**Paso 3**

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

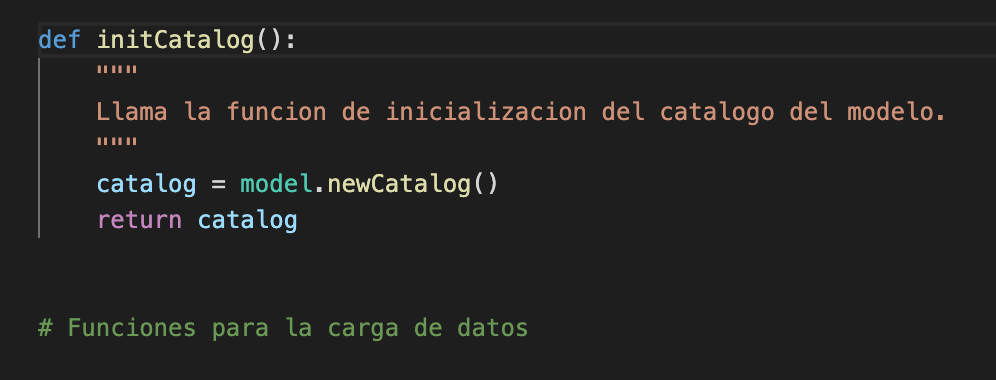
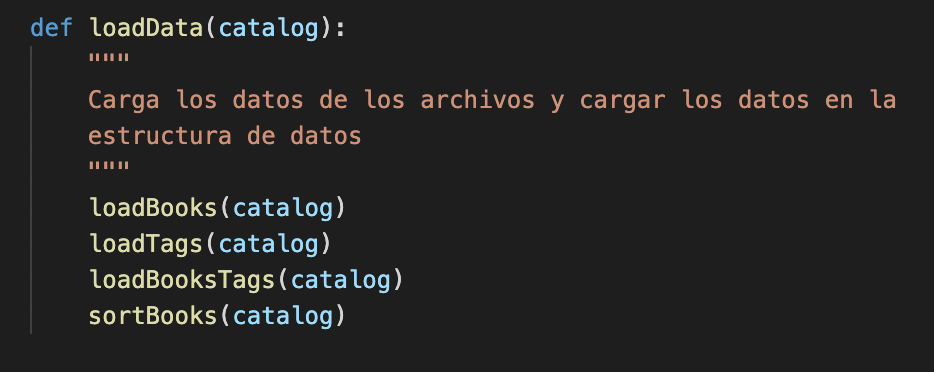
El programa recibe los parámetros a través del view y estos, dependiendo de la opción seleccionada, le va a solicitar a controller.py que ejecute la función indicada en model.py, se encargará de procesar la función y modificará los datos o devolverá información (Dependiendo de las indicaciones dadas por el usuario).

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

NewCatalog crea un diccionario con cuatro llaves listas vacías como valor (books, author, tags y book\_tags, lo envía a view y view llamara a la función de loadData para que cargue los datos del csv. LoadData va a llamar a otras 4 funciones que cargaran libros, autores, tags y book tags respectivamente y van a modificar las listas del diccionario catálogo.

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

Modelo y vista no se conectan directamente, sin embargo, las siguientes funciones les permiten tener conexión mediante el controlador:

* InitCatalog:  
  
* loadData:  
  

Si bien no son funciones, las opciones 2 a 4 en el menú de view también hacen conexión con model

Paso 4

**¿Cómo se crea una lista?**

RTA: Nos manda a una función donde se intentará crear un arreglo o una lista encadenada dependiendo del tipo de lista especificada. Si es un “ARRAY\_LIST” se hará un arreglo, pero si no es igual, se creará una lista encadenada.

Si creo un arreglo, mandará a DataStructures\arraylist.py donde se ejecutará la función newlist(); la cual será aquella que se encargará de hacer la lista al crear un diccionario que tenga por dentro las llaves:  
  
- [“elements”] = Elementos iniciales de la lista  
- [“size”] = Tamaño de la lista  
- [“type”] = Tipo de lista  
- [“key”] = La llave de asignación

Las cuáles serán configuradas por los parámetros dados en la función; por ejemplo, los elementos de la lista serán extraídos de un archivo dado como argumento (filename), y serán agregados al llamar a otras funciones ya definidas. Mientras que cmpfunction, key y delim se encargan de acomodar ciertas preferencias del arreglo.  
  


De lo contrario, mandara a DataStructures\singlelinkedlist.py para que se cree una lista encadenada al ejecutarse newlist(); la cual será aquella que creara la lista al definir un diccionario que tendrá las siguientes llaves:

- ["first”] = Elemento inicial de la lista  
- [“last”] = Elemento final de la lista  
- [“size”] = Tamaño de la lista  
- [“key”] = La llave de asignación  
- [“type”] = Tipo de lista, enlazada en este caso.  
Llaves que inicialmente estarán vacías pero la función newlist() de singletangledlist.py está pasando parámetros para que eso campos sean llenados al llamar otras funciones definidas.   


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

RTA: Determina que se utilizara la función de comparación por defecto.

**¿Qué hace la función addLast()?**

RTA: addLast() se encarga de, dependiendo del tipo de lista utilizada, agregar un elemento al final de un arreglo o una lista encadenada simple.   
Si el tipo de lista es un arreglo, se importará una función del archivo arrayList.py donde solo se agrega un elemento al final y se le suma 1 al conteo de elementos de la lista.  
  
De lo contrario, si es una lista encadenada simple, importara una función del archivo singlelinkedlist.py donde se crea un nuevo nodo y se evalúa si la lista está vacía o no; en caso de que este vacía agregara el nodo como el valor principal, de lo contrario, solo agregara el nodo al final y lo asignara a la llave [“last”].

**¿Qué hace la función getElement()?**  
  
RTA: De la misma manera que en los anteriores, dirige a una función de otro archivo dependiendo del tipo de lista que sea. Esta se encarga de obtener un elemento de la lista, dado su índice.  
Si es un arreglo, solo se iniciará una función donde al índice se le resta 1 y se busca el dato en con ese nuevo índice: list[index-1]

En caso del getElement en una singlelinkedlist, se inicia una función que se encargara de recorrer la función de inicio hasta llegar a la posición dada, donde retornara la información del nodo.

¿Qué hace la función subList()

Sublist se encarga de crear una nueva lista a partir una posición y longitud dada.   
En el arrayList, se implementa una función que crea una lista vacía, con las mismas características de la lista principal, para posteriormente tomar los valores a partir de pos-1 con ayuda de un while.

En la lista encadenada, se utiliza getElement() para traer uno por uno de los elementos a agregar con ayuda de un while.

Paso 5.

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

RTA: Me pareció que el proceso de carga fue más rápido en la lista encadenada, sin embargo, en realidad pareciera que todo funciona de la misma manera pues no vi ningún otro tipo de cambio notable porque el comportamiento fue básicamente el mismo.