

# Documento de diseño: Librería

## Taller 06

Camilo Morillo Cervantes y Juan Sebastián Ortega Romero

*Departamento de Ingeniería de Sistemas y Computación, Universidad de los Andes*

27 de noviembre de 2021

### 1. Libros en categorías que no existen.

Para agregar una nueva categoría cuando un libro posee dicha categoría y esta no esta registrada en el archivo de categorías se modificara el método **cargarCatalogo** de la clase Librería del modelo. Este método es llamado desde el constructor de la clase y tiene como parámetro el nombre del archivo de libros. Dado que el constructor primero llama al método **cargarCategorias**, a la hora de llamar al método cargarCatalogo ya tendremos las categorías cargadas y por lo tanto comprobaremos si la categoría de cada libro ya existe en las categorías cargadas. En caso de que no exista dicha categoría se agregara mediante un nuevo método de la misma clase Librería llamado **agregarNuevaCategoria**. Posteriormente mediante una nueva clase dedicada al manejo de excepciones llamada **CustomNullException**, la cual podemos observar en la Figura 1, además de un nuevo método en la clase InterfazLibreria llamado **errorHandlerInterfaz**, vamos a dar la notificación del nombre de las categorías agregadas y el numero correspondiente de libros pertenecientes a esta.

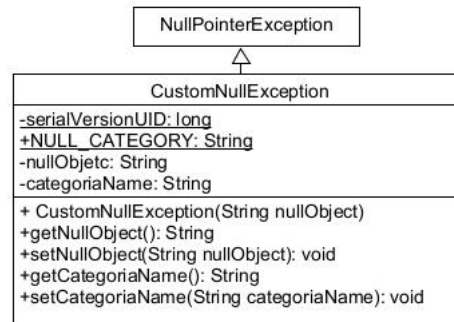


Figura 1: Clase CustomNullException

### 2. Renombrar categorías.

Para poder renombrar una categoría estaremos modificando en primer lugar la clase PanelBotones de la interfaz para así agregar un botón que cumpla esta funcionalidad. Una vez este botón halla sido presionado se llamara a un nuevo método de la clase InterfazLibreria (Clase principal de la interfaz) llamado **renombrarCategoria**. Dicho método le pedirá al usuario que categoría quiere renombrar y el nuevo nombre que se le asignara a esta categoría. En caso de que dicho nombre se encuentre repetido y con el objetivo de cumplir con la restricción propuesta (La interfaz no puede ser la encargada de comprobar que se encuentre repetido un nombre) la clase Librería sera la encargada de comprobar si el nombre se encuentra repetido. Esto se hará mediante el nuevo método **obtenerCategoria** de la clase Librería el cual retorna la categoría cuyo nombre se cambiara o en caso de que el nombre ingresado este repetido tirara una excepción. La excepción dada es una nueva excepción creada llamada **CustomRepeatedException** cuya clase podemos observar en la Figura 2. Una vez esta excepción es tirada sera finalmente atrapada por el método anterior de renombrarCategoria de la clase InterfazLibreria, el cual finalmente notificara al usuario de que el nombre ingresado esta repetido.

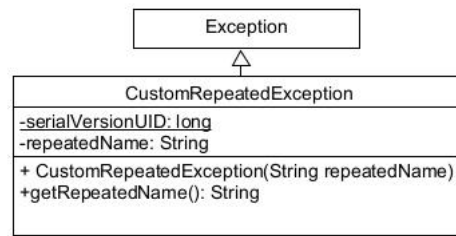


Figura 2: Clase CustomRepeatedException

### 3. Borrar libros.

En primer lugar, para implementar esta funcionalidad se modificó la clase PanelBotones para agregar un botón encargado de esta funcionalidad. Una vez dicho botón es presionado se ejecutará el nuevo método **eliminarLibrosAutor** de la clase InterfazLibreria. Dicho método le pedirá al usuario que ingrese el nombre de los autores cuyos libros se eliminarán (con los nombres separados por comas). En orden de cumplir con la restricción propuesta (La interfaz no puede encargarse de revisar la existencia de los autores) se revisará la existencia de los autores utilizando la clase Librería. Esto mediante el nuevo método **buscarLibrosAutor** de la clase Librería el cual conseguirá los libros a eliminar del autor ingresado y se agregará el autor a los autores existentes. En caso de que dicho método no encuentre libros del autor ingresado arrojará una excepción nueva la cual se puede observar en la clase **CustomNullAuthors** de la Figura 3a. Esta excepción es agarrada nuevamente por el método **eliminarLibrosAutor** de la clase InterfazLibreria, la cual agrega dicho autor a los autores cuyos libros no se han encontrado para posteriormente arrojar un mensaje con los nombres de autores encontrados y no encontrados (en caso de que no exista algún autor). En caso de que se encuentren libros de todos los autores ingresados, se procederán a eliminar todos los libros de dichos autores mediante el nuevo método **eliminarLibro** de la clase Librería. Este método a su vez en caso de que no pueda eliminar algún libro arrojará una nueva excepción llamada **CustomMissingBook** cuya clase se puede observar en la Figura 3b. Esta excepción es nuevamente agarrada por el método **eliminarLibrosAutor** el cual mostrará un mensaje de error al usuario. Finalmente, se mostrará un mensaje con la cantidad de libros eliminados de forma exitosa.

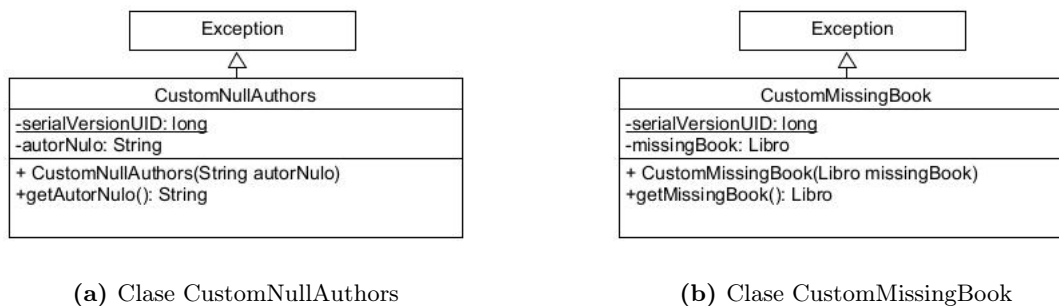


Figura 3: Excepciones para la funcionalidad de borrar libros