

Observaciones del laboratorio 3

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 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 función 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"?

Observaciones:

1. Los mecanismos de interacción que tiene el view.py son los siguientes:
 - Inputs por medio de consola: tiene la funcion de mostrar un menu de opciones en enumeradas, de las cuales el usuario elije dando como input a la consola un string de numero.
 - Outputs por medio de consola: tiene la funcion de enviar un mensaje de la opcion requerida por el usuario con un informe que se imprime en la consola.
2. model.py lee los archivos tipo .csv por medio de las herramientas disponibles en la carpeta DISClib y los almacena en array lists.

```
catalog['books'] = lt.newList()
catalog['authors'] = lt.newList('ARRAY_LIST',
                                cmpfunction=compareauthors)
catalog['tags'] = lt.newList('ARRAY_LIST',
                             cmpfunction=comparetagnames)
catalog['book_tags'] = lt.newList('ARRAY_LIST')
```

3. Las funciones que comunican son aquellas presentes en el controller.py, esas son las siguientes:

- *initCatalog()*
 - *loadBooks(catalog)*
 - *loadTags(catalog)*
 - *loadBooksTags(catalog)*
 - *sortBooks(catalog)*
 - *getBooksByAuthor(catalog, authorname)*
 - *getBestBooks(catalog, number)*
 - *countBooksByTag(catalog, tag)*
4. ¿Cómo se crea una lista?
- Inicialmente, importamos de la librería “DISClib.ADT” el comando `list`, luego de ello, ejecutamos el comando “`list.newlist()`”.
5. ¿Qué hace el parámetro `cmpfunction=None` en la función `newList()`?
- La función `newList` en el archivo `list.py` tiene como parámetro `cmpfunction=None`, lo cual significa que si el valor no es proporcionado a la hora de hacer el llamado, entonces se debe proporcionar el parámetro `Key` y se invocará la función `defaultfunction` que se encuentra en los archivos `singlelinkedlist.py` y `arraylist.py`. Posteriormente `defaultfunction` hará una comparación de tamaño entre `id1` y `id2`, retornando `1` si `id1` es mayor y `-1` si es viceversa, en el caso de que sean iguales se retorna `0`.
6. ¿Qué hace la función `addLast()`?
- Se adiciona un elemento en la última posición de la lista y se actualiza el apuntador a la última posición.
7. ¿Qué hace la función `getElement()`?
- La función `getElement()` recorre la lista proporcionada como argumento y encontrará el objeto en la posición proporcionada, para hacer este recorrido la función invoca a las funciones presentes en la carpeta `Datastructures` en los archivos `singlelinkedlist.py` y `arraylist.py`, dependiendo si el parámetro `lst['tipo'] == 'ARRAY_LIST'`, else implica `singlelinkedlist.py`.
8. ¿Qué hace la función `subList()`?
- Se retorna una lista que contiene los elementos a partir de una posición determinada, con una longitud de elementos dada. Se crea una copia de dichos elementos y se retorna una lista nueva.

Ronald Yesid Diaz Pardo | 202111309

Edgar Luis Vanegas Roa | 201921269

9. ¿Observó algún cambio en el comportamiento del programa al cambiar la implementación del parámetro “ARRAY_LIST” a “SINGLE_LINKED”?

Una vez probado el cambio, no se observa un cambio sustancial en el funcionamiento del programa o algún error a la hora de correr el programa, probablemente el cambio de datastructure sólo implique un cambio de complejidad para ciertas tareas, donde será más rápido array_list o Single_linke dependiendo de esta.