



PROGRAMACIÓN EN PYTHON

Desde los principios básicos
hasta la visualización de datos



Proyecto de Módulo 3 Billboard Hot 100

Objetivo general

El objetivo de este proyecto es crear una aplicación para manejar y almacenar un listado de canciones y calcular algunas estadísticas sobre estas. En el desarrollo de esta aplicación pondrás en práctica los conceptos del módulo 3.

Objetivos específicos

En el desarrollo de este proyecto mostrarás tu capacidad para:

1. Leer y escribir archivos con formato CSV.
2. Implementar algoritmos de recorrido y modificación de listas.
3. Utilizar la estructura de datos compleja de diccionarios dentro de listas.
4. Usar la técnica “Dividir y Conquistar”.

Instrucciones generales

La descripción de la aplicación, que se presenta al comienzo, te permitirá conocer el alcance y las funcionalidades esperadas de la aplicación que desarrollarás en este proyecto. Las tres actividades, descritas posteriormente, detallan lo que debes realizar y al final, se especifica su forma de entrega.

Ten en cuenta que a lo largo de la descripción de las actividades encontrarás secciones con el título "**ATENCIÓN**" con indicaciones para conseguir que el resultado de la valoración y calificación de tu proyecto corresponda con lo esperado. Es importante que las sigas cuidadosamente.

Antes de empezar, te sugerimos leer con atención todo el proyecto. Mientras lo lees, trata de reconocer los conceptos del módulo que tendrás que poner en práctica.

Recuerda que este proyecto debe realizarse de forma **completamente individual**.

Descripción de la aplicación

Billboard es una revista semanal estadounidense y una lista especializada en información sobre la industria musical. También es una de las revistas especializadas más antiguas del mundo. Mantiene varias listas musicales, reconocidas internacionalmente, en las que se encuentran las canciones y álbumes más populares, clasificados en varias categorías. La lista más famosa, «Billboard Hot 100», clasifica las 100 primeras canciones independientemente del género musical al que pertenezcan, basándose en las ventas físicas, digitales, streaming y emisiones de radio.

Se desea crear una aplicación para manejar y almacenar el “Billboard Hot 100” desde 1965 hasta 2015 y calcular algunas estadísticas sobre las canciones que hacen parte del Top 100 de cada año.

El programa debe ser capaz de recibir las canciones a partir de un archivo en formato CSV (Comma-Separated Values) que tiene 5 columnas:

- La posición de la canción en el ranking (listado) de Billboard en un año particular.
- El nombre de la canción.
- El nombre del artista que interpreta la canción.
- El año del listado al cual pertenece la canción.
- La letra de la canción.

Por ejemplo, el siguiente archivo sería válido para ser cargado dentro del sistema:

```
Rank,Song,Artist,Year,Lyrics
1,wooly bully,sam the sham and the pharaohs,1965,sam the sham miscellaneous.....
2,i cant help myself sugar pie honey bunch,four tops,1965, sugar pie honey bunch...
3,i cant get no satisfaction the rolling stones,1965,NA
```

Nota: ten en cuenta que aquellas canciones que tienen NA en el campo de su letra, son únicamente instrumentales o el csv no contiene la letra correspondiente.

La aplicación debe permitir al usuario ejecutar las siguientes acciones:

1. Cargar un archivo con las canciones del listado de Billboard.
2. Buscar una canción dado el nombre y el año del ranking al cual pertenece y mostrar por pantalla toda su información: posición, año, nombre de la canción, nombre del artista y letra.
3. Consultar las canciones de un año dado. De cada canción se debe mostrar toda su información a excepción de la letra.
4. Consultar las canciones de un artista dado en un periodo de tiempo definido por el usuario (rango de años). De cada canción se debe mostrar toda su información a excepción de la letra.
5. Consultar todas las canciones de un artista dado. De cada canción se debe mostrar toda su información a excepción de la letra.
6. Consultar todos los artistas que han interpretado una canción dada.
7. Consultar los artistas más populares. Esto es, los nombres de los artistas que han tenido más de un número determinado de canciones (dado por el usuario) a lo largo de todos los años en el Billboard y la cantidad de canciones que ha tenido cada uno de ellos. Si un artista tiene la misma canción en 2 años diferentes esta se deberá contar dos veces.
8. Consultar el artista estrella de todos los tiempos. Esto es, el nombre del artista que ha tenido el mayor número de canciones a lo largo de todos los años en el Billboard y la cantidad de canciones que ha tenido. Si un artista tiene la misma canción en 2 años diferentes esta se deberá contar dos veces.
9. Consultar la lista completa de artistas del Billboard junto con sus canciones. Esto es, los nombres de los artistas y para cada uno de estos, los nombres de sus canciones. Si un artista aparece más de una vez en la lista de Billboard, se debe mostrar una sola vez. De igual manera, si una canción de un artista aparece más de una vez en la lista de Billboard, se debe mostrar una sola vez.
10. Consultar la cantidad promedio de canciones que los artistas tienen en el listado de Billboard.

Actividad 1: Preparación del ambiente de trabajo

1. Crea una carpeta de trabajo.
2. Descarga de Coursera el archivo con el “esqueleto” del proyecto (m3-billboard-esqueleto.zip) que se encuentra en la guía de trabajo y descomprímelo en tu carpeta de trabajo. El esqueleto consiste en un conjunto de archivos que vas a usar o a modificar.
3. Abre Spyder y reemplaza la carpeta de trabajo por la carpeta con el esqueleto.

Actividad 2: Construir el módulo de funciones

4. Usando Spyder, crea en tu carpeta de trabajo un nuevo archivo con el nombre “billboard.py”. En este archivo tu vas a construir el módulo en el que vas a implementar las funciones que responden a los requerimientos de la aplicación.

Define, documenta e implementa las funciones descritas a continuación en tu nuevo archivo.

Función 1: implementa una función que reciba como parámetro el nombre de un archivo que contiene la información de canciones y la cargue en el programa bajo la forma de **una lista de diccionarios**. Cada canción se va a representar utilizando un diccionario que tiene las siguientes llaves:

Llave	Descripción de la llave
<i>posicion</i>	Es un entero que indica la posición que ocupa la canción en el ranking (listado) en un año particular.
<i>nombre_cancion</i>	Es una cadena con el nombre de la canción.
<i>nombre_artista</i>	Es una cadena con el nombre del artista que interpreta la canción.
<i>año</i>	Es un entero que indica el año del ranking al cual pertenece la canción.
<i>letra</i>	Es una cadena con la letra de la canción.

Esta función debe retornar la lista de canciones (diccionarios) creada.

Función 2: implementa una función que reciba como parámetro la lista completa de canciones (diccionarios), el nombre de una canción y el año del ranking al cual pertenece dicha canción y retorne un diccionario con toda la información de dicha canción. En caso de que no se encuentre la canción, esta función debe retornar el valor None.

Función 3: implementa una función que reciba por parámetro la lista completa de canciones (diccionarios) y un año y retorne una nueva lista de diccionarios con la información de las canciones que pertenecen al año que entra por parámetro. Cada diccionario debe guardar la información de una canción, exceptuando su letra. En caso de que no haya ninguna canción en el año que se recibe por parámetro, la función debe retornar una lista vacía.

Función 4: implementa una función que reciba por parámetro la lista completa de canciones (diccionarios), el nombre de un artista, un año inicial y un año final y retorne una nueva lista de diccionarios con la información de las canciones del artista dado por parámetro en el periodo de tiempo definido por los parámetros año inicial y año final. Cada diccionario debe guardar la información de una canción, exceptuando su letra. En caso de que no haya ninguna canción del artista en ese periodo de tiempo, la función debe retornar una lista vacía.

Función 5: implementa una función que reciba por parámetro la lista completa de canciones (diccionarios) y el nombre de un artista y retorne una nueva lista de diccionarios con la información de las canciones que

pertenecen al artista que entra por parámetro. Cada diccionario debe guardar la información de una canción, exceptuando su letra. En caso de que no se encuentre el artista, la función debe retornar una lista vacía.

Función 6: implementa una función que reciba por parámetro la lista completa de canciones (diccionarios) y el nombre de una canción y retorne una lista de strings con los nombres de los artistas que han interpretado la canción que entra por parámetro. En caso de que no se encuentre la canción, la función debe retornar una lista vacía.

Función 7: implementa una función que reciba por parámetro la lista completa de canciones (diccionarios) y un valor entero que determina una cantidad mínima de canciones y retorne un diccionario cuyas llaves son los nombres de los artistas que han tenido un número total de canciones (a lo largo de todos los años) superior a la cantidad recibida por parámetro y cuyos valores son la cantidad de canciones que ha tenido cada uno de los artistas. Si un artista tiene la misma canción en 2 años diferentes esta se deberá contar dos veces. En caso de que ningún artista haya tenido un número de canciones superior al valor recibido por parámetro, la función debe retornar un diccionario vacío.

Función 8: implementa una función que reciba por parámetro la lista completa de canciones (diccionarios) y retorne un diccionario cuya llave es el nombre del artista que ha tenido el mayor número de canciones a lo largo de todos los años en el Billboard y cuyo valor es la cantidad de canciones que ha tenido dicho artista. Si un artista tiene la misma canción en 2 años diferentes esta se deberá contar dos veces.

Función 9: implementa una función que reciba por parámetro la lista completa de canciones (diccionarios) y retorne un diccionario cuyas llaves son los nombres de los artistas y los valores son la lista de canciones de cada artista. Si una canción de un artista aparece más de una vez en la lista de Billboard, esta se debe incluir una sola vez en el diccionario de retorno.

Función 10: implementa una función que reciba por parámetro la lista completa de canciones (diccionarios) y retorne la cantidad promedio de canciones que los artistas tienen en el listado de Billboard. Esto es:

$$\text{Canciones promedio por artista} = \text{cantidad total de canciones diferentes} / \text{cantidad de artistas diferentes}$$

Nota que, al calcular la cantidad total de artistas, hay que tener cuidado para que se cuente una sola vez cada artista diferente. Es decir que si un artista aparece varias veces, sólo debe contarse una vez. Así mismo, las canciones de un artista que aparecen más de una vez asociadas al artista, deben contarse una sola vez.

Actividad 3: Completar la interfaz de usuario basada en consola

5. En esta actividad tú tienes que construir la interfaz basada en consola para que el usuario interactúe con la aplicación. Para construir esta interfaz debes completar el archivo `consola_billboard.py`, la cual ya tiene una parte implementada que te facilitará tu trabajo. Debes modificar los elementos marcados con la etiqueta `TODO`.

Actividad 4: Probar el correcto funcionamiento del programa

6. Prueba la interfaz por consola ejecutando el archivo `"consola_billboard.py"`. Verifica que las funcionalidades de tu aplicación se comporten de acuerdo con lo esperado.
7. Prueba cargando el archivo `billboard.csv` o crea tu propio archivo de prueba respetando el mismo formato.

Entrega

8. Comprime tu carpeta de trabajo con tu proyecto resuelto como un .zip.
9. Entrega el archivo comprimido a través de Coursera en la actividad designada como **“Evaluación por pares del Proyecto M3”**.