

OBSERVACIONES DEL LA PRACTICA

Juan Camilo Bonet Cod 202022466

Jesus Jimenez Cod 202020431

1) ¿Cuáles son los mecanismos de interacción (I/O: Input/Output) que tiene el **view.py** con el usuario?
Los mecanismos principales de interacción con el usuario son los `input(...)` y `print(...)`: el primero sirve para obtener datos del usuario y el segundo para enseñar resultados al mismo.

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

3) ¿Cuáles son las funciones que comunican el **view.py** y el **model.py**?
La comunicación entre ambos .py ocurre en el controlador, esto a través de varias funciones en el archivo `controller.py`: `initCatalog()`, `loadData()`, `getBestBooks()`, `getBooksByAuthor()` y `CountBooksByTag()`.

4) ¿Cómo se crea una lista?

Con la función `newList`. Internamente esta función se encarga de decidir qué estructura de lista usar y cada uno de sus valores iniciales a través de sus parámetros.

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

El parámetro `cmpfunction` recibe un criterio de evaluación para la comparación de elementos dentro de una lista.

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

Agrega un elemento en la última posición de la lista. Tal elemento y lista son obtenidas como parámetros de la función.

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

Retorna el elemento en la posición `pos` de la lista. La posición del elemento a conseguir y la lista son obtenidos como parámetro de la función.

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

Retorna una lista extraída de la lista original. Esta función toma como parámetros el número de elementos y cuál es la posición inicial.

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

Al cambiar el tipo de lista se observó un cambio en los tiempos de ejecución, siendo un poco más lento **"SINGLE_LINKED"** al ejecutar las funciones del proyecto.