FUNCIONALIDAD:

El controlador (clase ControladorBiblioteca) se encarga de gestionar la interacción entre la vista (clase InterfazBiblioteca) y los modelos (clases Usuario, Libro y Biblioteca) a través de las respectivas instancias de cada clase que pasan a ser sus atributos.

La vista se encarga de:

1 – Reproducir la interfaz con la que el usuario interactúa al llenar el formulario destinado a la creación de un objeto Usuario que lo represente.

2 – Permitirle al usuario el alquilar y devolver libros.

3 – Permitirle al usuario el agregar libros a la colección de la biblioteca.

4 – Permitirle al usuario el buscar libros según título, autor o género.

5 – Permitirle al usuario cancelar las distintas operaciones iniciadas y salir del menú.

El controlador se encarga de:

1 – Validar los datos ingresados en un campo (si son nulos, si son campos vacíos…) y arrojar la excepción pertinente según sea el caso (NullPointerException, InputMismatchException) la cual será impresa por la vista a través de un mensaje de error.

2 – Crear libros, o bien sumar o restar libros según la operación (alquilar / devolver).

3 – Devolver el inventario de libros según el tipo de operación (vista general, alquiler, devolver, buscar).

Los modelos se encargan de:

(Usuario) -> Instanciar, alquilar y devolver libros pertenecientes a la lista de libros alquilados por el usuario. Sobreescribe los métodos hashCode y equals en base al nombre, lo que permite al usuario no perder sus datos asociados.

Datos -> nombre(String), librosAlquilados(LinkedList)

(Libro) -> Instanciar, agregar, prestar y actualizar ejemplares, como así determinar su disponibilidad. Sobreescribe los métodos hashCode y equals en base al título y autor, lo que permite al programa evitar la generación de repetidos.

Datos -> titulo(String), autor(String), cantEjemplares(int), librosAlquilados(Genero)

(Biblioteca) -> Instanciar, prestar libros como así también agregar usuarios o libros. Implementa la interface Busqueda, lo que le permite al usuario realizar búsquedas en la biblioteca en base a título, autor o género.

Datos -> usuarios(LinkedList), libros(LinkedList)

VALIDACIÓN:

Se implementó la validación en el controlador, el cual es invocado por la vista cada que se ingresan datos por el usuario. Las validaciones hechas son las siguientes:

1 – Método “esNulo”: Recibe un String y arroja un NullPointerException de encontrar un null. Un método de iguales características se aplica en el caso del dato Genero.

2 – Método “esValido”: Recibe un String y arroja un InputMismatchException de encontrar que esté vacío o es un espacio.

INFORME:

Fue preciso manejar los null que podían desencadenarse en medio de un ingreso de datos, al presionar “Cancelar” o “X”. Para eso se estableció un curso de acción:

1 – De obtener un nulo por parte de los distintos métodos de la interfaz, se le ofrecería al usuario la opción de cancelar la operación (método cancelFrame)

2 – cancelFrame a su vez le dará la opción de regresar al menú (menuFrame) o salir, tras lo cual el usuario se vería obligado a registrarse o iniciar sesión (userFrame) a determinar según el nombre que se ingrese).

3 – De cancelarse la anterior operación, el usuario podrá terminar o no con la ejecución del programa (cancelUserFrame).

También fue difícil determinar cómo evitar la reaparición de ventanas previamente cerradas, lo que se solucionó combinando break y return; dentro de menuFrame por cada una de las opciones ingresadas.