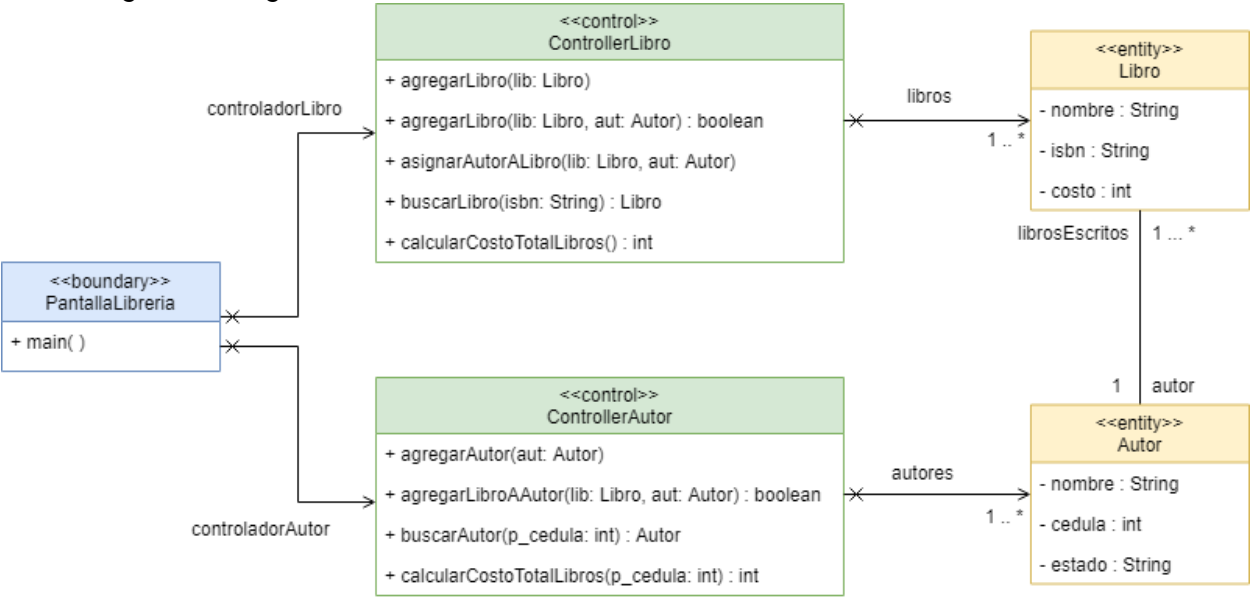


La entrega del laboratorio consistirá de **1 archivos zip**, con el proyecto de maven. No incluya la carpeta target. Este archivo contendrá el código fuente del proyecto Java realizado por el estudiante.

Enunciado:

Para el siguiente diagrama de UML:



- Cree tres paquetes: view (clase *PantallaLibreria*), controller (para las clases *ControllerLibro* y *ControllerAutor*) y model (para las clases *Libro* y *Autor*).
- Cree las clases y métodos mostrados en el siguiente diagrama de clases (no se muestran constructores ni accessors pero deben ser creados).

Reglas de Negocio:

- Los posibles valores del estado del autor son 'ACTIVO', 'INACTIVO'

Observaciones de métodos de ControllerAutor:

- `agregarAutor(Autor aut)`: Agrega un nuevo autor
 - No pueden existir dos autores con el mismo número de cédula
 - Cuando se cree el autor cree la lista de libros vacía
- `agregarLibroAAutor(Libro lib, Autor aut)`: Agrega un Libro que debe existir a un Autor que ya existe
 - Se debe buscar el autor que llega como parámetro.
 - No se pueden agregar a un autor dos libros con el mismo isbn.
 - Al autor se le debe vincular el libro que llega como parámetro.
 - En la lista de 'librosEscritos' sólo se pueden asignar libros de la lista de 'libros'.
- `calcularCostoTotalLibros(int p_cedula)`
 - Calcular el costo total de los libros del autor cuya cédula llega como parámetro.
- `buscarAutor(int p_cedula)`
 - Buscar y retornar el autor cuya cédula llega como parámetro.

Observaciones de métodos de ControllerLibro:

- agregarLibro(Libro lib): Agrega un libro nuevo sin Autor.
 - No pueden existir dos libros con el mismo isbn
- agregarLibro(Libro lib, Autor aut): Agrega un libro nuevo con un autor aut
 - No pueden existir dos libros con el mismo isbn
 - El autor debe existir en la lista de autores
- asignarAutorALibro(Libro lib, Autor a): Asigna a un libro que debería existir a un autor que debe existir
 - El autor debe existir en la lista de autores
 - No se pueden asignar a los libros autores que tengan estado = 'INACTIVO'
- buscarLibro(String isbn): buscar y retorna un libro dado su isbn.
- calcularCostoTotalLibros(): calcular el costo total de todos los libros existentes en el sistema.

Relación Muchos a muchos

Modifique su aplicación para que un libro tenga muchos autores. Tenga en cuenta que:

- Se deben hacer cambios en los atributos y accessors de las clases Libro y Autor
- Los controladores deben incluir los métodos que permitan vincular un libro a un conjunto de autores y viceversa

Adicionalmente se pide:

- En el método main() de la pantalla usted debe crear los objetos y las llamadas necesarias que permitan probar cada uno de los métodos de los dos controladores.
- Se deben probar en cada método todos los caminos (if..)
- Sobrecargue el método *toString()* de las clases Libro y Autor, para imprimir por pantalla de forma personalizada la información de los objetos instancias de estas clases.

Bono Adicional (+0.5 pts.) !!!

- Modifique su aplicación para que realice la relación muchos a muchos entre libro y autor a través de una clase intermedia, como se explicó en clase.
- Adicionalmente, sobrecargue el método equals() de las clases Libro y Autor para que se usen los datos de ISBN y cédula respectivamente como atributos que identifican a cada objeto y no se deben repetir. Con este cambio, los métodos agregar* pueden usar el método contains(...) de la colección para evitar repetidos.