OBSERVACIONES DEL LA PRACTICA

Sergio Avendano-201923730

Estudiante 2 Cod XXXX

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

El mecanismo de interaccion input se puede ver en el while ya que en esta parte del codigo se guarda la opcion elegida por el usaurio y dependiendo de esta misma, el programa arrojara lo pedido por el usuario.

Los mecanismos output serian las funciones print como

def printBestBooks(books):

size = lt.size(books)

if size:

print(' Estos son los mejores libros: ')

for book in lt.iterator(books):

print('Titulo: ' + book['title'] + ' ISBN: ' +

book['isbn'] + ' Rating: ' + book['average\_rating'])

else:

print('No se encontraron libros')

o como

elif int(inputs[0]) == 4:

label = input("Etiqueta a buscar: ")

book\_count = controller.countBooksByTag(catalog, label)

print('Se encontraron: ', book\_count, ' Libros')

ya que mediante lineas de codigo como estas el programa arroja los resultados deseados por el usurio.

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

Los datos de GoodReads estan distribudidos y almacenados en listas distintas

catalog = {'books': None,

'authors': None,

'tags': None,

'book\_tags': None}

Y mediante distintas funciones se pueden editar dichas listas.

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

Ninguna funcion comunica el view.py con el model.py, ya que esta conexión se da solamente mediante el controlador como se ve en la siguiente funcion:

Desde una funcion en View.py se llama al controlador

def initCatalog():

"""

Inicializa el catalogo de libros

"""

return controller.initCatalog()

y desde el controlador.py, se llama al model

def initCatalog():

"""

Llama la funcion de inicializacion del catalogo del modelo.

"""

catalog = model.newCatalog()

return catalog

1. ¿Cómo se crea una lista?

def newList(datastructure='SINGLE\_LINKED',

cmpfunction=None,

key=None,

filename=None,

delimiter=","):

"""Crea una lista vacia

Dentro de list.py se crea una lista mediante la funcion “newList”, la cual recibe por parametro la estructura de datos a usar, otro parametro que tiene una funcio de comparacion de los elementos de la lista, un identificador el cual se utiliza para comparar los elementos de la lista con una funcion de comparacion por defecto. Por ultimo, se crea un delimitador el cual ayuda a separar los elementos de la lista.

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

Este parametro ayuda a comparar los elementos de la lista, mediante una funcion de comparacion.

1. ¿Qué hace la funció **addLast()**?

Esta funcion anade un elemento en la ultima posicion de la lista y recibe por parametro la lista a la cual se va a agregar el elemento y el nuevo elemento.

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

Esta funcion recibe por parametro la lista y la posicion del elemento requerido. Por esto que se recorre toda la lista hasta llegar a la posicion dada por parametro la cual debe ser mayor que cero.

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

Esta funcion recibe por parametros la lista, la posicion y la longitud de la lista pequena que se quiere extraer, retorna una copia de los elemento contenidos dede la posicion inicial dada hasta la longitud ingresada

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

No se evidencio un gran cambio ya a debido el tiempo que tomo en cargar los archivos fue bastante en ambos tipos de listas. Para buscar algo en especifico, usando single linked se pudo ver que se demoro un poco mas, pero no es tan evidente la diferencia