OBSERVACIONES DEL LA PRACTICA

Marilyn Stephany Joven Fonseca 202021346

Samuel Alejandro Jiménez Ramírez 202116652

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

Los mecanismos de interacción con el usuario de forma input que encontramos estaban enfocados a: primero, obtener lo que el usuario desea realizar con el programa en general como, por ejemplo: cargar el archivo, consultar libros, libros por géneros, entre otros; segundo, una vez que el usuario haya elegido la acción que desea realizar en los inputs anteriores se puede requerir otro tipo de información según la necesidad del programa, tal como nombres, géneros, cantidades. Por otro lado, los mecanismos de interacción de forma output podrían delimitarse a las indicaciones y posibles salidas que tiene el usuario para interactuar con el programa, o las respuestas que este le da al usuario.

Cabe resaltar que ambos mecanismos de interacción se despliegan desde la ejecución del apartado de menú principal.

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

Este proceso se puede dividir en varias partes:

* Primero, se crea el modelo de listas para guardar la información en categorías como autor, género, etiquetas y demás información de los libros.
* Luego, se va guardando la información del libro en diferentes categorías como libro, autor, tag, y tag del libro.
* El guardado de datos mas complejo se realiza en las ramas de autores, puesto que el modelo compara todos los autores para que no se repitan, y si no se repiten, se almacena el autor y los libros que hacen referencia a este. Del mismo modo, los libros también apuntan al autor.
* Para tags, y book tags, el guardado se hace libro a libro, primero se hace una lista de tags en general, y por esta misma se clasifican los libros según los tags que tengan

.

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

Estos dos apartados no tienen una comunicación directa y esto se debe al modelo que tiene el programa (MVC), se comunican mediante el controller que es el punto intermedio entre la ruta del controlador y el modelo. De manera general el modelo interactúa completamente con el view, ya que este necesita de las funciones de model para funcionar y almacenar.

1. **¿Cómo se crea una lista?**

Con la función newList(), y se piden cinco parámetros como datastructure que nos define el tipo de lista, cmpfunction del que hablaremos mas adelante, key que es el identificador para hacer comparaciones por defecto, filename que en caso de ser usada crea una lista con los elementos encontrados en un csv y por último delimiter que se utiliza para separar los elementos de la lista.

La lista se crea según el tipo de lista primordialmente, ya que esta va a definir en qué estructura primitiva de python se va a almacenar, como por ejemplo, una arraylist se guarda en un diccionario.

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

Es un parámetro para comparar el tipo de dato que se requiera en una lista, puede ser el nombre del autor, el tag del libro, entre otros. Para este parámetro se retornará un valor de 0 si este es encontrado, o de -1 si no. Cuando el parámetro tiene como valor none se ejecutará con defaultfunction. Cabe aclarar que los parámetros internos que este parámetro recibe son funciones.

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

En general en esta función se agrega un elemento a la última posición de una lista, y a esta lista se le aumenta el valor de su tamaño. Pero la complejidad realmente está cuando el tipo de lista es ***SINGLELINKEDLIST***  ya que trabaja con nodos o apuntadores que dirige el anterior elemento guardado a este nuevo elemento para conectarlos entre si y guardar su ubicación.

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

Esta función tiene como parámetros la lista a evaluar y la posición deseada en esta lista, y nos retorna el elemento que se encuentre en dicha posición.

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

Esta función tiene como parámetros la lista a evaluar, la posición desde la que se va a partir y el número de elementos desados en la nueva lista. Y como podemos ver, en esta función se crea una sublista a partir de una posición deseada de una lista existente.

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

Para cuantificar el comportamiento implementamos un contador de tiempo que nos arrojara el lapso de tiempo en el que se implementaban dichos procesos, estos contadores de tiempo fueron ubicados en distintos puntos del modelo MVC, que tenían que ver con el cambio realizado.

Otra manera de comparar el comportamiento del programa era con los outputs del programa que se diferenciaban entre uno y otro tipo de lista.

Pudimos ver en las librerias que la estructura de la array list era menos compleja que la estructura de single linked.

Aunque, a pesar de que en teoría y para el tipo de datos que estamos utilizando debería ser mas eficiente la array list, nuestros resultados tuvieron desempeños similares o identicos.