

OBSERVACIONES DEL LA PRACTICA

Estudiante 1: Samuel Josue Freire Tarazona Cod 202111460

Estudiante 2: Jose Martinez Cod 202116677

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

Los únicos inputs que tiene son el de menú de opciones, donde el usuario puede digitar números y/o texto de acuerdo a las opciones que se presentan, las cuales hacen acciones diferentes. Los siguientes tres imágenes son un ejemplo de como: en primer lugar el usuario puede digitar números para la opción, en segundo lugar digita el ranking quiere buscar y por último puede digitar texto con el nombre del autor que quiere buscar.

```
if int(inputs[0]) == 1:
```

En este primer input y como los de opciones se le pide al usuario que seleccione cierto entero, el cual le va a permitir al programa realizar ciertas funciones, por ejemplo, si selecciona la primera opción, le permite al usuario cargar toda la información del archivo.

En La segunda opción despliega otro input el cual se ve en la siguiente imagen:

```
number = input("Buscando los TOP ?: ")
```

El cuál es el entero que el usuario ingresa para saber que posiciones quiere saber.

Por otro lado, en la opción tres se le despliegan al usuario otro input más el cual es de texto, y le va a permitir al usuario ingresar el nombre del autor, que el programa utilizara para desplegar su información.

```
authorname = input("Nombre del autor a buscar: ")
```

Por ultimo encontramos el input 4 el cual le va a permitir al usuario desplegar etiquetas del libro y desplegar su información.

```
label = input("Etiqueta a buscar: ")
```

Cuando el autor selecciona la 4 opción como un entero puede ingresar la parte superior que es otro input lo donde el usuario puede escribir el nombre del texto.

Hablando de los outputs o la información que se le muestra al usuario, en primer lugar se despliega el menú de opciones:

Donde el usuario puede visualizar las diferentes operaciones que puede realizar y sus descripciones.

```

C:\Users\megarex-mastercraft\Desktop>python.exe C:/Users/megarex-mastercraft/Desktop/
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

```

Cada uno de estas opciones están puestas como print, o mensaje para el usuario.

En primer lugar cuando el usuario despliega la opción 1 como output le sale la siguiente imagen

```

Libros cargados: 149
Autores cargados: 156
Géneros cargados: 34252
Asociación de Géneros a Libros cargados: 999
Bienvenido

```

Cada uno de estos mensajes representa las diferentes funcionalidades de tamaño de cada archivo, los que se le enuncian al usuario en el output de opciones.

Por otro lado, encontramos el output opciones dos que él sirve al usuario para guiarse en que función puede realizar esta opción. Cuando el usuario despliega la opción 2, se va a encontrar con otro mensaje es decir, con el mensaje que le permite ingresar el entero. Luego de que el usuario digite el nombre que quiere buscar le va saltar el siguiente mensaje al usuario:

```

Buscando los TOP 5: 5
Estos son los mejores libros:
Titulo: Harry Potter and the Deathly Hallows (Harry Potter, #7) ISBN: 545010225 Rating: 4.61
Titulo: A Storm of Swords (A Song of Ice and Fire, #3) ISBN: 055357342X Rating: 4.54
Titulo: Harry Potter and the Half-Blood Prince (Harry Potter, #6) ISBN: 439785960 Rating: 4.54
Titulo: Harry Potter and the Goblet of Fire (Harry Potter, #4) ISBN: 439139600 Rating: 4.53
Titulo: Harry Potter and the Prisoner of Azkaban (Harry Potter, #3) ISBN: 043965548X Rating: 4.53
Bienvenido

```

Este output le muestra al usuario lo que se esperaba de la aplicación con el ranking de los libros, lo importante es que otro medio de interacción de muestra de información.

Por la misma línea cuando el usuario ingresa la opción tres y le sale otro input para agregar el nombre del autor a buscar, resulta en el output como el siguiente:

```

Seleccione una opción para continuar
3
Nombre del autor a buscar: J.K. Rowling
Autor encontrado: J.K. Rowling
Promedio: 0
Total de libros: 7
Titulo: Harry Potter and the Sorcerer's Stone (Harry Potter, #1) ISBN: 439554934
Titulo: Harry Potter and the Prisoner of Azkaban (Harry Potter, #3) ISBN: 043965548X
Titulo: Harry Potter and the Order of the Phoenix (Harry Potter, #5) ISBN: 439358078
Titulo: Harry Potter and the Chamber of Secrets (Harry Potter, #2) ISBN: 439064864
Titulo: Harry Potter and the Goblet of Fire (Harry Potter, #4) ISBN: 439139600
Titulo: Harry Potter and the Deathly Hallows (Harry Potter, #7) ISBN: 545010225
Titulo: Harry Potter and the Half-Blood Prince (Harry Potter, #6) ISBN: 439785960

```

El cual le muestra al usuario en pantalla la información de cierto autor que el selecciono, de manera ordenada este es otro tipo de output que se muestra en el programa.

A grandes rasgos los mecanismo de input principales del programa son los de ingresar enteros texto, cada uno en los casos que ya se mostraron antes. Donde los enteros principalmente son los del menú de opciones que le permiten al usuario desplegar cierta información u otros inputs como los de texto, los cuales le permiten al usuario ingresar información específica como el tag, el nombre de un autor que desea buscar.

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

En primer lugar, hay que definir las estructuras usadas durante la creación del modelo para almacenar los datos. Al principio encontramos que el catalogo esta creado a basa de un diccionario con sus respectivas llaves: books, authors, tags, booktags. Donde a su vez cada una de estas llaves es una estructura por separado en su mayoría cada una de estas llaves es un 'ARRAY_LIST', esto es igual a decir que es un arreglo es decir que los datos están consecutivos. Además, de esto en las llaves Authors y Tags, al crear la lista ordenada se le pide que además compare los autores y compare los tagsnmaes respectivamente. Luego de haber analizado las estructuras vamos a analizar como ose guarda la información y que estructuras se usan o se crean. En primer lugar cuando se quiere agregar un nuevo autor, se crea a base de un diccionario el cual cuenta con las llaves: name, book y avreage_ratings; la primera guarda el nombre y la segunda crea una nueva estructura de datos interna tipo 'Array_List' la cual es una nueva lista enlazada para guardar información de libros. Luego cuando se quiere agregar un nuevo tag, lo que se hace es generar un nuevo diccionario con ciertas variables donde se van a guardar la información del nombre y el id. Y por último, para crear un nuevo booktag es otro diccionario con información del tag y el book_id, cada una de estas están hechas a base de diccionarios que cuentan algunos con nuevas estructuras 'ARRAY_LIST' donde se almacén datos. Luego cuando ya se tiene creadas estas estructuras de datos para almacenar datos, se pueden agregar otras cosas también. Por ejemplo cuando se quieren agregar nuevos bookstags lo primero que se hace es crear el book tag con la estructura y se añade al final de la estructura que ya teníamos con la función addlast; pero eso es tema para luego. Caso similar sucede cuando se quiere agregar un nuevo Tag. Sin embargo, cuando se crea registrar un nuevo autor es diferentes, en primer lugar se tiene que verificar si el autor ya está si es hace se retoman los datos de este y se añaden al final, y cuando no esta se crea el autor y se añade al final. Todo esto manejado por medio de la estructura de datos 'ARRAYList' que permite agregar archivos sin necesidad de tener el orden consecutivo. Por ultimo está la posibilidad de añadir libros la cual añade el libro al final y se añade su referencia al autor.

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

Las funciones que se comunican el view y el model son en el modulo controller.py con las funciones de:

initCatalog: Se llama la función de iniciación del catálogo del modelo.

“load”s: Las funciones loads se encargan principalmente para cargar los archivos de los .csv a las funciones del model como lo son los datos de la estructura de datos, los libros del archivo (con sus respectivos autores y al libro que se procesa), cargar los tags del archivo, además se agregan a la lista de tags y cargar la información que asocia tags con libros.

sortBooks: Ordena los libros por average_rating (El promedio del rating solicitado).

getBooksByAuthor: Retorna los libros de un autor.

getBestBooks: Retorna los mejores libros solicitados por el usuario.

countBooksByTag: Retorna los libros que fueron etiquetados con el tag dado por el usuario.

Qué son las que se encargan de la comunicación de datos del “catalog” que es la solicitud dada del usuario transformada en los datos requeridos y simplificada por este mismo.

4) ¿Cómo se crea una lista?

Para crear una lista sin importar del modelo en el Tad, toca recurrir a la función newList()

```
def newList(datastructure='SINGLE_LINKED',
            cmpfunction=None,
            key=None,
            filename=None,
            delimiter=","):
    """Crea una lista vacia
```

Esta función permite al usuario generar el tipo de Tad lista que hay: 'SINGLE_LINKED' y 'ARRAY_LIST', sin importar cuál de estos tipos quiera crear se usa la misma función como se muestra en la imagen esta función cuenta con varios parámetros que le permiten crear la lista a su antojo. En primer lugar encontramos el parámetro dataestructure, aquí es donde el usuario puede identificar el tipo que quiere crear como ya se dijo anteriormente estas dos 'ARRAY_LIST' y 'SINGLE_LINKED', donde representan la lista ordenada y la lista enlazada respectivamente. Luego, está el parámetro cmpfunction, que sirve como su nombre lo dice para comparar, aquí se puede comparar alguna condición que se quiera evaluar en los elementos. Siguiendo nos encontramos con el parámetro key, el cual sirve para comparar dos elementos de acuerdo a las funciones por defecto. Luego encontramos el parámetro Filename, el cual puede ser el nombre de un archivo, y de ahí se puede crear archivos de datos. Al final, encontramos el parámetro delimiter que puede servir para poder dividir todos datos de los archivos anteriores. Como por ejemplo, los archivos CSV, el delimiter son las comas. Para crear una lista solo se tiene que especificar todos los parámetros según lo que se quiera, y cada

uno de estos son los tipos de TAD. Lo que esta función va a retornar es la lista creada dependiendo del tipo que se halla elegido con cada uno de los parámetros anteriores. A continuación se verá que paso a cuando uno de los parámetros específicamente la cmpfunction no lo ponemos o ponemos None. Sin olvidar que esto se tiene que asignar a una variable como aparece en la siguiente imagen.

```
catalog['books'] = lt.newList()
catalog['authors'] = lt.newList('ARRAY_LIST',
                                cmpfunction=compareauthors)
catalog['tags'] = lt.newList('ARRAY_LIST',
                             cmpfunction=comparetagnames)
catalog['book_tags'] = lt.newList('ARRAY_LIST')
```

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

Lo que pasa es que va a utilizar la función por defecto, es decir, se va utilizar una función predefinida que va realizar cierta comparación, en la siguiente imagen podemos encontrar la función por defecto:

```
def defaultfunction(id1, id2):
    if id1 > id2:
        return 1
    elif id1 < id2:
        return -1
    return 0
```

Esta función como podemos ver compara los dos elementos de una lista y los compara y dice si el primer elemento es mayor o menor que el segundo según el entero que nos resulte. También cabe aclarar que al hacer que la comparación sea None se tiene que poner un valor en Key, para hacer la comparativa con la función estándar. Lo que hace esta función es organizar los elementos teniendo en cuenta la función proporcionada en este caso, lo haría por el orden accedente.

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

Esta función recibe dos parámetros, en primer lugar recibe el nombre de la estructura de datos donde quiere que se realice el cambio, y el segundo es el dato que quiere guardar. Como ya se dijo la función de esta opción es agregar. Sin embargo, no agrega en cualquier posición en la estructura de datos, sino que agrega al final, lo que genera que la tamaño de la estructura cambie y se añada un dato más. Para complementar, cabe aclarar que el último elemento de la lista, tiene como valor siguiente None, porque no tiene ningún vinculado luego de él. El retorno de esta función es la misma lista pero modificada, es decir, no tiene ningún retorno solo modifica la lista a no ser de que se le asigne a alguna variable. También, como ya se dijo al añadirse al final, este queda vinculado a un dato previo pero no a uno posterior.

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

Esta función recibe como parámetro, dos variables la primera la lista que se quiere analizar y la segunda la posición que se desea conocer el valor, este número es un entero. Cuando la función recibe los dos parámetros, lo primero que hace es empezar a recorrer la estructura de datos puesta (Sirve para la lista enlazada y la lista ordenada), ya que no va por orden sino por posición, hasta el punto que se necesita, y luego extrae la información de esa posición. El retorno de esta función es la información encontrada por el algoritmo. El entero o la posición que el usuario desee conocer tienen que claramente estar en el rango de la lista.

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

```
def subList(lst, pos, numelem):  
    """ Retorna una sublista de la
```

Esta función recibe tres parámetros, el primero de ellos es la estructura de datos (lista enlazada y lista ordenada) donde se quiere realizar la sustracción, el segundo 'Pos' el cual nos va a indicar una posición en la lista y por último el numelem nos va a indicar el tamaño de la sublista que queremos crear. Lo que esta función busca es crear una lista más pequeña a partir de un punto dado y una longitud dada, estos son los valores que recibe como parámetros. En primer lugar, se va a ubicar la posición de donde se quiere empezar, luego a partir de ahí se van a contar la cantidad de posiciones de acuerdo al número de elementos que se quiere sea el tamaño de la sublista. Luego todos los elementos que queden en ese subrango de números, se sustraen y se asignan a una nueva lista generada automáticamente. El retorno de esta función es la nueva lista con los elementos seleccionados.

9) ¿Observó algún cambio en el comportamiento del programa al cambiar la implementación del parámetro "ARRAY_LIST" a "SINGLE_LINKED"?

```
Promedio: 0  
Total de libros: 7  
Titulo: Harry Potter and the Sorcerer's Stone (Harry Potter, #1) ISBN: 439554934  
Titulo: Harry Potter and the Prisoner of Azkaban (Harry Potter, #3) ISBN: 043965548X  
Titulo: Harry Potter and the Order of the Phoenix (Harry Potter, #5) ISBN: 439358078  
Titulo: Harry Potter and the Chamber of Secrets (Harry Potter, #2) ISBN: 439064864  
Titulo: Harry Potter and the Goblet of Fire (Harry Potter, #4) ISBN: 439139600  
Titulo: Harry Potter and the Deathly Hallows (Harry Potter, #7) ISBN: 545010225  
Titulo: Harry Potter and the Half-Blood Prince (Harry Potter, #6) ISBN: 439785960  
Bienvenido  
1- Cargar información en el catálogo  
2- Consultar los Top 5 libros por promedio
```

2

Buscando los TOP 5: 5

Estos son los mejores libros:

Titulo: Harry Potter and the Deathly Hallows (Harry Potter, #7) ISBN: 545010225 Rating: 4.61

Titulo: A Storm of Swords (A Song of Ice and Fire, #3) ISBN: 055357342X Rating: 4.54

Titulo: Harry Potter and the Half-Blood Prince (Harry Potter, #6) ISBN: 439785960 Rating: 4.54

Titulo: Harry Potter and the Goblet of Fire (Harry Potter, #4) ISBN: 439139600 Rating: 4.53

Titulo: Harry Potter and the Prisoner of Azkaban (Harry Potter, #3) ISBN: 043965548X Rating: 4.53

Promedio: 0

Total de libros: 7

Titulo: Harry Potter and the Sorcerer's Stone (Harry Potter, #1) ISBN: 439554934

Titulo: Harry Potter and the Prisoner of Azkaban (Harry Potter, #3) ISBN: 043965548X

Titulo: Harry Potter and the Order of the Phoenix (Harry Potter, #5) ISBN: 439358078

Titulo: Harry Potter and the Chamber of Secrets (Harry Potter, #2) ISBN: 439064864

Titulo: Harry Potter and the Goblet of Fire (Harry Potter, #4) ISBN: 439139600

Titulo: Harry Potter and the Deathly Hallows (Harry Potter, #7) ISBN: 545010225

Titulo: Harry Potter and the Half-Blood Prince (Harry Potter, #6) ISBN: 439785960

Bienvenido

Buscando los TOP 5: 5

Estos son los mejores libros:

Titulo: Harry Potter and the Deathly Hallows (Harry Potter, #7) ISBN: 545010225 Rating: 4.61

Titulo: A Storm of Swords (A Song of Ice and Fire, #3) ISBN: 055357342X Rating: 4.54

Titulo: Harry Potter and the Half-Blood Prince (Harry Potter, #6) ISBN: 439785960 Rating: 4.54

Titulo: Harry Potter and the Goblet of Fire (Harry Potter, #4) ISBN: 439139600 Rating: 4.53

Titulo: Harry Potter and the Prisoner of Azkaban (Harry Potter, #3) ISBN: 043965548X Rating: 4.53

Bienvenido

Las anteriores imágenes, son cada una con el respectivo tipo de estructura de datos, la primera corresponde a el tipo: 'ARRAY_LIST' y las segundas corresponden al tipo 'SINGLE_LINKED'. Como se puede ver no existe diferencia entre ninguna de las dos estructuras. Se probó en diferentes dispositivos y con los archivos simplificados, ya que los archivos grandes no fue posible abrirlos en ningún de los dos dispositivos. Además, la cargar demoraba lo mismo mostraba los mismo resultado. En conclusión, al hacer el cambio de una estructura otra, en este caso 'ARRAY_LIST' y 'SINGLE_LINKED', no existe ninguna diferencia de funcionamiento, ni de duración del algoritmo. Esto se puede dar ya que en las operaciones del List sirven para ambos tipo de estructura sin modificación alguna, por otro lado también puede ser porque solo se cambian esas dos estructuras