OBSERVACIONES DEL LA PRACTICA

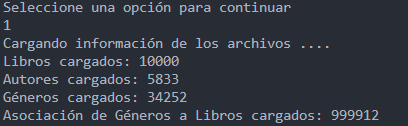
Estudiante 1 Cod 202021368

Estudiante 2 Cod 201914771

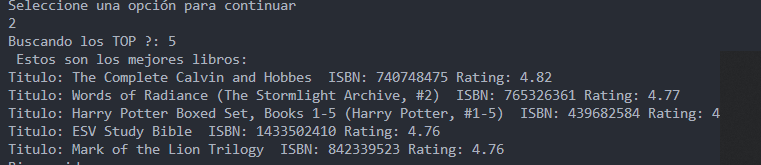
1. **¿Cuáles son los mecanismos de interacción (I/O: Input/Output) que tiene el view.py con el usuario?**

Los mecanismos INPUT que tiene view.py con el usuario se encuentran en el menú principal. En este se genera un menú en el cual se le muestran 5 opciones que el usuario puede ejecutar:

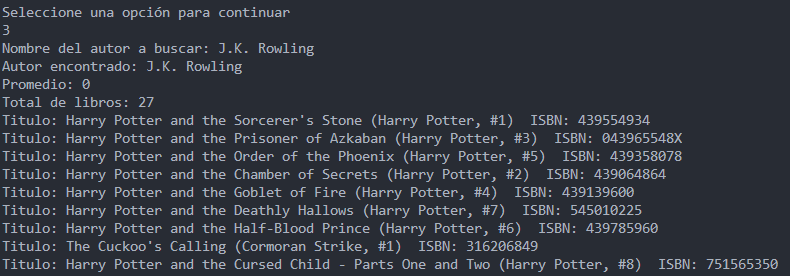
* + - 1. Cargar la información en el catálogo: su OUTPUT es información básica sobre la cantidad de libros, autores y géneros cargados después de haber guardado la información en listas.



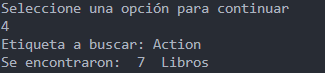
* + - 1. Consultar el top X (X: INPUT) libros por promedio: su OUTPUT son los nombres de los libros ordenados por su puntaje junto con su ISBN.



* + - 1. Consultar los libros de un autor (autor: INPUT): su OUTPUT son los libros del autor escogido y el número de libros encontrados.



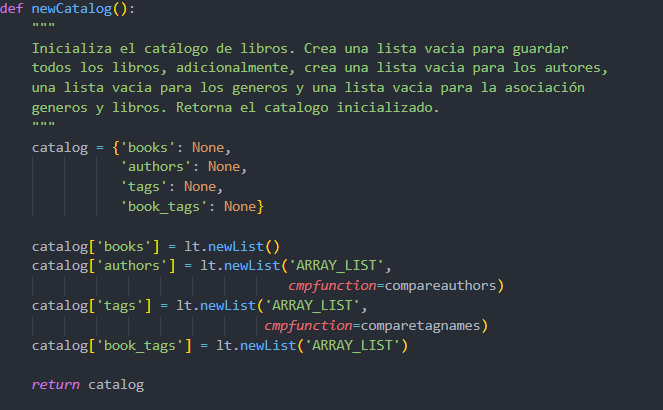
* + - 1. Libros por género (género: INPUT): su OUTPUT es la cantidad de libros del género ingresado.



* + - 1. Salir de la aplicación

1. **¿Cómo se almacenan los datos de GoodReads en el model.py?**

Los datos de GoodReads se almacenan en diccionario llamado “catalog” que tiene llaves para cada uno de los csv: books, authors, tags y books\_tags. Estos csv se guardan en linked lists y array lists como se puede evidenciar a continuación:



1. **¿Cuáles son las funciones que comunican el el view.py y el model.py?**

Las funciones que comunican el view.py y el model.py son las siguientes:

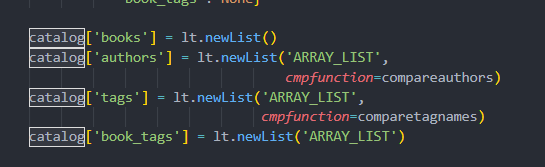
* + - 1. initCatalog()
      2. loadData()
         1. loadBooks()
         2. loadTags()
         3. loadBooksTags()
      3. sortBooks()
      4. getBooksByAuthor()
      5. getBestBooks()
      6. countBooksByTags()

1. **¿Cómo se crea una lista?**

La lista se crea en model.py de la siguiente manera:

* + - 1. lt.newlist(Args): los argumentos que se le puede incluir son los siguientes:
         1. datastructure: array list o single linked list
         2. cmpfunction: función de comparación
         3. key: identificador de comparación
         4. filename: si se provee, se crea una lista a partir de un archivo
         5. delimiter: si se pasa un archivo, se utiliza para separar los elementos

Ejemplo:

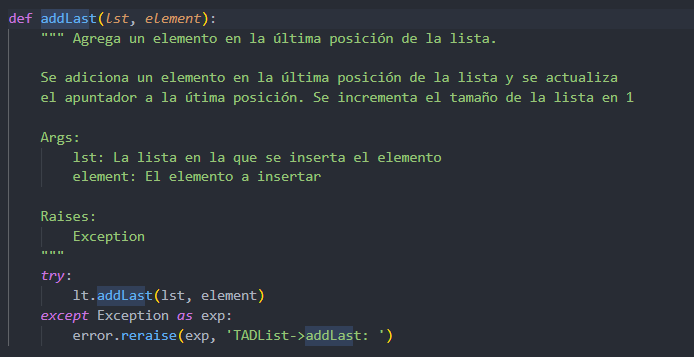


1. **¿Qué hace el parámetro cmpfunction=None en la función newList()?**

El parámetro cmpFunction = None en newList() indica que que se usará la función de comparación por defecto.

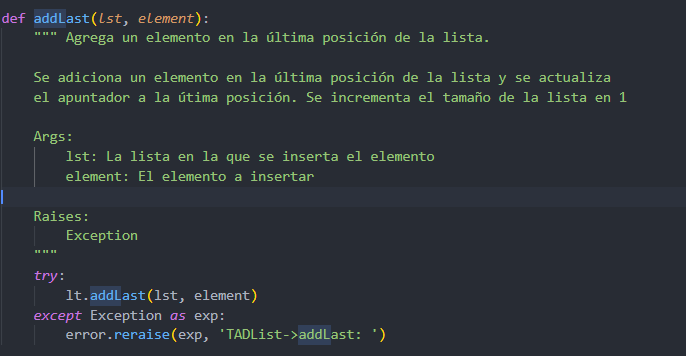
1. **¿Qué hace la función addLast()?**

La función addLast(lst,element) agrega un elemento dado al final de la lista.



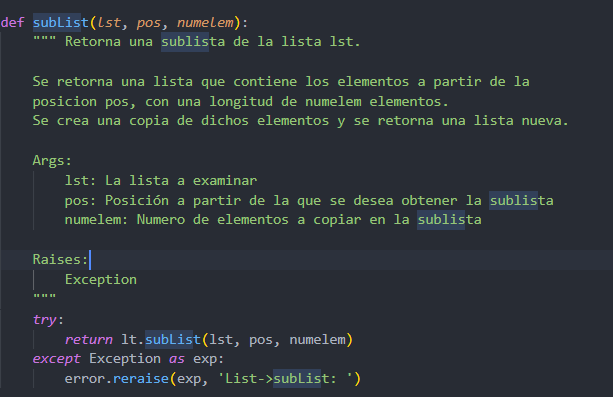
1. **¿Qué hace la función getElement()?**

La función getElement(lst,pos) retorna el elemento que se encuentra en la posición dada en los argumentos.



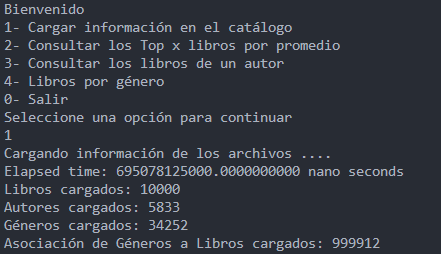
1. **¿Qué hace la función subList()?**

La función subList(lst,pos,numelem) retorna una sublista que contiene una copia de los elementos que están a partir de la posición dada con longitud numelem.

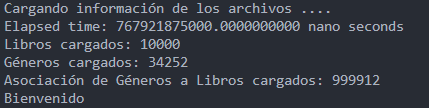


1. **¿Observó algún cambio en el comportamiento del programa al cambiar la implementación del parámetro “ARRAY\_LIST” a “SINGLE\_LINKED”?**

Array:



Linked list:



Al cambiar la implementación del parámetro “ARRAY\_LIST” a “SINGLE\_LINKED”, el programa se demoró más tiempo en ejecutarse.