OBSERVACIONES DEL LA PRACTICA

Valentina Goyeneche Cod 201922380

Sebastian Contreras Cod 202020903

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

Partiendo del código que se encuentra en el archivo view.py se puede observar que los input son aquellos que piden una respuesta del usuario, como el entero inputs que guarda la selección del usuario sobre el menú. Por otro lado los outputs son aquellos que presentan información al usuario como la función printMenu() la cual muestra las opciones del menú al usuario, así como los datos que se entregan al usuario conforme a lo que ingrese en el menú de opciones.

Para inputs:

inputs = input('Seleccione una opción para continuar\n')

Para outputs:

def printMenu():

    print("Bienvenido")

    print("1- Cargar información en el catálogo")

    print("2- Consultar los Top x libros por promedio")

    print("3- Consultar los libros de un autor")

    print("4- Libros por género")

    print("0- Salir")

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

Los datos que se encuentran almacenados en los archivos de Goodreads se cargan en el controller utilizando una librería de csv para luego ser trasladados al modelo y almacenados en una lista.

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

Funciones de controller que son utilizadas en vista

initCatalog()

loadData(catalog)

controller.getBestBooks(catalog, int(number))

controller.getBooksByAuthor(catalog, authorname)

controller.countBooksByTag(catalog, label)

1. ¿Cómo se crea una lista?

Dentro de list.py una lista se crea por medio de la función newList() que recibe por parámetro el tipo de estructura de datos a utilizar, cmpfunction (función de comparación de elementos de la lista), key, filename(si se da este valor se crea la lista a partir de los elementos del archivo): y delimiter(delimitar y separar los campos).

def newList(datastructure='SINGLE\_LINKED',

           cmpfunction=None,

           key=None,

           filename=None,

           delimiter=","):

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

El parámetro cmpfunction realiza la función de comparar los elementos de la lista, si no se le da un valor como en este caso, se hace uso de un comparador por defecto que requiere de que se agregue el valor de key para comparar 2 elementosde la lista por comparación por defecto.

1. ¿Qué hace la funció **addLast()**?

Como se puede observar en los comentarios del código dentro del archivo list.py esta función “agrega un elemento a la ultima posición de la lista y se actualiza el puntador a la ultima posición.”

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

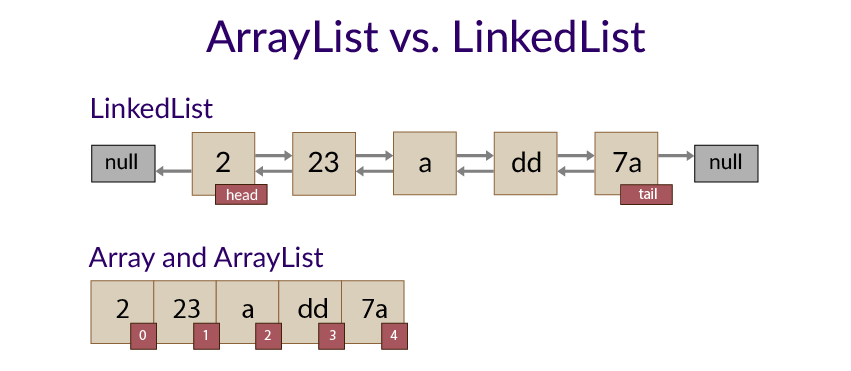
Como se puede observar en los comentarios del código dentro del archivo list.py esta función “Recorre una lista hasta el elemento que entra por parámetro, este elemento debe ser mayor que cero y menor o igual al tamaño de la lista, debe retornar el elemento en la posición que se entrega por parámetro, la lista no debe ser vacía”.

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

Se crea una copia de los elementos que entran por parámetro y se retorna una lista nueva a partir de los parámetros que entran por función.

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

En primer lugar, se observó que la función para cargar los datos dentro del menú tarda menos tiempo en comparación con la situación inicial, con ARRAY\_LIST esta función tardo 448 segundos mientras que con SINGLE\_LINKED tardo 405 segundos; esto ocurre porque el SINGLE\_LINKED usa listas de doble enlace que le permite ser manipulado de forma más rápida ocupando más memoria debido a que cada elemento de la lista posee un apuntador que informa sobre que elemento va antes y después. Por otro lado, observamos que las funciones que extraían elementos de en medio de los datos como la opción 2, 3 y 4 del menú principal tardaban más con el SINGLE\_LINKED que con el ARRAY\_LIST.



Tomada de: [original.jpeg (851×371) (codegym.cc)](https://cdn.codegym.cc/images/article/100c27ce-9ff8-447d-b92d-7feabee15a16/original.jpeg)