Valentina Calderón: 202020771.

Daniela Espinosa: 202022615.

Juan Camilo Neira: 201922746.

Enlace: <https://github.com/jneiracampos/Taller6_Libreria.git>

**Parte 1.**

En la clase Librería agregamos un ArrayList de las nuevas categorías y un HashMap que tiene como llaves los nombres de las nuevas categorías y como valores los ArrayList de los libros que pertenecen a cada una. Después, en el método cargarCatalogo, se revisa si después del método buscarCategoria la variable laCategoria es null (es decir que la categoría no existe). De ser así, se crea la categoría y el libro asociado a esta para poder llenar las nuevas estructuras de datos con sus respectivos datos.

Por otro lado, en la clase PanelCategoria agregamos el método actualizarCategoriasNuevas que agrega las nuevas categorías al ComboBox y en la clase InterfazLibreria en el método cargarArchivos invocamos ese nuevo método del PanelCategoria.

Posteriormente, en InterfazLibreria también agregamos un nuevo método al final cuantosLibrosCategoriasNuevas que informa al usuario las nuevas categorías que se crearon y el número de libros que se agregaron a cada una.

Finalmente en PanelBotones agregamos el botón CATEGORIAS\_NUEVAS que es el encargado de llamar al método cuantosLibrosCategoriasNuevas para mostrar el aviso.

**Parte 2.**

Se creó un nuevo método en la clase Librería que se llama cambiarNombreCategoria(). Este recibe dos cadenas de caracteres: el nombre de la categoría que se quiere cambiar y el nuevo nombre. Esté método hace uso interno del método buscarCategoria(), el cual retorna una variable de tipo Categoria si se encontró el nombre de la categoría y, en caso contrario, un valor nulo. Este método se utilizó para conocer si el nuevo nombre ya es usando por una categoría. En caso de que buscarCategoria() no retorne null (es decir que el nuevo nombre ya existe) se crea una excepción que dice “ya existe una categoría con este nombre” y, en caso contrario, se llama al método cambiarNombre() de la clase Categoría. Asimismo, en InterfazLibreria se debe manejar el siguiente error. El usuario debe ingresar el nombre de la categoría que quiere cambiar, pero es posible que ingrese un nombre que no existe y por tanto el método buscarCategoria() retornaría un null y esto generaría un error. Así que se creó un método en la clase Librería que se llama existeCategoria() que retorna un valor booleano si el nombre que entra por parámetro es usado por alguna categoría. De tal modo, se puede verificar que el nombre que el usuario quiere cambiar en efecto existe. Cabe señalar que si el nombre que se quiere cambiar no existe o el nuevo nombre está repetido, se genera una pestaña en la que se le informa al usuario sobre su error para que este lo corrija.

**Parte 3.**

Agregar a la aplicación la funcionalidad para eliminar libros requirió de la creación de un método borrarLibros y eliminarLibrosAutor, y se utilizó try catch y exception para el manejo de errores.

En InterfazLibreria se creó el método borrarLibros que se encarga de interactuar con el usuario. Se le solicita a los usuarios que escriba los autores de los libros que desea eliminar. Se usa un try catch para que en caso de que todos los autores pertenezcan a un libro en la librería, se retorne los libros eliminados, y en caso de que no, retorne un error e informe que no se pudo borrar porque un autor x no tiene un libro.

En Librería se creó el método eliminarlibrosAutor que genera una excepción si los libros no pueden eliminarse. Lo anterior debido a que la aplicación no puede borrar los libros si dentro de la lista de autores al menos uno no existía. En este caso de que no todos tengan un libro, al usuario se le debe informar cuáles de los nombres digitados correspondían a autores y cuáles no. En caso de que todos existan, al usuario se le debe informar la cantidad de libros borrados.