

## OBSERVACIONES DE LA PRACTICA

Karen Quevedo 1 Cod 2023213858

Santiago Perico 2 Cod 202313958

Felipe Gonzalez 3 Cod 202316625

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

El mecanismo O (Output) es el responsable de abrir el menú de opciones que debe elegir el usuario, sabiendo esto, podemos observar que el I (Input) es el numero que toca poner para elegir la opción.

- 2) ¿Cómo se almacenan los datos de **GoodReads** en el **model.py**?

El código encargado es la función, newCatalog, se encarga de crear diccionarios y al recoger el archivo va agregando elementos a este mismo

- 3) ¿Cuáles son las funciones que comunican el **view.py** y el **model.py**?

El view y el model no se comunican directamente, pero se comunican por medio del controller, por ende las funciones con prefijo controller comunican al view con el controller y las de prefijo model comunican el controller con el model.

- 4) ¿Cuál es la función que permite crear una lista?, ¿Qué datos son necesarios?

La función que es newList() y el tipo de estructura de datos que se desea crear, esta en el parámetro "datastructure", junto a "cmpfunction".

- 5) ¿Para qué sirve el parámetro **datastructure** en la función **newList()**?, ¿Cuáles son los posibles valores para este parámetro?

Indica que tipo de estructura se desea para guardar los datos y crear la lista. Se usa, 'Array\_list' y 'Single\_linked', representado la lista en un arreglo y una lista enlazada simple.

- 6) ¿Para qué sirve el parámetro **cmpfunction** en la función **newList()**?

Se utiliza para proporcionar una función de comparación personalizada que se utiliza para ordenar los elementos en la lista. Esta función es opcional, se utilizan funciones de comparación personalizadas “compareauthors” y “comparetagnames” para ordenar autores y etiquetas.

7) ¿Qué hace la función `addLast()`?

Se utiliza para agregar un elemento al final de la lista especificada. En el laboratorio, esta para agregar libros a la lista de libros, agregar libros a la lista de libros asociados con un autor y agregar etiquetas a la lista de etiquetas.

8) ¿Qué hace la función `getElement()`?

Se utiliza para obtener el elemento en la posición especificada de la lista. Por ejemplo, para obtener autores, libros y etiquetas en diferentes partes del código.

9) ¿Qué hace la función `subList()`?

La función crea una copia de los elementos a partir de la posición ‘pos’ y hasta ‘numelem’ elementos, y luego retorna esta nueva sublista. Es decir te permite obtener una lista dada, cuando tienes la posición inicial y el número de elementos para copiar en la sublista.

10) Revise el uso de la función `iterator()` en las funciones `printAuthorData(author)` y `printBestBooks(books)` en la Vista que aplican a una lista de libros. ¿Qué hace la función `iterator()`?

- `printAuthorData`: Se encarga de recorrer la lista que tiene la información de los libros de un autor específico para luego imprimirlos.
- `printBestBooks`: Recorre la lista de libros, para imprimir la información de los mejores libros, esto se determina con el ranking.

11) ¿Observó algún cambio en el comportamiento del programa al cambiar el valor del parámetro '`datastructure`' en la creación de las listas?

Esto es lo que sale cuando se hace sin el cambio para el numeral 11.

```
4- Libros por género
0- Salir
Seleccione una opción para continuar
1
Cargando información de los archivos ....
Libros cargados: 10000
Géneros cargados: 34252
Asociación de Géneros a Libros cargados: 999912
```

Así se ve la consola después de hacer los cambios.

```
Bienvenido
1- Cargar información en el catálogo
2- Consultar los Top x libros por promedio
3- Consultar los libros de un autor
4- Libros por género
0- Salir
Seleccione una opción para continuar
1
Cargando información de los archivos ....
```

De esto podemos ver la importancia del correcto uso del “ARRAY\_LIST” y “SINGLE\_LINKED”, donde al tener que acceder a los datos dentro de los archivos, array únicamente utiliza el index, en cambio single\_linked tiene que utilizar los nodos hasta llegar a la información, saltando de un lado de la memoria a otro, por eso es que al momento de ubicar de manera inversa los elementos a usar, el tiempo para que el código corra aumenta de manera significativa.