OBSERVACIONES DEL LA PRACTICA

Estudiante 1 Cod 201912123

Estudiante 2 Cod 201911107

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

Los mecanismos de interacción Input que el usuario pide son 5 opciones:

Cargar la información en el catálogo, consultar los top x libros por promedio,consultar los libros de un autor, libros por género y salir. Dependiendo de lo que se elija, se pregunta por el top x, el nombre del autor o la etiqueta del libro.

Los outputs muestran la información de los archivos (cantidad de libros, autores, géneros y Asociación de géneros)

También muestra los libros primero por el título, luego por el ISBN y el rating.

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

Se almacenan en 3 ARRAY\_LIST, uno para libros, uno para géneros y otro para autores.

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

Inicialización del Catálogo de libros:

initCatalog()

Funciones para la carga de datos:

loadData(catalog)

loadBooks(catalog)

loadTags(catalog)

loadBooksTags(catalog)

sortBooks(catalog)

Funciones de ordenamiento:

sortBooks(catalog)

Funciones de consulta sobre el catálogo:

getBooksByAuthor(catalog, authorname)

getBestBooks(catalog, number)

ountBooksByTag(catalog, tag)

1. ¿Cómo se crea una lista?

Para crear una lista, en list.py hay una función que se llama newlist, la cual pide:

El datastructure que es el tipo de estructura de datos que se va a utilizar, que puede ser ARRAY\_LIST y SINGLE\_LINKED. además, tiene cmpfunction, key, filename y delimiter.

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

cmpfunction: Es una funcion de comparacion para los elementos de la lista.

Si no se provee función de comparación se utiliza la función

por defecto pero se debe proveer un valor para key.

Si se provee una función de comparación el valor de Key debe ser None.

1. ¿Qué hace la funció **addLast()**?

Agrega un elemento en la última posición de la lista.

Se adiciona un elemento en la última posición de la lista y se actualiza

el apuntador a la útima posición. Se incrementa el tamaño de la lista en 1

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

Retorna el elemento en la posición pos de la lista.

Se recorre la lista hasta el elemento pos, el cual debe ser mayor

que cero y menor o igual al tamaño de la lista.

Se retorna el elemento en dicha posición sin eleminarlo.

La lista no puede ser vacia.

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

Retorna una sublista de la lista lst.

Se retorna una lista que contiene los elementos a partir de la

posicion pos, con una longitud de numero de elementos.

Se crea una copia de dichos elementos y se retorna una lista nueva.

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

Considero que el principal cambio que notamos fue que el tiempo de carga de la información del catálogo fue mayor que cuando se usó ARRAY\_LIST. Sin embargo, al implementar las otras opciones, el tiempo en el que respondió el programa parece ser el mismo tanto para ARRAY\_LIST como para cuando se usa SINGLE\_LIST.