OBSERVACIONES DEL LA PRACTICA

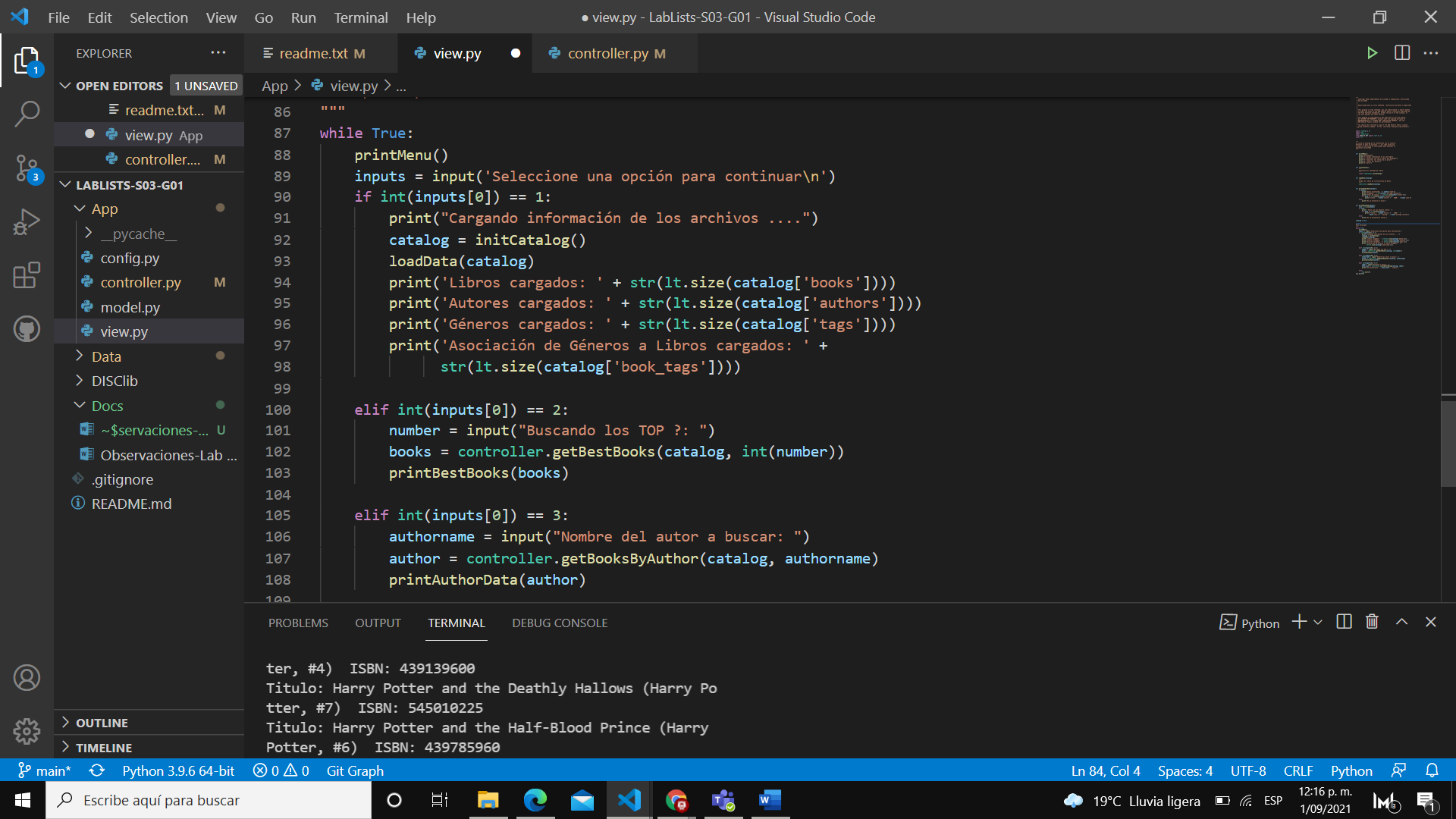
Estudiante 1 – Luis Borbón Holguin – Cod. 202110887

Estudiante 2 – Jaime Alfonso Ruiz – Cod. 202116525

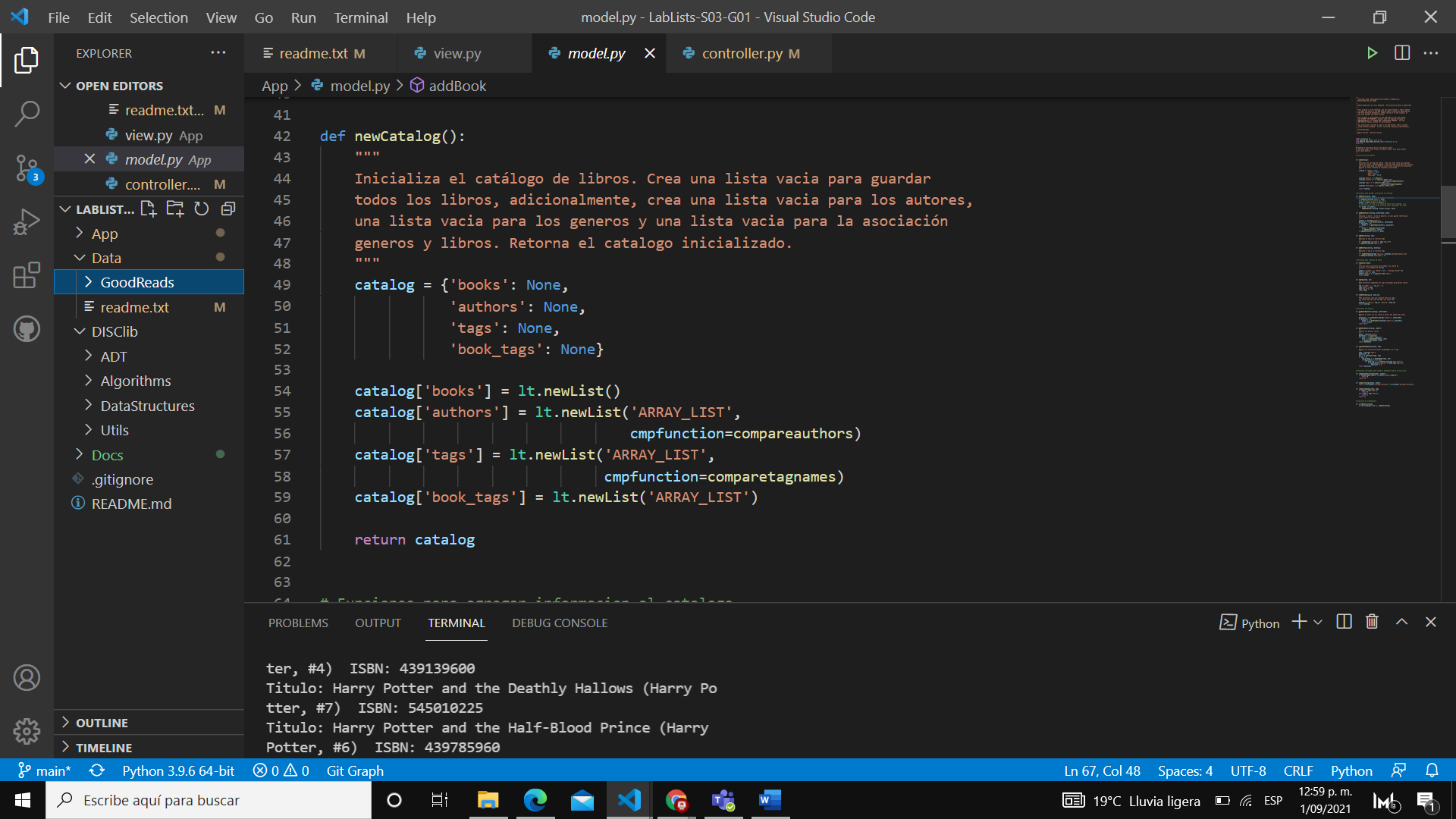
1. ¿Cuáles son los mecanismos de interacción (I/O: Input/Output) que tiene el **view.py** con el usuario?
2. ¿Cómo se almacenan los datos de **GoodReads** en el **model.py**?
3. ¿Cuáles son las funciones que comunican el el **view.py** y el **model.py**?
4. ¿Cómo se crea una lista?
5. ¿Qué hace el parámetro **cmpfunction=None** en la función **newList()**?
6. ¿Qué hace la funció **addLast()**?
7. ¿Qué hace la función **getElement()**?
8. ¿Qué hace la función **subList()**?
9. ¿Observó algún cambio en el comportamiento del programa al cambiar la implementación del parámetro **“ARRAY\_LIST”** a **“SINGLE\_LINKED”**?

**Solución**

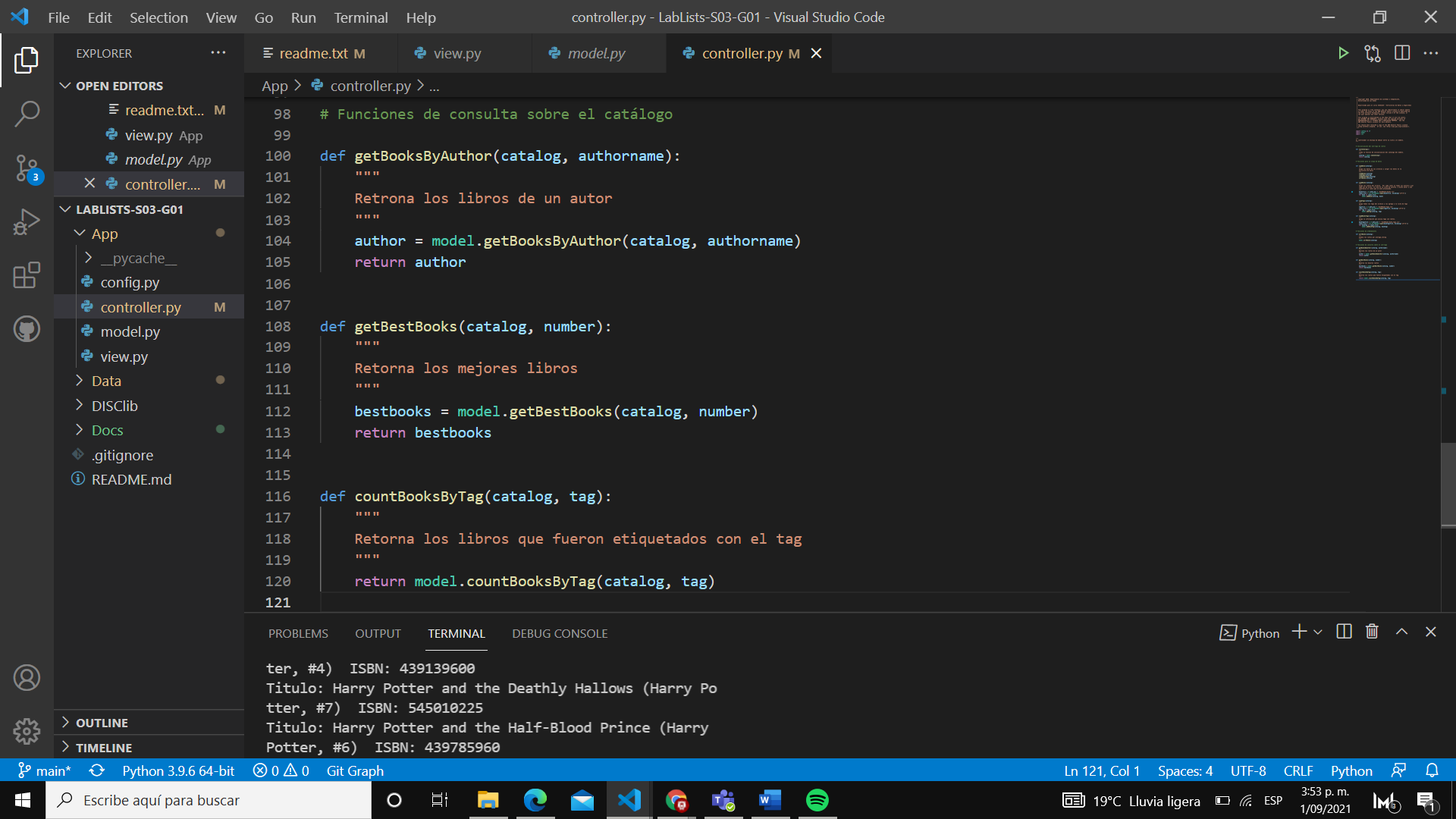
1. El mecanismo de interacción que se presenta principalmente es Input, donde se construye una variable llamada “inputs”, la cual contiene la función input y esta se usa para la interacción y pedirle información al usuario acerca de algo que necesite ingresar el usuario para la búsqueda y ejecución del programa. Por otra parte, se usa como mecanismo de interacción Output el print, el cual se usa para mostrarle al usuario en pantalla los resultados obtenidos por el programa, o la interacción con este para mostrarle un mensaje o menú.

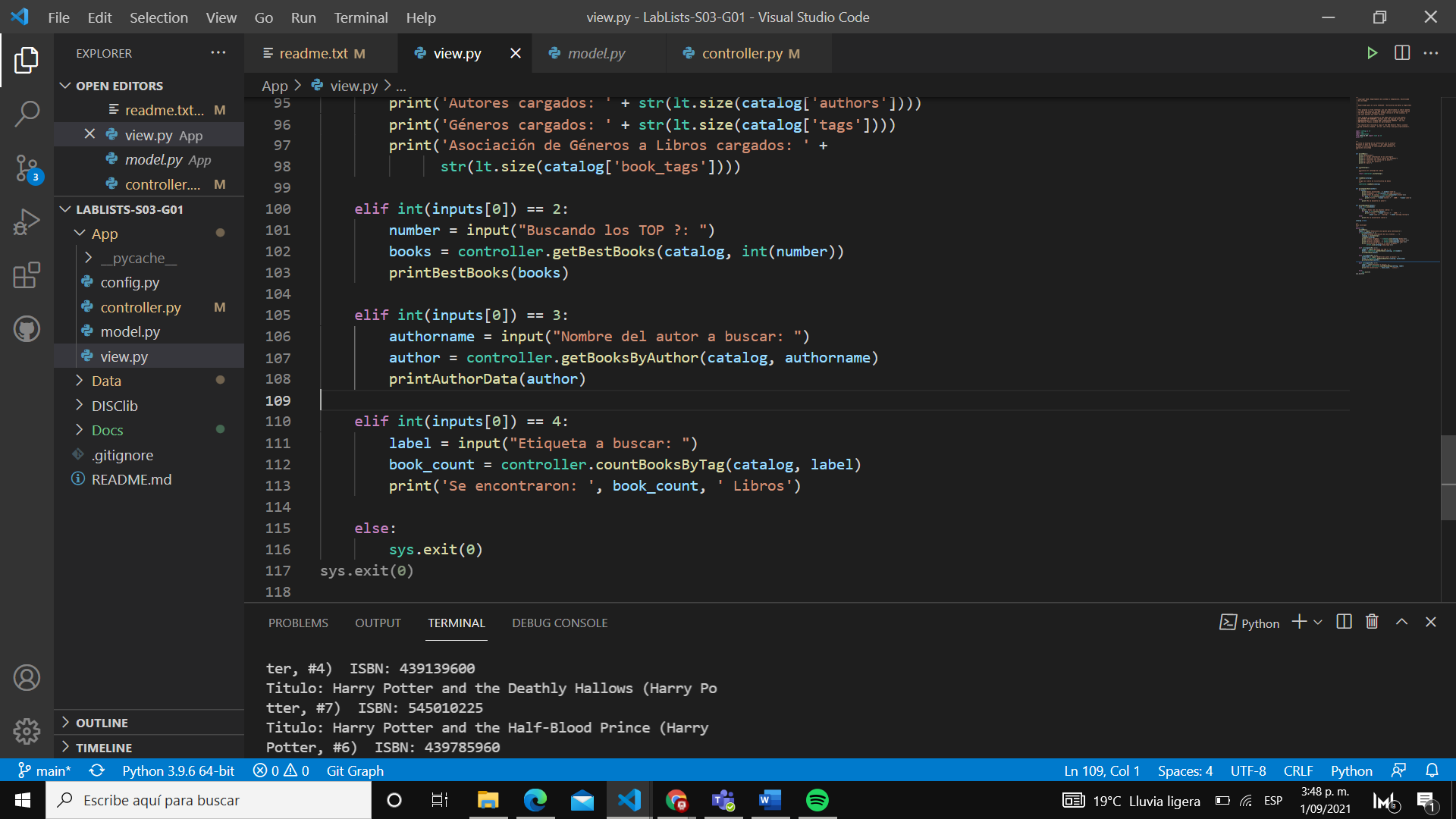


1. Los GoodReads se almacenan en model.py por medio de la función newCatalog(), en la cual se crea una lista para guardar todos los libros y esta empieza a llenar la lista con la información de los GoodReads.

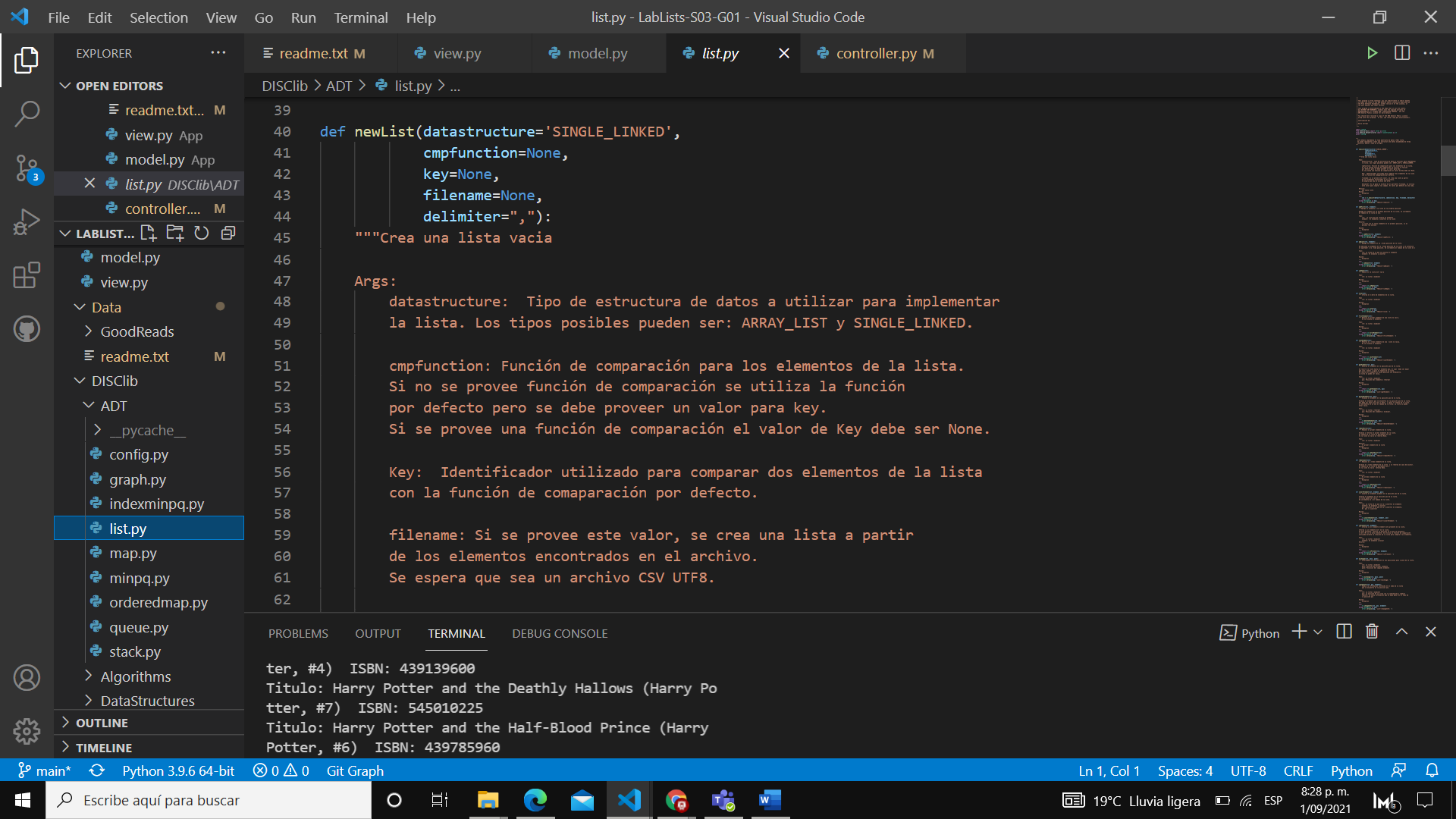


1. Las funciones que comunican del view.py con el model.py se encuentran en controller.py, las cuales son getBooksByAuthor(), getBestBooks(), countBooksByTag(), estas funciones llaman a las que se encuentran en model.py y estas después pasan al view.py por medio de la extensión controller. y el nombre de la función.





1. Por medio de la función newList se crea la nueva lista, y por medio de otras funciones se va llenando dicha lista que esta vacia.



1. Compara los elementos de la lista. Esa función es None ya que no va a comparar dichos elementos.
2. Agrega un elemento en la última posición de la lista y se actualiza el apuntador a la última posición, lo que hace que se incremente el tamaño de la lista.
3. Recorre la lista hasta el elemento necesario. Retorna el elemento en dicha posición sin eliminarlo, por lo que la lista no puede ser vacía.
4. Retorna una sublista de la lista. Se retorna una lista que contiene los elementos a partir de la posición requerida, con una longitud un numero de elementos. Se crea una copia de dichos elementos y se retorna una lista nueva.
5. No se encontraron cambios significativos, pero si se notó un cambio en el tiempo de respuesta--.