

Taller 6

Integrantes:

- Nicolas Merchan Cuestas - n.merchan@uniandes.edu.co - 202112109
- Julián Camilo Rivera Monroy - jc.riveram1@uniandes.edu.co - 202013338
- Juan Felipe Serrano - j.serrano@uniandes.edu.co - 201921654

Parte 1: Libros en categorías que no existen

El registro de libros en categorías que no existen en el archivo "categorías.csv" se realiza por medio de la adición de una condición adicional en el método "buscarCategoria()" de la clase "librería". De no encontrarse la categoría del libro referente, se hace uso del nuevo método "agregarCategoria()" de la clase "librería". El método "agregarCategoria()" retorna la nueva categoría previamente inexistente y agrega dicha categoría en el arreglo referente a las categorías en la clase "librerías", "categorías". Para ello es necesario crear otro arreglo con una posición adicional para la nueva categoría y al cual le vamos a agregar todas las categorías ya presentes y la nueva categoría. Luego reemplazamos el arreglo de categorías de "librería" por aquel que acabamos de crear y que incluye la nueva categoría. Vale la pena resaltar que ello se realiza antes de crear el libro en cuestión, ya que es necesario tener una categoría para incluir en esta el libro. Consideramos que la implementación ya mencionada es simple y permite evitar errores relacionados con la creación de libros en categorías inexistentes en el arreglo de categorías.

Parte 2: Renombrar categorías

El proceso de renombrar categorías se realiza por medio de la creación de métodos adicionales en la clase "librería" y condicionales en la interfaz que permiten mostrar los cuadros de dialogo referentes a los diferentes errores que se pueden presentar. En primer lugar, el método "renombrarCategoria()" de la interfaz hace uso de dos cuadros de dialogo para permitirle al usuario ingresar el nombre de la categoría que desea renombrar y, posteriormente, ingresar el nuevo nombre que desea asignarle a la categoría. En ese orden de ideas, cuando el usuario ingresa la categoría que desea renombrar se verifica que dicha categoría exista por medio del método "buscarPosCategoria()" de la clase "librería". Dicho método retorna la posición de la categoría en el arreglo de categorías de la clase "librería", categorías. De no existir la categoría se retorna -1.

En segundo lugar, si la categoría existe, entonces se hace uso del método "renombrarCategoria()" de la clase "librería". Dicho método hace uso, a su vez de un método auxiliar "existeCategoria()" para verificar que el nuevo nombre de la categoría no coincida con el nombre de otra categoría ya existente. Así, si no existe otra categoría con el mismo nombre, se crea una nueva categoría con el nuevo nombre y se reemplaza la posición de la categoría del nombre anterior con la nueva categoría renombrada. Después, se vuelven a cargar los libros agregando aquellos de la categoría anterior en la nueva categoría. Vale la pena mencionar que existen dos condicionales en la interfaz

que permiten informarle al usuario ya sea que la categoría que desea renombrar no existe o que el nuevo nombre ya es utilizado por otra categoría, los únicos dos casos posibles. Consideramos que la implementación ya mencionada permite validar de manera eficiente que no existen categorías con el nuevo nombre y permite agregar una categoría renombrada conservando los libros pertenecientes a la clasificación con el nombre anterior.

Parte 3: Borrar libros

El proceso de borrar libros se realiza por medio de la creación de métodos adicionales en la clase “libreria” y condicionales en la interfaz que permiten mostrar los cuadros de dialogo referentes a los diferentes errores que se pueden presentar. En primer lugar, el método “borrarLibros()” de la interfaz hace uso de un cuadro de dialogo para permitirle al usuario ingresar los nombre de los autores de los libros que desea eliminar. En ese orden de ideas, cuando el usuario ingresa los nombres de los autores se verifica que dichos autores tenga al menos un libro registrado en la lista “catalogo” de la clase “libreria” por medio del método “existenAutores()” también presente en dicha clase. El método “existenAutores()” crea un mapa con los nombre de los autores y va removiendo los autores de dicho mapa a medida que va recorriendo los libros y verificando que son autores de al menos un libro registrado. De tal forma, “existenAutores()” retorna un mapa vacío si todos los autores tiene al menos un libro registrado y un mapa con al menos un autor cuando al menos uno de los autores no tiene libros registrados. Si uno o más de los autores no tienen registrados libros, se hace uso del mapa para generar un cuadro de dialogo en el cual se le informe al usuario cuales autores no tienen libros registrados.

En segundo lugar, si todos los autores tienen libros registrados, entonces se hace uso del método “eliminarLibros()” de la clase “libreria”. Dicho método retorna el número de libros eliminados y elimina los libros de la lista “catalogo” y las categorías del arreglo “categorias” de la clase “libreria” que tengan como autores aquellos ingresados por parámetro. Así mismo, se considera un try/catch que permite saber la información general de un libro y el error en caso de que se produzca un error en la eliminación de algún libro en el recorrido de las listas. Consideramos que la implementación ya mencionada facilita generar cuadros de dialogo que le permitan al usuario saber que autores no tienen libros registrados y permite eliminar todos los libros de los autores tanto de la lista de libros como de las categorías teniendo en cuenta posibles errores en la eliminación de libros.