OBSERVACIONES DEL LA PRACTICA

Estudiante 1 Cod 201922006

Estudiante 2 Cod 202014143

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

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

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

Inicialmente se construye un modelo el cual va a ser newCatalog, el cual va a contener los libros, autores y tags almacenados en una lista vacía.

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

El proceso inicia con la función newCatalog y todas las funciones load en el model para luego pasar al controler para que sean comunicadas por las funciones initCatalog y loadData, estas serán poteriormente invocadas en el view por las funciones initCatalog y loadData.

1. ¿Cómo se crea una lista?

Texto

Descripción generada automáticamente

En el modelo se crea una nueva lista utilizando el TAD(“ARRAY\_LIST”) obtenido de la carpeta Disclib.

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

El parametro cmpfunction=None tiene como función comparar los elementos que se van a añadir a la lista con un orden establecido.

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

Se encargar de agregar un último elemento al final de la lista

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

Esta función nos permite obtener un elemento de la lista

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

Retorna una lista con elemento que contienen la lista original

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

Con la lista encadenada se demora mucho más que con el arreglo, la lista encadenada nos permite usar menos memoria lo cual sería valioso cuando usemos los datos completos. Dependiendo de la función que se vaya a usar cambia el orden de implementación del Tad.