OBSERVACIONES DEL LA PRACTICA

Javier Garcia - jr.garciab@uniandes.edu.co - 201922294

Daniel Lozano - j.lozanom@uniandes.edu.co – 201911107

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

**RTA/** El menú del usuario, porque le permite al usuario poder digitar los numeros que coincidan con las opciones que esté buscando (input) y de esa manera la terminal le enseña al usuario los valores encontrados de esas busquedas (output).

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

**RTA/** Se almacenan mediante un catálogo de libros, con una lista para los libros, una para los autores y otra para los géneros.

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

**RTA/** Ninguna, porque la vista no tiene contacto directo con el modelo ni viceversa, el controlador hace de intermediario entre los dos.

1. ¿Cómo se crea una lista?

**RTA/** Primero se crea una lista vacía, con el parámetro que no dice qué tipo de estructura será usada, un parámetro con función para comparar los elementos de la lista, una parámetro que funciona como indentificador para comparar dos elementos de la lista, otro parámetro que si es dado se crea una lista de los datos encontrados en el archivo y un delimitador para el parámetro anterior para separar los campos con una “,”.

1. ¿Qué hace el parámetro **cmpfunction=None** en la función **znewList ()**?

**RTA/** Realiza la comparación de los elementos de la lista, si no se provee la función de comparación se utiliza la función por defecto pero se debe tener un valor para key, pero si sí se provee el valor de key debe ser None.

1. ¿Qué hace la función **addLast()**?

**RTA/** Agrega un elemento a la última posición de la lista y se actualiza el apuntador a la última posición, el tamaño de la lista se aumenta en 1.

1. ¿Qué hace la función **getElement()**?

**RTA/** Se recorre la lista hasta el elemento dado por el parámetro, el cual debe de ser mayor a cero y menor o igual al tamaño de la lista, se retorna el elemento en dicha posición sin que sea eliminado. La lista no puede estar vacía.

1. ¿Qué hace la función **subList()**?

**RTA/** Retorna una lista que contiene los elementos a partir de la posición dada en el parámetro, con una longitud de elementos dada. Se crea una copia de dichos elementos y se retornan en una lista nueva.

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

**RTA/** Podemos notar que la busqueda usando **ARRAY\_LIST** es más rápida en comparación a cuando usamos **SINGLE\_LINKED**, lo que va de la mano con lo visto en las clases teóricas en donde entendimos que los procesos de busqueda en an un array es de O(1) y en una single es de O(n)