

PRÁCTICA 3 – TAD

David Fernández Reimúndez – david.fernandez.reimundez@udc.es

Hugo Varela Sánchez – hugo.varela.sanchez@udc.es

Hemos desarrollado un programa que opera con datos de distintas películas para obtener distintos resultados: ordenación, eliminación de repetidas, extracción de información según sus atributos o la obtención de varias métricas. Para ejecutar este programa, es necesario aportar un archivo de texto en el que venga la información de todas las películas. Los datos de una película se corresponden con una línea del archivo. Además cada línea debe de tener cuatro datos, separados por un **punto y coma(,)**, y ofrecer la siguiente información en este orden:

{Director};{Nombre_Película};{Año_Estreno};{Puntuación}

Los atributos {Director} y {Nombre_Película} se tratarán como tipo “**string**”, {Puntuación} como un “**float**” y {Año_Estreno} como un “**int**”

Este programa se ejecutará desde una terminal.

Los tres/cuatro archivos necesarios para la ejecución(tres archivos .py y un archivo de texto) deben estar en el mismo directorio para evitar errores.

Una vez que desde la terminal se haya llegado a la ruta correcta, se tendrá que ejecutar el archivo main.py con **python3**:

python3 main.py

Sin embargo, esto dará error porque también hay que pasar el archivo de los datos de las películas, quedando finalmente algo así:

python3 main.py archivo.txt

FASES DE DESARROLLO

Respecto a los archivos de código, comenzamos la práctica creando un archivo llamado class_peliculas.py

Aquí, creamos una clase llamada Película, en la que se van asociando los valores de cada línea pasados por el archivo de texto en un solo elemento(del tipo clase película). Además de estos cuatro atributos, para los cuales solo definimos método **getter** (no cambiarán), decidimos crear varios métodos que posteriormente nos ayudaron en la realización de partes de la práctica, como los métodos mágicos de comparación para poder ordenar las listas, o de recopilación de datos para realizar los dataframes con la librería pandas, además del “__str__”. Quedando entonces los siguientes métodos:

__eq__ __str__ __ge__ __gt__ datavalues

Todos ellos están explicados en la documentación de la clase Película.

Además en este archivo class_pelicula.py creamos una excepcion que posteriormente usamos para el manejo de selecciones no válidas en los menús utilizados a lo largo de la práctica:

NumberNotInMenu

Creado este archivo, nos pasamos a escribir el código del main.py. Lo primero que hicimos, aparte de importar al código el archivo de la clase “Película” y el/los archivos de la clase

“ArrayOrderedPositionalList”/“LinkedOrderedPositionalList”, fue programar la extracción de la información del archivo de texto.

Para sacar los datos del archivo de texto hay que utilizar el open de Python y leer el archivo como texto. Una vez guardado en una variable esa lectura, procedemos a acceder a cada línea del archivo mediante la separación por líneas de todo el texto, en Python con un .split('\n'). Por último se revisa que la longitud de la línea sea mayor que cero y que haya cuatro datos mediante otro .split(',') de cada línea pero en este caso por separación por punto y coma(;) y no por saltos de línea.

Una vez guardado, nos dispusimos a conseguir todas las variables que nos harían falta en algún momento de la práctica. Por cada línea se obtienen los datos de una película, y creado el objeto de la **clase película** se añade con el método .add() a la lista ordenada que ordena ya automáticamente, y conseguimos una lista posicional ordenada.

Tras eso, buscamos crear otra lista en la que no se almacenarán repetidos. Para ello decidimos hacer una copia de la lista de películas para no modificar la original, desde esa fuimos comparando el primer elemento con el segundo. Si eran iguales(en base a los atributos que se comparan) entre sí simplemente se elimina el primero y a la lista de no repetidos no se añadía nada. Si eran diferentes se añade a la lista de no repetidos el primero y posteriormente este se elimina, siendo así el que era segundo el nuevo primero. Todo esto se repite de forma indefinida por el bucle **while True**, hasta que se hace un **break** cuando el elemento segundo de la lista es igual al último y en este caso se añaden los dos(primer y segundo) y termina el bucle.

Ya con las dos listas, creamos los Data Frames relacionados a cada lista. Hacemos para cada lista un bucle, donde metemos en una lista auxiliar listas con los valores de cada película obtenidos con el método datavalues de la clase Pelicula. De ahí, añadimos la lista de listas a un dataframe.

Una vez obtenidas todas las herramientas, es hora de jugar con ellas.

Creamos dos funciones que son las que van a mover el programa:

menus accion

Cada iteración del bucle, llama a menus(1). La función menus almacena todos los menús de la práctica, pidiéndote poner a que menú acceder al llamarlo. Cada uno devuelve un número entre 1-n siendo n el número de opciones. Ese valor que devuelve decide el camino que toma el programa. Después de esto se llama a accion, con el valor que devuelve menus(1). Depende de cada acción seguirá un camino u otro mientras recurre a menus para poder llegar a lo que se desea.

Ofrecemos un esquema que refleja muy bien el funcionamiento del programa, y de la función menus. Se recomienda tener dicho esquema abierto, durante la ejecución del programa ya que ofrece una visión general de la ejecución

Además para esta práctica, hemos decidido implementar la posibilidad de elegir siempre como quieres hacer las cosas, por ejemplo: selección de qué tipo de estructura de datos se quiere usar, elección de Data Frames o listados tabulados para mostrar los datos, o coger el listado de todas las películas o solo las que no están repetidas. Todo esto para evitar malentendidos profesor-alumno, durante las explicaciones de las sesiones presenciales o con el propio enunciado de la práctica.