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

- Al ejecutar el programa, el archivo view.py se encarga de imprimir las diferentes funciones que hay disponibles para que el usuario elija una. Posteriormente, el programa recibe como respuesta el numero asignado a determinada función por medio de la terminal. También, cuando se ejecuta una función, en algunos casos se pide al usuario que ingrese datos adicionales a través de la consola o se imprime información en consola para informar un resultado.

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

- En primer lugar, en el controller.py se utiliza la función DictReader para recopilar toda la información contenida en un archivo csv. Luego se recorre el objeto iterable generado y cada una de sus iteraciones se pasa como parámetro en la función correspondiente del model, la cual se encarga de agregar el objeto en cuestión a la lista.

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

**-** Básicamente, todas las funciones del archivo controller.py ayudan a comunicar el view.py y el model.py entre si. Cuando se ejecuta una función en el view.py, en el caso que se requiera se invoca la función del controller.py encargada de invocar a su vez aquella que se requiera utilizar del model.

**¿Cómo se crea una lista?**

- Se crea utilizando la siguiente estructura: ls.newList(“tipo\_de\_lista”)

En donde:

ls: abreviación para referirse a la clase list del modulo DISClib.ADT

newList(): función para crear una lista nueva

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

- Da a cmpfunction un valor nulo.

**¿Qué hace la función addLast()?**

- Agrega un elemento al final de la lista.

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

- Busca determinado elemento en una lista.

¿Qué hace la función subList()?