas.

La respuesta original de la API de Spotify cuenta con muchos datos, pero para fines del problema necesitamos los siguientes:

- Nombre
- ID
- Géneros
- Seguidores
- Popularidad

Así que para ello se creó una estructura de tipo diccionario, el cual lleva por nombre artista.

```
artista = {  
    "nombre": info["name"],  
    "id": info["id"],  
    "generos": info["genres"],  
    "seguidores": info["followers"]["total"],  
    "popularidad": info["popularity"]}
```

Diseño de la Nueva Estructura de Datos

Para poder almacenar los datos ya depurados con la información que necesitamos, se necesita una lista de diccionarios, para ellos se van a crear cuatro listas, las primeras 3 listas serán de los artistas que se presentaron en cada día del festival, y la cuarta lista será una lista de listas, donde las listas que serán guardadas en esa lista serán las primeras 3.

```
datos_dia1 = []
datos_dia2 = []
datos_dia3 = []
datos_coachella = [datos_dia1, datos_dia2, datos_dia3]
```

Se declara una variable llamada información en la cual se va a acceder el elemento i de la lista de datos_coachella, Y se van a guardar los datos del artista actual.

```
informacion = datos_coachella[i]
```

Después se va a agregar el diccionario artista a la lista correspondiente del día i.

```
informacion.append(artista)
```

Tenemos que tomar en cuenta que esta acción y todas las anteriores están dentro del bloque try, en el cual cuenta con una excepción qué es el error de conexión con la API de Spotify

```
except requests.exceptions.ConnectionError:
    print("Error : no se pudo conectar a la API. verificar conexion y token")
+= 1
```

Después tenemos el incrementable del contador i

```
i += 1
```

Y por último el retorno de los datos, recordemos que todas las acciones previamente mencionadas están dentro de la función declarada datos_artista()

```
return informacion
```

Almacenamiento de Datos

Para el almacenamiento de datos se decidió usar los archivos con extensión .txt

Definimos una variable llamada fo en donde abrimos un archivo con el nombre PIA_ E2_ ARCHIVO.txt, En caso de que no exista con el comando “w” va a crearlo

```
fo = open("PIA_E2_ARCHIVO.txt", "w")
```

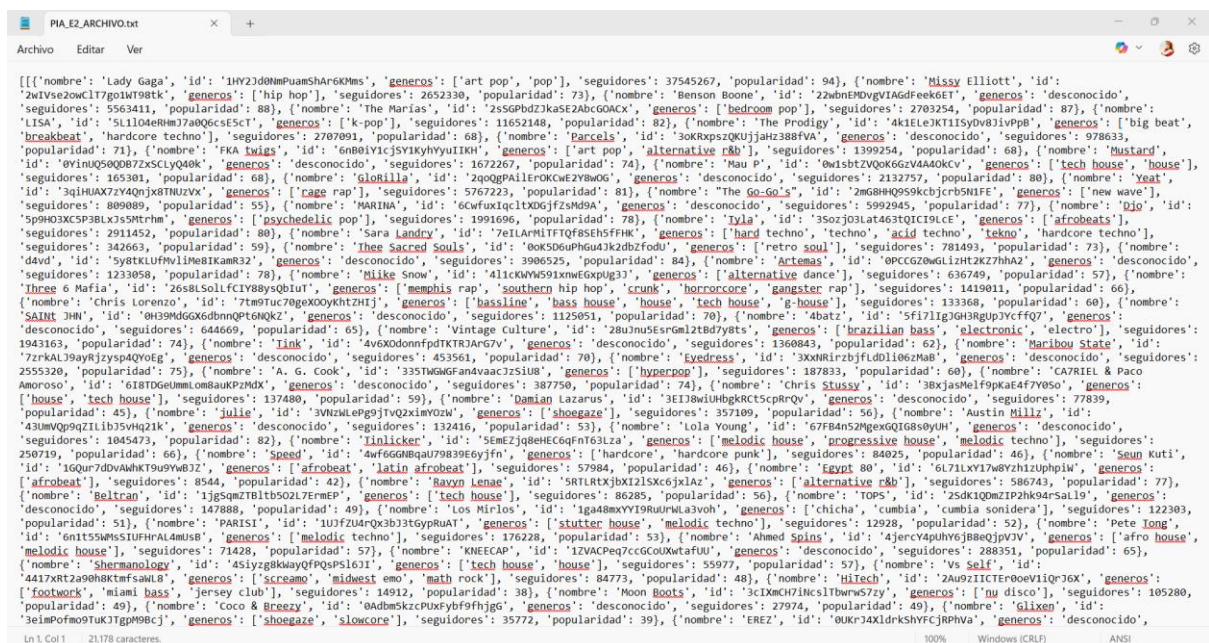
y después con la función write vamos a convertir los datos a un tipo string para que se puedan escribir en el archivo .txt, los datos que se van a escribir será la lista datos_coachella.

```
fo.write(str(datos_coachella))
```

Cerramos el archivo para que se pueda guardar los datos

```
fo.close()
```

Abrimos el archivo .txt y podemos ver que si se guardan los datos de la lista.



```
[{"nombre": "Lady Gaga", "id": "1HY23d0MmPuaMShAr6KMs", "generos": ["art pop", "pop"], "seguidores": 37545267, "popularidad": 94}, {"nombre": "Missy Elliott", "id": "2w1VseZowcl7g0w1W9t8t", "generos": ["hip hop"], "seguidores": 2652330, "popularidad": 73}, {"nombre": "Benson Boone", "id": "22wbNEMDvGVIAAdFeeK6ET", "generos": "desconocido", "seguidores": 5563411, "popularidad": 88}, {"nombre": "The Marias", "id": "2S56PbDZ3kaSE2AbcGOACK", "generos": ["bedroom pop"], "seguidores": 2703254, "popularidad": 87}, {"nombre": "LISA", "id": "5L1104eRhmJ7aBQ6csE5cT", "generos": ["k-pop"], "seguidores": 11652148, "popularidad": 82}, {"nombre": "The Prodigy", "id": "4kiELeJKT1IsyDv83lvPpB", "generos": ["big beat", "breakbeat", "hardcore techno"], "seguidores": 2707091, "popularidad": 68}, {"nombre": "Parcels", "id": "3oKRxpszQKUjjahZ388fVA", "generos": "desconocido", "seguidores": 978633, "popularidad": 71}, {"nombre": "FKA twigs", "id": "6nB01V1cJSY1kyhyvuiIKH", "generos": ["art pop", "alternative r&b"], "seguidores": 1399254, "popularidad": 68}, {"nombre": "Mustard", "id": "0YinUQ50Q087ZxSclyQ40k", "generos": "desconocido", "seguidores": 1672267, "popularidad": 74}, {"nombre": "Mau P", "id": "0wisbtZVQ0K6GzVAA40Kcv", "generos": ["tech house", "house"], "seguidores": 165381, "popularidad": 68}, {"nombre": "GloRilla", "id": "2uQ0gPAlIerOCkEzV8u0G", "generos": "desconocido", "seguidores": 2132757, "popularidad": 80}, {"nombre": "Yeat", "id": "3qIHUAX7ZY4QjX8TU2Vx", "generos": ["cage rap"], "seguidores": 5767223, "popularidad": 81}, {"nombre": "The Go-Go's", "id": "2mG8w4Q959KcbJcRb5HfE", "generos": ["new wave"], "seguidores": 809089, "popularidad": 55}, {"nombre": "MARINA", "id": "6Cwfux1qclTX0GjZsM4G", "generos": "desconocido", "seguidores": 5992945, "popularidad": 77}, {"nombre": "Dio", "id": "5p9H03XC5P3BLX35Mtrhm", "generos": ["psychedelic pop"], "seguidores": 1991696, "popularidad": 78}, {"nombre": "Tyla", "id": "35ozj03Lat463tQIC19LcE", "generos": ["afrobeats"], "seguidores": 2911452, "popularidad": 80}, {"nombre": "Sara Landry", "id": "7e1LArM1TFTQF8SEH5FFHK", "generos": ["hard techno", "techno", "acid techno", "tekno", "hardcore techno"], "seguidores": 342663, "popularidad": 59}, {"nombre": "Thee Sacred Souls", "id": "0okSD6uPhu4Jk2dbZfodu", "generos": ["retro soul"], "seguidores": 781493, "popularidad": 73}, {"nombre": "david", "id": "5y8TKLUHvMv1Me8IKamR32", "generos": "desconocido", "seguidores": 3906525, "popularidad": 84}, {"nombre": "Artemas", "id": "0pCCGZ0wGL1ZHT2KZ7hhA2", "generos": "desconocido", "seguidores": 1233058, "popularidad": 78}, {"nombre": "Mike Snow", "id": "41icKwM591xwF0pUg3", "generos": ["alternative dance"], "seguidores": 636749, "popularidad": 57}, {"nombre": "Three 6 Mafia", "id": "2es8is0llFCfY8yS0btu", "generos": ["memphis rap", "southern hip hop", "crunk", "horrorcore", "gangster rap"], "seguidores": 1419011, "popularidad": 66}, {"nombre": "Chris Lorenzo", "id": "72m0u7G0eX00yKtZJH", "generos": ["bassline", "bass house", "house", "tech house", "g-house"], "seguidores": 133368, "popularidad": 60}, {"nombre": "SAINT JHN", "id": "0H39MdG6d0nnQPt6WQKZ", "generos": "desconocido", "seguidores": 1125051, "popularidad": 70}, {"nombre": "4batz", "id": "5fi7lIgJGH3RgUpYvcF07", "generos": "desconocido", "seguidores": 644669, "popularidad": 65}, {"nombre": "Vintage Culture", "id": "28u3nu5EsrgM1zTBD7y8ts", "generos": ["brazilian bass", "electronic", "electro"], "seguidores": 1943163, "popularidad": 74}, {"nombre": "Tink", "id": "4v6X0donnfpdTKTRJArG7v", "generos": "desconocido", "seguidores": 1360843, "popularidad": 62}, {"nombre": "Maribou State", "id": "72rkAL39ayRjzysp4QYOtg", "generos": "desconocido", "seguidores": 453561, "popularidad": 70}, {"nombre": "Eyedress", "id": "3XXNRirzbJfLD1l06zMaB", "generos": "desconocido", "seguidores": 255520, "popularidad": 75}, {"nombre": "A. G. Cook", "id": "335TMAwGFah4VaacJ25lU8", "generos": ["hyperpop"], "seguidores": 187833, "popularidad": 60}, {"nombre": "CA/RIEL & Paco Amoroso", "id": "618T0geumL0m0tauxP2MdK", "generos": "desconocido", "seguidores": 387750, "popularidad": 74}, {"nombre": "Chris Stussy", "id": "3Bxja9Melf9pkAe47V9so", "generos": ["house", "tech house"], "seguidores": 137480, "popularidad": 59}, {"nombre": "Damian Lazarus", "id": "3E1I8wLiUhgkRcT5cprQv", "generos": "desconocido", "seguidores": 77839, "popularidad": 45}, {"nombre": "Julie", "id": "3VnZMLeg9jTVQ2ximv0Zu", "generos": ["shoegaze"], "seguidores": 357109, "popularidad": 56}, {"nombre": "Austin Millz", "id": "43uWQp9qZILib75vH2ik", "generos": "desconocido", "seguidores": 132416, "popularidad": 53}, {"nombre": "Lola Young", "id": "67FB4n52MxexGQT688oyUH", "generos": "desconocido", "seguidores": 1045473, "popularidad": 82}, {"nombre": "Iinlicker", "id": "5EmEzjg8HEC6qfnt63Lza", "generos": ["melodic house", "progressive house", "melodic techno"], "seguidores": 250719, "popularidad": 66}, {"nombre": "Speed", "id": "4wf6G0NBqau79839E6yJfn", "generos": ["hardcore", "hardcore punk"], "seguidores": 84025, "popularidad": 46}, {"nombre": "Seun Kutii", "id": "1G0u7D0vAWHKT9u9vWb32", "generos": ["afrobeat", "latin afrobeat"], "seguidores": 57984, "popularidad": 46}, {"nombre": "Egypt 80", "id": "6L71xYL17w8Yzh1zUpHplW", "generos": ["afrobeat"], "seguidores": 8544, "popularidad": 42}, {"nombre": "Rayn Lenae", "id": "58TlRtXjBjZ1SK6jvIaZ", "generos": ["alternative r&b"], "seguidores": 586743, "popularidad": 77}, {"nombre": "Beltran", "id": "1jG5qmZTLbt050ZL7rMfEP", "generos": ["tech house"], "seguidores": 86285, "popularidad": 56}, {"nombre": "TOPS", "id": "25dkIQ0mZP2zhK04r5aLl9", "generos": "desconocido", "seguidores": 147888, "popularidad": 49}, {"nombre": "Los Mirlos", "id": "1ga48mXVY19UuWMLa3voh", "generos": ["chicha", "cumbia", "cumbia sonidera"], "seguidores": 122303, "popularidad": 51}, {"nombre": "PARISI", "id": "1U3fZU4RqXb33t6yprUAT", "generos": ["stutter house", "melodic techno"], "seguidores": 12928, "popularidad": 52}, {"nombre": "Pete Tong", "id": "6nit55Mm6SIUHFHAlAMuSB", "generos": ["melodic techno"], "seguidores": 176228, "popularidad": 53}, {"nombre": "Ahmed Spins", "id": "4jercv4puhvej8B8eqjPjVJ", "generos": ["afro house", "melodic house"], "seguidores": 71428, "popularidad": 57}, {"nombre": "KNEECAP", "id": "1ZVACPeq7ccGCOUxtafUU", "generos": "desconocido", "seguidores": 288351, "popularidad": 65}, {"nombre": "Shermanology", "id": "45iyzg8kwayQFPqs16J1", "generos": ["tech house", "house"], "seguidores": 55977, "popularidad": 57}, {"nombre": "Vs Self", "id": "44170RT209hKtuf5au18", "generos": ["screamo", "midwest emo", "math rock"], "seguidores": 84773, "popularidad": 48}, {"nombre": "idlesb", "id": "2Au9ZiCTE00u0vI0g76X", "generos": ["footwork", "miami bass", "jersey club"], "seguidores": 14912, "popularidad": 38}, {"nombre": "Moon Boots", "id": "3cIXmCH1HcslTBw857zy", "generos": ["nu disco"], "seguidores": 105280, "popularidad": 49}, {"nombre": "Coco & Breazy", "id": "0adbmSkzCuxYvbf9fhjgg", "generos": "desconocido", "seguidores": 27974, "popularidad": 49}, {"nombre": "Glixen", "id": "3e1mPofm0adUKJ7gpm98Cj", "generos": ["shoegaze", "slowcore"], "seguidores": 35772, "popularidad": 39}, {"nombre": "EREZ", "id": "0UKr34XldrksHYFCJRPfHvA", "generos": "desconocido", "seguidores": 165381, "popularidad": 68}
```