

1. Los mecanismos de interacción del programa son:
 - a. Inputs:
 - i. Menu: Recibe una opción por parte del usuario para ejecutar una de las funcionalidades que ofrece.
 - ii. Top libros por promedio: Recibe un puntaje promedio para buscar los libros con este valor.
 - iii. Nombre del autor: Recibe un valor tipo string que corresponde al autor del que se quieren buscar libros.
 - iv. Etiqueta a buscar: Recibe un valor tipo string que corresponde a una etiqueta.
 - b. Outputs:
 - i. Menú: Imprime el menú que muestra las opciones que puede seleccionar el usuario.
 - ii. Cargar información del catálogo: Imprime un mensaje con el número de libros, autores y géneros cargados y la asociación de géneros a libros cargados.
 - iii. Consultar los top x libros por promedio: Imprime un mensaje con los mejores x libros, donde x es el número seleccionado por el usuario
 - iv. Consultar los libros de un autor: Imprime un mensaje con todos los libros en el catálogo escritos por el autor que seleccionó el usuario.
 - v. Libros por género: Imprime un mensaje con el número de libros en un género seleccionado por el usuario.
2. Cada tipo de dato (tags, booktags, books) se almacenan en arrays, cada array contiene información del mismo tipo.
3. Las funciones que comunican el view.py y el model.py se encuentran en controller.py. Estas son:
 - a. LoadData: Contiene Load Books, load Tags, loadBooksTags, sortBooks. Corre estas funciones para cargar los datos cuando lo pide el view.py y los manda al model.py para organizarlos en arrays
 - b. getBooksbyAuthor: Recibe el input del usuario obtenido en view.py y lo manda al model.py para que busque los libros relacionados con el input (string que contiene el nombre del autor).
 - c. GetBestBooks: Es usado por el view.py y llama a la función get Best Books del modelo.
 - d. countBooksByTag: Es usado por el view.py y llama a la función count Books By Tag del modelo.
 - e. En general las funciones de controller.py son usadas por el view.py y llaman una función del model.py según el usuario lo requiere (los inputs del view llaman estas funciones).

1. Las listas se pueden crear usando la función New List del módulo lst, que recibe cómo parámetros data structure (que muestra el tipo de estructura de datos que se usará para implementar la lista), cmpfunction (la función de comparación para los elementos de la lista), key (el identificador utilizado para comparar dos elementos en la lista, que por defecto es la función de comparación), filename (que se usa para cargarle datos a la lista a partir de un documento) y el delimiter (que es el valor que se usa para separar los campos.)

2. El parámetro cmpfunction=None en newList() fija la función de comparación por defecto.
 3. La función AddLast le agrega un elemento al final de la lista, después de la última posición. También aumenta el tamaño de la lista en uno.
 4. La función getElement retorna el elemento en la posición solicitada de la lista sin eliminarlo. La posición debe ser mayor a 0 y menor al tamaño de la lista.
 5. La función subList tiene como parámetros (lst,pos, num elem) y copia los elementos desde la posición pos con longitud num elem, crea una lista nueva con ellos y la retorna.
1. Al cambiar la implementación del parámetro "ARRAY_LIST" a "SINGLE_LINKED" no se observó ningún cambio en el funcionamiento del programa, los tiempos de respuesta eran casi inmediatos en ambos casos y aparecían los mismos resultados. Al final es lo que se espera porque es una estructura alterna que cumple la misma función.