OBSERVACIONES DEL LA PRACTICA

Daniel Arango Cruz Cod 202110646

Oscar Ivan Garica Montanez Cod 201630048

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

El usuario al ejecutar el programa se encuentra con el menú en el que inicialmente como input le solicitan un número que es transformado a un entero para que dependiendo de este se ejecute una función y como output despliega la información y código relacionado con esa función.

Funciones que el primer output puede tener y que inputs solicita:

Opción 1: En esta opción una vez recibido el input “1” del usuario se ejecuta la función initCatalog() que carga toda la información relacionada con los archivos .csv, para que de esta forma a manera de output muestre un resumen de la información cargada entre ellos: “Libros cargados”, “Géneros cargados” entre otros.

Opción 2: En esta opción luego de ejecutarse le solicita al usuario un número para que este a modo de input luego pueda mediante la función getBestBooks del controller filtrar y mostrar, a modo de output, el top de los mejores libros dentro del rango indicado por el usuario.

Opción 3 En esta opción al igual que la anterior es una función de consulta por lo que de input le solicita al usuario el nombre de un autor en especifico para buscar los libros que este escritor ha hecho. Eso por medio de la función getBooksByAuthor() que recibe el input del usuario y como output muestra una serie de información relacionada con el autor y los libros encontrados.

Opción 4: Esta función una vez mas es de consulta por medio de un string por lo que de input le solicita al usuario el filtro de “Genero” a modo de tag, para que de esta forma la función countBooksBytag reciba de parámetro la información suministrada por el usuario y a modo de output muestre la cantidad de libros encontrados con ese tag.

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

En el model.py los datos se almacenan dentro de un diccionario de listas de tipo Arraylist las cuales algunas tienen cierta función o tarea en especifica ya asignada por medio de otras funciones creadas, una de ellas es: compareauthors.

Estas listas inicialmente son vacías y luego llenadas por funciones como addBook, addBookAuthor, addTag y addBookTag. Adicionalmente hay funciones extras que en caso de querer agregar nueva información al diccionario de listas se pueda realizar mediante newTag, newBook, newAuthor.

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

Las funciones que comunican el view y el model en gran mayoría se encuentran en el controller.

Estas son: LoadData(),initCatalog(),Getbestbooks(),Getbooksbyauthor() y Countbooksbytag()

1. ¿Cómo se crea una lista?

Para crear una nueva lista se ejecuta la función newlist () que solicita ciertos parámetros para su creación, estos son: datastructure,cmpfunction,key,filename,delimiter. Todos estos requerimientos funcionan a modo de las características que la lista tendrá para que de esta forma al final la función retorne una nueva lista con la información cargada, ordenada y separada de la manera en la que fue indicada en los parámetros.

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

El parámetro cmpfunction en la función newlist se encarga de asignarle alguna función que se encargue de compara los datos que se encontrarán dentro de la lista. En el caso de que esta tenga el valor none, quiere decir que se tomara la comparación por defecto y debe asignársele un valor al key.

1. ¿Qué hace la funció **addLast()**?

La función addLast recibe como parámetro la lista en la que se desea agregar el elemento y lo que se quiera agregar en dicha lista, para que de esta forma asigne en la ultima posición de la lista el elemento solicitado en la lista brindada y incrementa el tamaño de la lista en 1.

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

La función getElement() recibe como parámetro la lista a la que se desea acceder y adicionalmente la posición en la que se encuentra el elemento, con esto la función retorna el elemento en la posición que brindó el usuario.

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

La función subList recibe de parámetros la lista de la que se desea tomar elementos, la posición desde la que se desea comenzar a obtener datos y el numero de elementos que desea que obtenga de la lista. De esta forma, la función retorna una nueva lista con los datos obtenidos de la lista recibida de parámetro.

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

Tras realizar varias pruebas con los dos tipos de estructuras de datos no llegamos a observar grandes cambios en el como se comportaba el programa, sin embargo, percibimos quizás una respuesta más rápida por parte del programa (probablemente fracciones de segundo muy pequeñas) al menos en la función de cargar datos, la cual puede ser de las funciones más tardadas en ejecutarse.