OBSERVACIONES DEL LA PRACTICA

Estudiante 1 Cod XXXX

Estudiante 2 Cod XXXX

1. ¿Cuáles son los mecanismos de interacción (I/O: Input/Output) que tiene el **view.py** con el usuario?
   1. La impresión del menu, que le permite al usuario ver la opciones de lo que puede hacerText

      Description automatically generated
   2. Las funciones input() de python, que permiten que el usuario ingrese datos que desea consultar y especifique que funciones desea correr. 
2. ¿Cómo se almacenan los datos de **GoodReads** en el **model.py**?
   1. El catalogo es un diccionario con 4 llaves (‘books’, ‘authors’, ‘tags’, ‘book\_tags’). Los valores de cada una estas llaves son TAD Listas y en este caso se inicializan a ‘ARRAY\_LISTS’. Text

      Description automatically generated
3. ¿Cuáles son las funciones que comunican el el **view.py** y el **model.py**?
   1. El view llama el controller en 6 ocacionesA screenshot of a computer

      Description automatically generated with medium confidence
   2. Luego desde estas funciones del controller se llama al modelo
   3. A partir de estos llamados en el view.py y en el controller.py, respectivamente, el view.py se puede comunicar con el model.py
4. ¿Cómo se crea una lista?
   1. Con la función **newList()** se crea un lista vacia, y luego a esta lista se le van añadiendo elementos con **addFirst(), addLast(), insertElement()**
5. ¿Qué hace el parámetro **cmpfunction=None** en la función **newList()**?
   1. Ese parametro indica cual es la función de comparción para los elementos de una listq. =None, significa que si no se da una función, se usa la función pr defecto pero debe darse un valor para key. Si se provee la función, el valor de key debe ser none.
6. ¿Qué hace la función **addLast()**?
   1. Agrega un elemento en la última posición de la lista y la actualiza el apuntador a última posición. Además, incrementa el tamaño de la lista por 1.
7. ¿Qué hace la función **getElement()**?
   1. Recorre la lista hasta llegar al indice indicado y retorna el elmento en dicha posición.
8. ¿Qué hace la función **subList()**?
   1. Reotorna una copia de lista orginal que va desde una posición indicada hasta el tamaño que se indica.
9. ¿Observó algún cambio en el comportamiento del programa al cambiar la implementación del parámetro **“ARRAY\_LIST”** a **“SINGLE\_LINKED”**?
   1. Cuando usamos **“SINGLE\_LINKED”** el programa se demoro 8min:23s cargando los archivos

Graphical user interface, application

Description automatically generated

* 1. Cuando usamos **“ARRAY\_LIST”** el programa se demoro 11min:31s cargando los archivos

Graphical user interface, application

Description automatically generated

* 1. Claramente el uso del **“SINGLE\_LINKED”** es más eficiente ya que reduce el tiempo que se demora en cargar los archivos por 3 minutos.