

En este proyecto trabajaremos con datos correspondientes a un listado de 1000 películas producidas entre 1930 y 2019. Los datos están tomados de Kaggle.com, donde se pueden encontrar diversas bases de datos sobre una gran variedad de temas.

Representaremos la información de entrada mediante listas de tuplas, y a partir de esta estructura implementaremos una serie de funciones que nos permitirán realizar varios tipos de consultas.

Trabajaremos con ficheros en formato CSV. Cada registro del fichero de entrada ocupa una línea y contiene ocho atributos sobre las películas(movie_name, genre, year, timeMin, imdb, metascore, votes, us_grossMillions). Estas son las primeras líneas del fichero de entrada:

1	movie_name	genre	year	timeMin	imdb	metascore	votes	us_grossMillions
2	Once Upon a Time... in Hollywood	Comedy, Drama	2019	161	77	830	439202	1425
3	Parasite	Comedy, Drama, Thriller	2019	132	86	960	367763	5337
4	Knives Out	Comedy, Crime, Drama	2019	131	80	820	272176	16536
5	1917	Drama, War	2019	119	84	780	278731	15923
6	The Ten Commandments	Adventure, Drama	1956	220	78	774	60668	9374
7	Little Women	Drama, Romance	2019	135	79	910	92139	1081
8	Avengers: Endgame	Action, Adventure, Drama	2019	181	84	780	697863	85837
9	Joker	Crime, Drama, Thriller	2019	122	85	590	754563	33545
10	The Shawshank Redemption	Drama	1994	142	93	800	2220730	2834

Para almacenar en Python la información de cada una de las líneas se usará la siguiente definición de namedtuple:

Registro = namedtuple("Registro", "movie_name, genre, year, timeMin, imdb, metascore, votes, us_grossMillions")

En el fichero movies.py se han implementado las siguientes funciones:

1. **lectura_fichero:** Recibe la ruta de un ficheroCSV codificado en UTF-8 y devuelve una lista de tuplas de tipo Registro(str, str, int, int,int,int,int,int)conteniendo todos los datos almacenados en el fichero.
2. **mostrar_cartelera_registros:** Muestra el conjunto de todas las películas registradas.
3. **mostrar_peliculas_por_año:** La función devuelve una lista de tuplas con las películas junto a su género y duración de un determinado año.
4. **suma_votos_imdb:** Muestra por pantalla el número de votos totales para las películas catalogadas por el mismo imdb.

5. **media_por_año**: Devuelve la valoración media del `us_grossMillions` de un año, en caso de no seleccionar ningún año devolverá la media del año más reciente del csv (2019).
6. **pelicula_menor_duracion**: Muestra la(s) película(s) de menor duración junto a su valoración de un año determinado.
7. **mayores_votos_por_año**: Accedemos a las `n` películas con mayor registro de votos del cualquier año marcado como parámetro.
8. **películas_por_genero**: Genera un diccionario que contiene como clave los distintos géneros o conjuntos de géneros de una película y asocia a ellos las películas como valores.
9. **películas_valoradas_años**: Obtenemos todas las películas existentes en el csv mejor valoradas, mostrando las `n` primeras, serán imprimidas mostrando los años más actuales.
La función cuenta con una función auxiliar para crear una lista con las películas y su valoración ordenadas de mayor a menor.

Incluye un fichero **test_movies.py** donde importa todas las funciones del módulo **movies** las cuales son testeadas y mostradas por consola.