

## OBSERVACIONES DE LA PRÁCTICA

Juan Camilo Colmenares Ortiz - 202011866

Juan Andrés Ospina Sabogal - 202021727

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

El primer input permite al usuario escoger una de las 5 opciones que ofrece el menú. Si el usuario escoge la opción 1, el output que da el programa, son las estadísticas del número de registros cargados de casa archivo. Si el usuario escoge la opción 2 se da como output una pregunta, la cual busca saber cuál es el top libros que quiere saber, aquí el usuario ingresa un input con el número de los top libros que quiere saber. Si el usuario escoge la opción 3, el output que da el programa es una pregunta la cual le pide al usuario un input con el nombre del autor del cual desea obtener la información de sus libros, una vez se ingresa como input el autor, el programa da como output los diferentes libros que ha publicado el autor. Si el usuario escoge la opción 4, el output que da el programa es una pregunta respecto al género del cual el usuario desea saber los libros que hay, una vez el usuario da como input el género, el programa responde con la cantidad de libros que se encontraron. Por último si el usuario escoge la opción 0, el programa finaliza.

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

Los datos de GoodReads se almacenan por medio de la función `newCatalog()` la cual crea una lista vacía para guardar todos los libros, también crea una lista vacía para los autores y una lista vacía para los géneros y una lista para la asociación géneros y libros. Como resultado final la función retorna el catálogo inicializado con toda la información de GoodReads.

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

Las funciones que comunican el `view.py` con el `model.py` son `initCatalog()` y `loadData()`.

4) ¿Cómo se crea una lista?

Las listas se crean con la función `newList()`

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

Este parámetro permite comparar los elementos de una lista.

6) ¿Qué hace la función **addLast()**?

Agrega un elemento en la última posición de la lista.

7) ¿Qué hace la función **getElement()**?

Esta función retorna el elemento en la posición pos(este es un argumento que se le da a la función) de la lista.

8) ¿Qué hace la función **subList()**?

Esta retorna una sublista de la lista con una longitud en base a los parámetros que ingresemos en pos y en numelem

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

Lo que notamos al realizar los cambios fue casi intangible en cuanto al tiempo de ejecución entre ambas estructuras de datos. Por un lado, ambos tardaron aproximadamente 8 minutos en cargar los datos y el resto de opciones que ofrece el menú fueron instantáneas a la hora de dar las respuestas.