

# OBSERVACIONES DEL LA PRACTICA

Estudiante 1 Cod 202020727

Estudiante 2 Cod 201821331

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

El view le pide información al usuario de dos maneras, al ejecutar la aplicación le pide un valor numérico para que el programa ejecute la opción que desea el usuario y luego dependiendo de la opción seleccionada se piden datos para las diferentes opciones. Al seleccionar la opción 2 le pide por medio de un input un valor numérico al usuario para buscar en el top que se indique por el input. Para la opción 3 se hace un trabajo análogo con el nombre del autor para consultar sus libros y para la opción 4 se pide por input al usuario la etiqueta a buscar. En cuanto al output, el view.py le muestra al usuario dos cosas; las opciones del menú cada vez que se corre la aplicación y los resultados asociados a cada función, una vez se corre el programa. Es decir en la opción 1 una vez se corra con éxito el view le muestra al usuario el numero de libros, autores, géneros y tags cargados por medio de un print. Similarmente el view imprime los resultados de las otras funciones una vez se recibieron los inputs necesarios del usuario. Una vez imprime los resultados de la función realizada, el view imprime nuevamente las opciones del programa (al menos que el usuario haya seleccionado 0) y espera al input del usuario para realizar la opción seleccionada.

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

Primero se crea el catalogo con las etiquetas de libros, autores, tags y booktags y su correspondiente espacio vacío, luego se crean una single linked list para los libros y 3 array list, para los autores, tags y booktags (estas listas son vacías). Para agregar los libros, se agrega en la última posición de la lista el libro y se agrega el autor del libro y crea el libro en la lista del autor el cual es el apuntador del libro. En cuanto a los tags y booktags estos se agregan a sus listas respectivas. Para cada autor se tiene su nombre, sus libros y su rating promedio. Los tags se almacenan con su nombre y su id. Los book tags se almacenan con su id del libro y del tag.

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

Initcatalog llama a la función del model que crea el catalog

Loadbooks llama a la función del model que agrega un libro al catalog y esto se hace de manera anloga con loadtags y loadBookTags.

Sortbooks ordena los libros por su rating en base a la función sortbooks del model, de manera análoga se hace con countBooksByTags getBestBooks y getBooksByAuthor.

Todas estas funciones son las que llama luego el view.py para realizar sus funciones y las llama desde el controller es decir todas las funciones mencionadas previamente que se encuentran en el controller son las que comunican el view y el model pues el controller llama a las funciones del model para su implementación y trabajo y luego el view las llama a medida que el usuario las solicite para su interacción.

- 4) ¿Cómo se crea una lista?
  - Se define una nueva función llamada “newlist” con parámetros datastructure, cmpfunction, key, filename, delimiter. Esta función retorna la nueva lista vacía.
- 5) ¿Qué hace el parámetro **cmpfunction=None** en la función **newList()**?
  - El parámetro cmpfunction se usa para comparar elementos de la lista creada. En este caso, al ser igual a None, se utiliza la función por defecto y se debe dar un valor a Key.
- 6) ¿Qué hace la función **addLast()**?
  - Esta función agrega un elemento a la última posición de la lista y actualiza el apuntador a esta nueva última posición. Además, incrementa el tamaño de la lista en 1.
- 7) ¿Qué hace la función **getElement()**?
  - Retorna el elemento especificado por “pos” de la lista. Para hacer esto, recorre la lista hasta el elemento “pos” (que debe ser mayor a 0 y menor o igual al tamaño de la lista) y lo retorna sin eliminarlo.
- 8) ¿Qué hace la función **subList()**?
  - Crea y retorna una sub lista de la lista original “lst”. La lista se crea copiando los elementos empezando por el elemento “pos” y con tamaño de “numelem” elementos.
- 9) ¿Observó algún cambio en el comportamiento del programa al cambiar la implementación del parámetro **“ARRAY\_LIST”** a **“SINGLE\_LINKED”**?
  - Cuando corrimos el programa con el parámetro “SINGLE\_LINKED”, notamos una disminución en el tiempo que tomaba el programa en cargar la información en el catálogo (la opción 1 en el menú).