OBSERVACIONES DEL LA PRACTICA

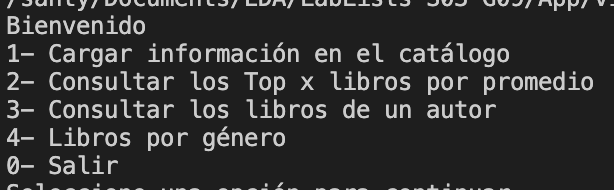
Santiago Pardo Morales - 202013025

Juan José Ramirez Cala - 202013228

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

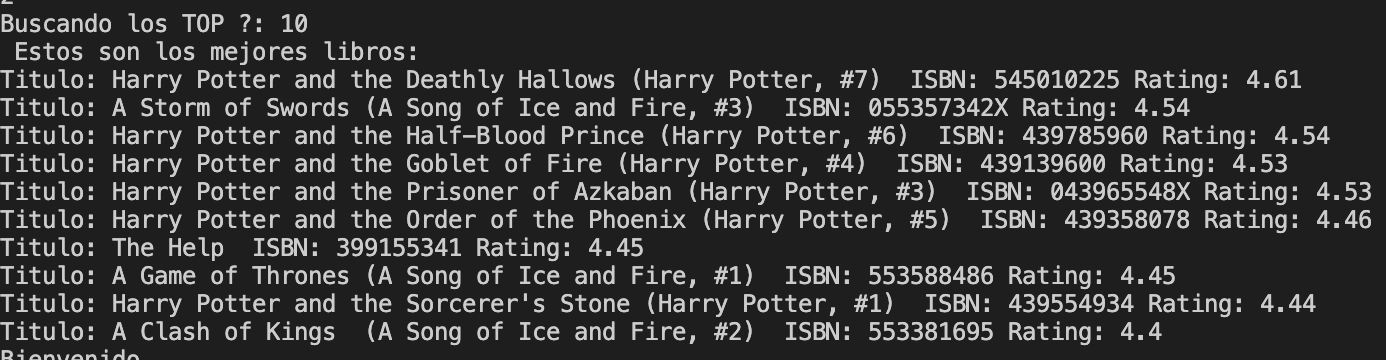
El mecanismo de interacción principal se encuentra dentro de los últimos comandos del archivo view, en el que se da por escoger al usuario entre los números 0 al cuatro dependiendo de la opción que quiera consultar.

En este caso, los inputs son los números que el usuario quiere insertar para consultar lo que quiere y el output es la continuación del menú dependiendo de la opción que escogió el usuario.

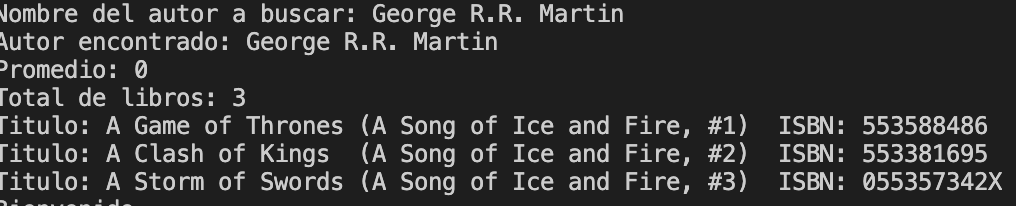


Cabe destacar que, si se quiere consultar algun detalle en específico, primero se debe presionar la opción 1, ya que los datos de GoodReads estarán almacenados dentro del menú

Por ejemplo, usando la opción 2, se utiliza un output que también sirve como un input (el número de libros promedio que quiere) para que se imprima el resultado final de la función

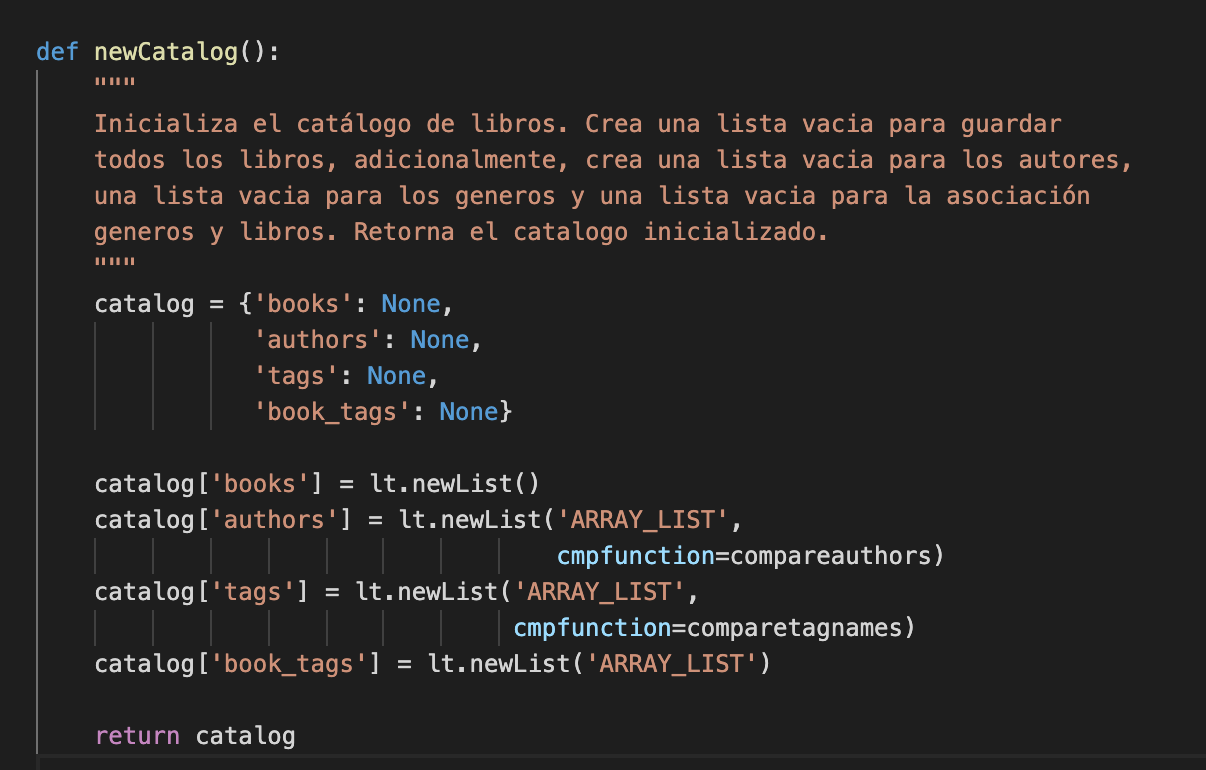


Así mismo, en la función 3, observamos que el output también sirve como un input (los libros de un autor en específico), cuyo output es la impresión final de los resultados



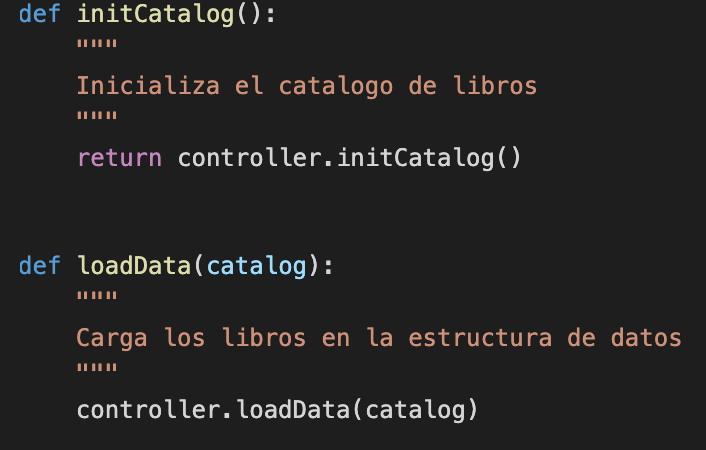
1. ¿Cómo se almacenan los datos de **GoodReads** en el **model.py**?

Los datos de Goodreads se utilizan para construir un “modelo inicial”, para ello se usa la función newCatalog(), la cual inicializa como tal el catálogo de los libros, a través de la creación de una lista vacia para guardar elementos como los libros, los autores, los géneros y la relación entre géneros y libros

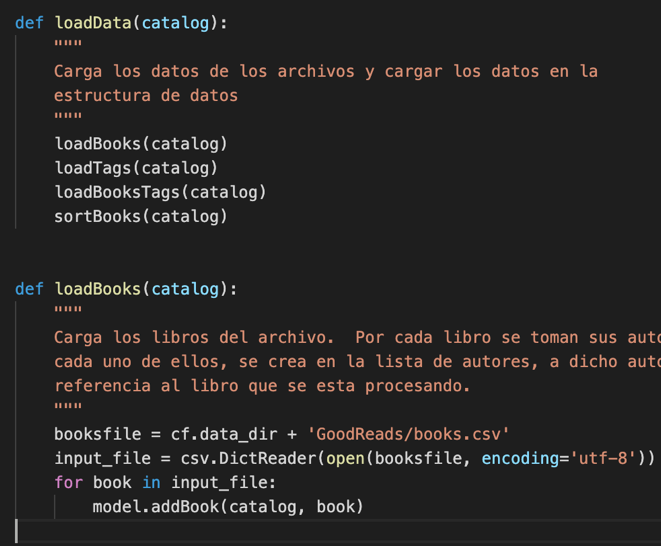


1. ¿Cuáles son las funciones que comunican el el **view.py** y el **model.py**?

Explícitamente, no existe alguna función dentro del archivo view que provenga del archivo model, incluso, el archivo model no fue importado dentro de este mismo archivo. Sin embargo, existen varias funciones cuyo origen proviene del archivo controller, como por ejemplo initCatalog() y loadData()



Haciendo una vista más detallada hacia el orígen de este tas funciones (el archivo controller), podemos observar que el archivo model si se encuentra importado, y existen varias funciones sacadas del archivo model, por lo que podemos concluir que las funciones del archivo controller sirven como medio de comunicación entre las funciones de los archivos model y view



1. ¿Cómo se crea una lista?

La lista se crea con la función newList, con ella podemos ver que tiene varios parámetros como el tipo de estructura de datos (SINGLE-LINKED), un identificador (key), el nombre del archivo, y un delimitador determinado por coma

Retorna una nueva lista, o en su defecto, una excepción la cual detecta un error dentro de la creación de la lista

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

Cuando está asignado el None se hace correr la función con la condición por defecto con la condición de que exista un key, caso contrario, el cmpfunction haría que los elementos de la lista entren en una comparación.

1. ¿Qué hace la funció **addLast()**?

Se adiciona un elemento pedido a la última posición de la lista, se actualiza el apuntador a la última posición y se incrementa el tamaño de la lista en 1.

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

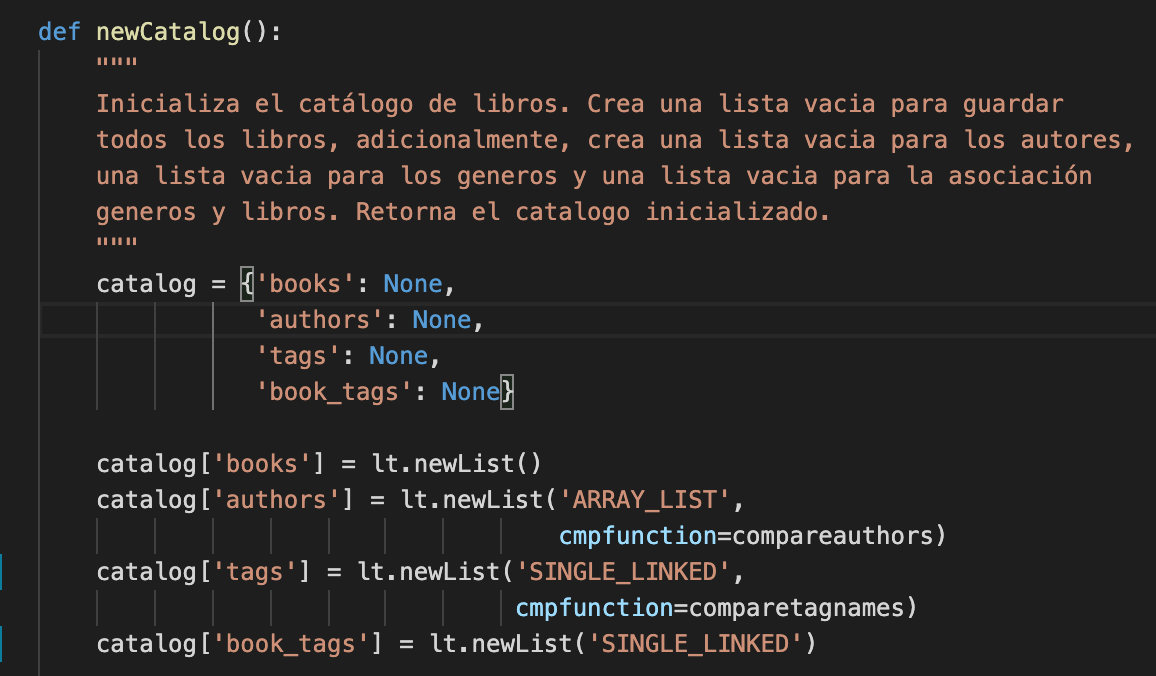
Recorre la lista en el rango de su tamaño, y retorna el elemento de una posicón pedida con el parámetro pos

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

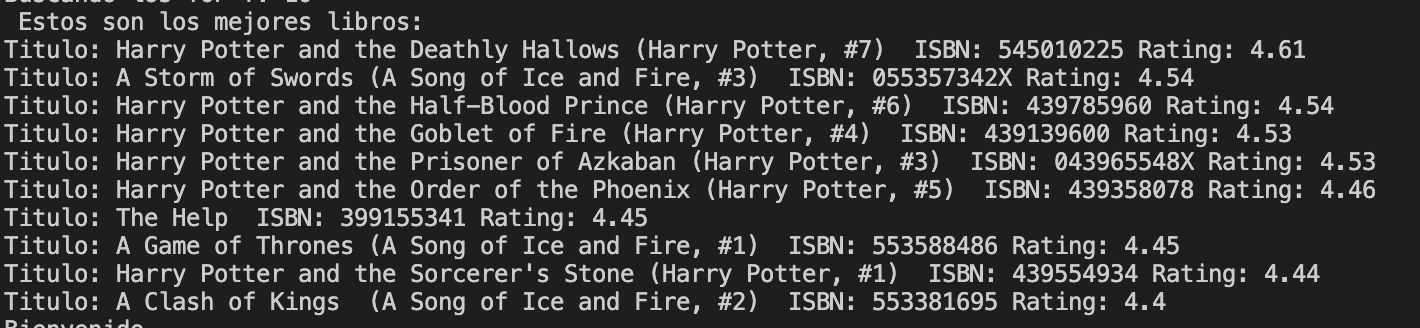
Retorna una lista con elementos a partir de la partir de la posición pos, con una longitud dicha por el parámetro numelem

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

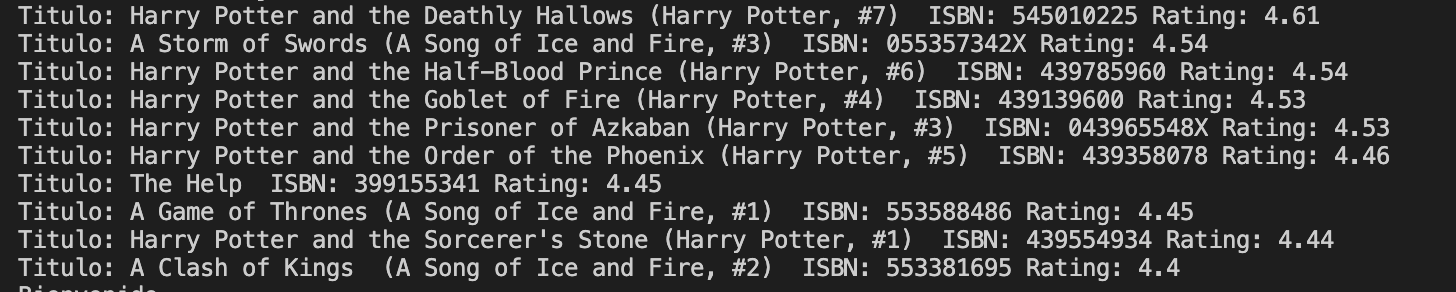
Cambios en la función newCatalog()



Funcionamiento de la consola antes del cambio en la función newCatalog()



Funcionamiento de la consola después del cambio en la función newCatalog()



No se observaron cambios mayoritarios dentro del funcionamiento del programa, a través del archivo view