



ISIS-1221

INTRODUCCIÓN A LA PROGRAMACIÓN

Nivel 3 – Laboratorio 4

Lectura de archivos y recorrido de diccionarios

Objetivos

1. Practicar la lectura de archivos con formato CSV.
2. Ejercitarse la implementación de algoritmos de recorrido de diccionarios y de listas.
3. Familiarizarse con estructuras complejas de datos, como un diccionario de listas de diccionarios.

Preparación del ambiente de trabajo

Para este laboratorio se le proporcionará un módulo de consola (`consola.py`) que contiene un menú interactivo para probar sus funciones.

Usted debe crear un nuevo módulo de funciones llamado `modulo.py` en el que escribirá las funciones correspondientes a las actividades 1 y 2 de este laboratorio. La consola importará automáticamente sus funciones desde este módulo.

También se le proporcionará un archivo CSV llamado `cupicharts.csv` que contiene información sobre canciones y sus características. El archivo CSV contiene datos organizados en columnas, donde cada fila representa una canción diferente.

Estructura del archivo CSV

El archivo `cupicharts.csv` contiene información musical organizada en las siguientes columnas:

Cuadro 1: Descripción de las columnas del archivo: `cupicharts.csv`

Posición	Nombre de la columna	Descripción	Tipo	Ejemplo
0	<code>title</code>	Título de la canción	str	“Die With a Smile”
1	<code>chart_week</code>	Fecha en que alcanzó el chart en formato: “YYYY-MM-DD”	str	“2025-06-28”
2	<code>performer</code>	Artista o intérprete de la canción	str	“Lady Gaga”
3	<code>peak_pos</code>	Posición máxima alcanzada en el chart	int	1
4	<code>wks_on_chart</code>	Número de semanas que la canción ha estado en el chart	int	44
5	<code>album_name</code>	Nombre del álbum al que pertenece la canción	str	“Die With a Smile”
6	<code>release_date</code>	Fecha de lanzamiento de la canción en formato: “YYYY-MM-DD”	str	“2024-08-16”
7	<code>popularity</code>	Popularidad de la canción en una escala del 1 al 100	int	98
8	<code>explicit</code>	Indica si la canción tiene contenido explícito	bool	False
9	<code>listeners</code>	Número de oyentes que han escuchado la canción	int	1412509
10	<code>play_count</code>	Número de reproducciones de la canción	int	23765285
11	<code>duration_s</code>	Duración de la canción en segundos	float	251.667
12	<code>genre</code>	Género musical de la canción	str	“art pop”

Actividad 1: Cargar datos desde archivo CSV

Escriba una función `cargar_cupicharts` que reciba como parámetro una cadena de caracteres con la ruta al archivo CSV y retorne un diccionario que organice la información de las canciones según su género musical.

Organice los datos en un diccionario, donde las llaves sean los nombres de los géneros musicales (strings en minúscula, sin espacios al inicio o al final). Los valores deben ser listas de diccionarios, donde cada elemento de la lista (cada diccionario) represente una canción de ese género, con sus atributos (a excepción del atributo género).

La función debe:

- Leer el archivo CSV línea por línea.
- Para cada línea (que representa una canción), extraer todos los campos y organizarlos en un diccionario.
- Agrupar las canciones por género musical, donde el género será la llave del diccionario principal.
- El género musical debe estar en minúscula y sin espacios al inicio o al final.
- Cada canción se almacenará como un diccionario con todos sus campos, excepto el campo `genre` (ya que el género es la llave del diccionario principal).

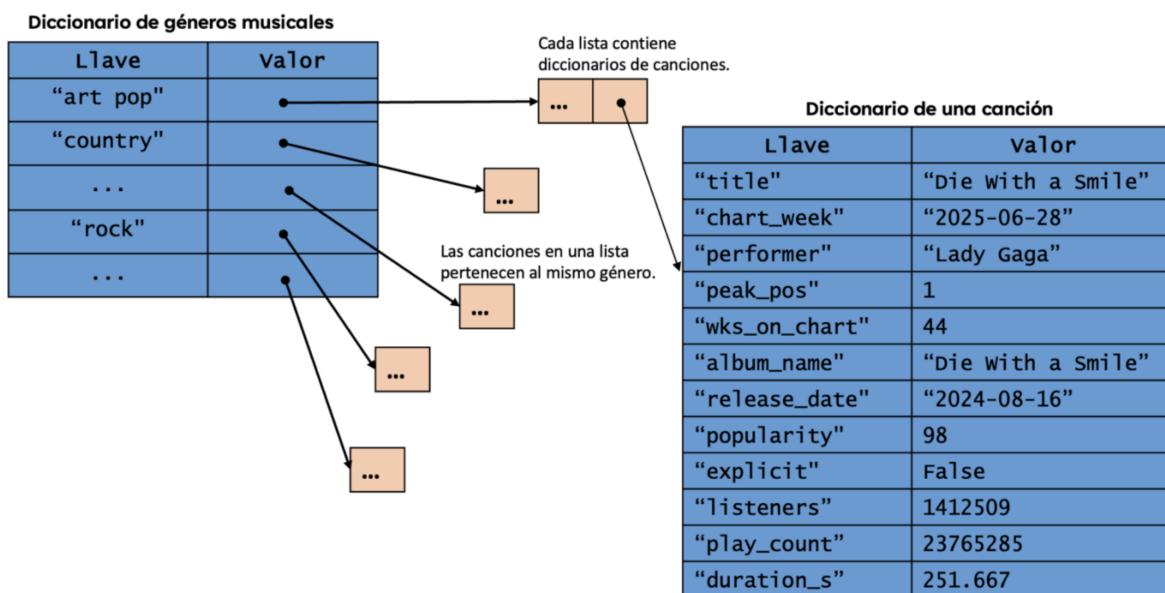


Figura 1: Ejemplo de la estructura deseada para el manejo de los datos.

Notas importantes: Al usar `readline()`, agregue `strip()` de la siguiente forma, para garantizar que se eliminan los saltos de línea: `readline().strip()`. Al usar `open()`, agregue la codificación "utf-8" de la siguiente forma, para garantizar la lectura de caracteres especiales del archivo CSV: `open(archivo, "r", encoding="utf-8")`. Para más información, consulte la documentación de `str.strip():` <https://docs.python.org/es/3/library/stdtypes.html#str.strip> y la documentación de `readline():` <https://docs.python.org/es/3/tutorial/inputoutput.html#methods-of-file-objects>.

Estructura del retorno: El diccionario returned debe tener la siguiente estructura:

```
{
    "pop": [
        {
            "title": "Canción 1",
            "performer": "Artista 1",
            "peak_pos": 5,
            ...
        },
        ...
    ],
    "rock": [
        {
            "title": "Canción 2",
            "performer": "Artista 2",
            "peak_pos": 3,
            ...
        },
        ...
    ],
    ...
}
```

Actividad 2: Canción más popular por género

Escriba una función `cancion_mas_popular_por_genero` que reciba como parámetros un diccionario de canciones (el mismo formato returnedo por `cargar_cupicharts`) y el nombre de un género musical, y retorne el título de la canción con mayor popularidad en ese género.

La función debe:

- Buscar el género especificado en el diccionario.
- Si el género no existe, debe retornar `None`.
- Si el género existe, recorrer todas las canciones de ese género y encontrar la que tenga el mayor valor en el campo `popularity`.
- Si hay empate en la popularidad máxima, debe retornar el título de la canción que aparezca primero en la lista.
- Retornar el título (campo `title`) de la canción más popular.

Si en el género “pop” hay canciones con popularidad 95, 98 y 95, la función debe retornar el título de la canción con popularidad 98. Si todas las canciones de “pop” tienen popularidad 95, debe retornar el título de la primera canción en la lista con esa popularidad.

Entrega

Entregue el archivo `modulo.py` a través de Brightspace en el laboratorio del Nivel 3 designado como “N3-L4: Lectura de archivos y recorrido de diccionarios”.

Puede usar el módulo `consola.py` proporcionado para probar sus funciones antes de la entrega. Su archivo debe funcionar correctamente desde la consola. Asegúrese de tener el archivo `cupicharts.csv` en el mismo directorio para probar su implementación.