

DOCUMENTACIÓN DE PROYECTO INTEGRADOR DE APRENDIZAJE

Programación básica

Análisis Estadístico y Lectura 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.



Lectura de Datos

Para esta entrega se utilizarán los datos obtenidos de la API de Spotify de la primera y segunda entrega, que se encuentran guardados en un archivo de texto.

Para poder usar los datos que están en el archivo de texto, primero se tienen que reacomodar los datos, para ello se crean dos diccionarios, el primero para poder representar a un artista y el segundo para poder dividir el evento.

```
#Reorganización de datos
artista = {"nombre" : "",
          "id" : "",
          "genero": [],
          "seguidores" : 0,
          "popularidad": 0}

evento = {"dia 1" : "NA",
          "dia 2" : "NA",
          "dia 3" : "NA"}
|
```

Después se declaran listas vacías para poder almacenar los datos que obtenemos del archivo de texto, de tal forma que podamos manipular la información (datos) y realizar cálculos estadísticos más adelante

```
|
lista_dia1 = []
lista_dia2 = []
lista_dia3 = []
lista_generos = []
lista_seguidores = []
lista_popularidad = []
lista_dia1_generos = []
lista_dia2_generos = []
lista_dia3_generos = []
lista_dia1_seguidores = []
lista_dia2_seguidores = []
lista_dia3_seguidores = []
lista_dia1_popularidad = []
lista_dia2_popularidad = []
lista_dia3_popularidad = []
```

Se declaran i, a, y b como contadores y se inicializan en 1,1 0 respectivamente

```
i=1
a=1
b=0
```

Abrimos el archivo en modo lectura

```
#Lectura de datos
with open ("PIA_E2_ARCHIVO.txt","r") as file:
```

Con un ciclo for separamos el contenido por días, usando “ }] ” que nos ayuda como delimitador para identificar el final de datos por un día

```
for line in file:
    info = line.split("} ]")
    evento["dia 1"] = info[0]
    evento["dia 2"] = info[1]
    evento["dia 3"] = info[2]
    . . . . .
```

Validación de Datos

dentro del mismo for, se utiliza otro for el cual recorre los 3 días (cantidad de día que dura el festival).

```
#Validación de datos
for dia in range(3):
```

Dentro de este mismo for, se declara un ciclo while true, el cual siempre va a entrar, si y solo si el argumento o condición a validar sea verdadera. Ya dentro de while los datos los separa con “ : “.

```
while True:
    infor = evento[f"dia {a}"].split(":")
```

después dentro del mismo while se definen varios patrones de expresión regular para poder extraer el nombre, ID, genero, seguidores y popularidad

```
infor = evento[f"dia {a}"].split(":")
patron_nombre = re.compile(r"\\"?\\?\\'?( [a-zA-Z0-9É\\s]+)\\"?\\'\\?\\?\\, (\\'id\\')")
mo1 = patron_nombre.search(infor[i])
artista["nombre"] = mo1.group(1)
patron_id = re.compile(r"\\"'(\\w+)\\'", (\\'generos\\'))
mo2 = patron_id.search(infor[i+1])
artista["id"] = mo2.group(1)
patron_generos = re.compile(r"(\\[?\\D+\\]?\\, (\\'seguidores\\'))")
mo3 = patron_generos.search(infor[i+2])
artista["genero"] = mo3.group(1)
patron_seguidores = re.compile(r"\\(d+)\\, (\\'popularidad\\)")
mo4 = patron_seguidores.search(infor[i+3])
artista["seguidores"] = mo4.group(1)
patron_popularidad = re.compile(r"\\(d+)\\}\\}\\}\\?\\?\\, ? \\{?(\\'nombre\\')?")
mo5 = patron_popularidad.search(infor[i+4])
```

Se realizan unas validaciones ´para poder encontrar la popularidad, en caso de que no este, se asigna el valor fijo de 93, si no está vacía se agrupa.

```
if mo5 == None:
    artista["popularidad"] = 93
else:
    artista["popularidad"] = mo5.group(1)
```

Se hace una copia de artista en la lista correspondiente al día.

```
datos = artista.copy()
```

y se agregan a la lista correspondiente al día.

```

if dia==0:
    lista_dia1.append(datos)
elif dia==1:
    lista_dia2.append(datos)
else:
    lista_dia3.append(datos)
i += 5

```

El ciclo while se termina cuando se encuentra 'Travis Scott'

```

if mol.group(1) == 'Travis Scott':
    a += 1
    i=1
    break

```

Preparación para Visualización

Se declara una lista de listas donde cada elemento de la lista principal es una lista para cada día del evento.

```
lista_evento = [lista_dia1, lista_dia2, lista_dia3]
```

Con un ciclo for anidado, recorremos la lista de listas, el primer for es para los días, y el segundo es por cada día se recorre la lista de artistas.

```
for dia in evento:
    for artista in lista_evento[b]:
```

después separamos los géneros

```
gen = artista["genero"].split(",")
```

se declara otro for dentro del segundo por anidado, el cual no soy ayuda a recorrer cada género.

```
for elemento in gen:
```

Con ayuda de las expresiones regulares se limpia el nombre del género y se guarda en la lista del día correspondiente, junto con los seguidores y la popularidad.

```
patron_gen = re.compile(r"\"[?(\\"|' ) ([a-zA-Zá-úñ\s\\-\\&]+) (\\"|' )\\]?")
mo6 = patron_gen.search(elemento)
if b==0:
    lista_dia1_generos.append(mo6.group(2))
elif b==1:
    lista_dia2_generos.append(mo6.group(2))
else:
    lista_dia3_generos.append(mo6.group(2))
if b==0:
    lista_dia1_seguidores.append(artista["seguidores"])
    lista_dia1_popularidad.append(artista["popularidad"])
elif b==1:
    lista_dia2_seguidores.append(artista["seguidores"])
    lista_dia2_popularidad.append(artista["popularidad"])
else:
    lista_dia3_seguidores.append(artista["seguidores"])
    lista_dia3_popularidad.append(artista["popularidad"])
b +=1
```

Análisis Estadístico

En este script se realizaron 2 cálculos, el primero fue la suma de los seguidores por cada día del festival, y la suma total de seguidores durante el festival.

```
#calculos estadisticos
#SUMA SEGUIDORES
lista_dia1_seguidores = [int(c) for c in lista_dia1_seguidores]
lista_dia2_seguidores = [int(c) for c in lista_dia2_seguidores]
lista_dia3_seguidores = [int(c) for c in lista_dia3_seguidores]

suma_dia1 = sum(lista_dia1_seguidores)
suma_dia2 = sum(lista_dia2_seguidores)
suma_dia3 = sum(lista_dia3_seguidores)
suma_coachella = suma_dia1 + suma_dia2 + suma_dia3
print("la suma de seguidores del dia 1 es: ", suma_dia1)
print("la suma de seguidores del dia 2 es: ", suma_dia2)
print("la suma de seguidores del dia 3 es: ", suma_dia3)
print("la suma total de seguidores:", suma_coachella) #con este
```

Los datos que se encuentran en las listas de seguidores por día tuvieron que ser convertidos a datos enteros para poder sumarlos.

Estos datos los interpretamos como la cantidad de asistentes que hubo en cada día del festival, y la cantidad total de asistentes durante todo el fin de semana que duró el festival.

El segundo cálculo fue obtener la moda o la frecuencia de los géneros que se presentaron en cada día del festival, es por eso por lo que se contaron la cantidad de géneros que hay en cada lista de días.

```
frec_dia1 = Counter(lista_dia1_generos)
frec_dia2 = Counter(lista_dia2_generos)
frec_dia3 = Counter(lista_dia3_generos)
```

Por lo que después se declaró una lista de llamada géneros, después se declaran 2 for, el primero para recorrer la lista principal, y el segundo para recorrer cada lista que se encuentra dentro de la lista principal, de tal forma que obtenemos el género y se va agregando a la lista de géneros previamente ya declarada.

```
generos = []
for lista in lista_generos:
    for genero in lista:
        generos.append(genero)
```

Esto nos permite poder contar cada género que se encuentra guardado, y así saber cuántos géneros se repiten o se presentaron más en el festival. Una vez hecho eso se realiza un for para poder mostrar en pantalla los 5 géneros más escuchados en cada día del festival.

```
frec_coachella = Counter(generos)
print("los 5 generos mas escuchados en el dia 1:")
print('{:<20}\t{:>12}'.format("Genero", "Apariciones"))
for genero in frec_dia1.most_common(5):
    print('{:<20}\t{:>12}'.format(genero[0], genero[1]))
print(frec_dia1.most_common()[0][0])

print("-----")

print("los 5 generos mas escuchados en el dia 2: ")
print('{:<20}\t{:>12}'.format("Genero", "Apariciones"))
for genero in frec_dia2.most_common(5):
    print('{:<20}\t{:>12}'.format(genero[0], genero[1]))
print(frec_dia2.most_common()[0][0])

print("-----")

print("los 5 generos mas escuchados en el dia 3: ")
print('{:<20}\t{:>12}'.format("Genero", "Apariciones"))
```

También los resultados se pueden mostrar en un archivo de texto.

```
#DATOS GUARDADOS EN ARCHIVO
"""
fo = open("CALCULOS_PIA_E3.txt", "w")
fo.write("Los 5 géneros más escuchados en el dia 1")
fo.write(str(frec_dia1.most_common(5)))

fo.write("Los 5 géneros más escuchados en el día 2:")
fo.write(str(frec_dia2.most_common(5)))

fo.write("Los 5 géneros más escuchados en el día 3:")
fo.write(str(frec_dia3.most_common(5)))

fo.write("Los 5 géneros más escuchados en COACHELLA:")
fo.write(str(frec_coachella.most_common(5)))

fo.write(f"\nSuma total de seguidores día 1: {suma_dia1}\n")
fo.write(f"Suma total de seguidores día 2: {suma_dia2}\n")
fo.write(f"Suma total de seguidores día 3: {suma_dia3}\n")
fo.write(f"Suma total de seguidores del festival: {suma_coachella}\n")

fo.close()
"""
```

Se encuentra comentado, pero los mismos resultados que se muestran en pantalla son los mismos que se guardan en el archivo.