OBSERVACIONES DE LA PRACTICA

Alejandro Hernández 202111716

Luisa Gabriela Hernández 202114093

1. ¿Cuáles son los mecanismos de interacción (I/O: Input/Output) que tiene el **view.py** con el usuario?
2. ¿Cómo se almacenan los datos de **GoodReads** en el **model.py**?
3. ¿Cuáles son las funciones que comunican el el **view.py** y el **model.py**?
4. ¿Cómo se crea una lista?
5. ¿Qué hace el parámetro **cmpfunction=None** en la función **newList()**?
6. ¿Qué hace la funció **addLast()**?
7. ¿Qué hace la función **getElement()**?
8. ¿Qué hace la función **subList()**?
9. ¿Observó algún cambio en el comportamiento del programa al cambiar la implementación del parámetro **“ARRAY\_LIST”** a **“SINGLE\_LINKED”**?

Primero el print del menú que muestra el menú al correr el programa: printMenu(), dentro de printMenu se encuentras estos prints:

1. def printMenu():
2. print("Bienvenido")
3. print("1- Cargar información en el catálogo")
4. print("2- Consultar los Top x libros por promedio")
5. print("3- Consultar los libros de un autor")
6. print("4- Libros por género")
7. print("0- Salir")

1.2 Hay un input para seleccionar una de las opciones del menú:

inputs = input('Seleccione una opción para continuar\n')

1.3 Luego, dentro de cada opción determinada por un if hay más prints:

if int(inputs[0]) == 1:

print("Cargando información de los archivos ....")

catalog = initCatalog()

loadData(catalog)

print('Libros cargados: ' + str(lt.size(catalog['books'])))

print('Autores cargados: ' + str(lt.size(catalog['authors'])))

print('Géneros cargados: ' + str(lt.size(catalog['tags'])))

print('Asociación de Géneros a Libros cargados: ' +

str(lt.size(catalog['book\_tags'])))

elif int(inputs[0]) == 2:

number = input("Buscando los TOP ?: ")

books = controller.getBestBooks(catalog, int(number))

printBestBooks(books)

elif int(inputs[0]) == 3:

authorname = input("Nombre del autor a buscar: ")

author = controller.getBooksByAuthor(catalog, authorname)

printAuthorData(author)

elif int(inputs[0]) == 4:

label = input("Etiqueta a buscar: ")

book\_count = controller.countBooksByTag(catalog, label)

print('Se encontraron: ', book\_count, ' Libros')

2. Se crean listas vacías para los libros, tags, autores, etc.; luego se devuelve ese catálogo inicializado y se definen varias funciones, cada una con el propósito de incluir elementos a una categoría diferente.

def newCatalog():

Catálogo nuevo y vacío

def addBook(catalog, book):

# Se adiciona el libro a la lista de libros

lt.addLast(catalog['books'], book)

# Se obtienen los autores del libro

authors = book['authors'].split(",")

# Cada autor, se crea en la lista de libros del catalogo, y se

# crea un libro en la lista de dicho autor (apuntador al libro)

for author in authors:

addBookAuthor(catalog, author.strip(), book)

Función que añade libros a la lista de libros dentro del catálogo

3. Las funciones que hacen referencia a controller ya que desde allí se llama a model para que cargue los elementos.

controller.loadData(catalog)

Se llama a controller

catalog = model.newCatalog()

controller llama a model para realizar la operación que pide view.

4. Se usa la forma nombredelalista = lt.newList(datastructure, cmpfunction, key, filename, delimiter) donde lt es la función list de la librería del curso.

5. Compara elementos de la lista, si no se da, se debe dar Key.

6. AddLast agrega un elemento al final de la lista.

7. getElement retorna el elemento de una lista dada, dada también su posición.

8. Crea una nueva lista que contiene elementos de lst a partir de un elemento en una posición.

9. El tiempo de ejecución del proyecto es mucho menor al usar single linked en comparación a array list.