

Actividad 2

Patrón Observer

Queremos modelar la dinámica de devolución de libros en una biblioteca. En el caso de que un libro devuelto esté en mal estado, se tendrán que desencadenar los procesos necesarios para la sustitución de dicho libro, de manera que se instrumenta un mecanismo para notificar a las áreas interesadas (Stock, Compras y Administración), en caso de que se devuelva un libro estropeado.

El proceso comienza cuando un lector devuelve un libro con la ejecución del método *devuelveLibro()* de la clase **Libros**, el cual no devuelve nada y recibe como argumento un *booleano* que pondrá a *true* la variable *malEstado*.

Si el lector devolvió el libro dañado entonces la aplicación avisa a las clases **Stock**, **Compras** y **Administración**, que se ven afectadas por este hecho. Cada vez que el libro se devuelve y está estropeado se avisa y se actualiza el atributo *malEstado* de la clase **Libros**; esta clase es el sujeto de interés para Stock, Compras y Administración. Dichas clases se convierten en observadoras de la clase **Libros**, que tiene la responsabilidad de notificarles el mal estado del libro, además del título del libro del que se trate. Para poder recibir esta notificación las clases deben suscribirse como observadoras, suscripción que pueden cancelar posteriormente, e implementar la interfaz **Observer**.

1. Proponer una solución usando el patrón *Observer*, y que se sirva de una filosofía PULL.
 - Dibujar el diagrama de clases.
 - Implementarlo en Java, incluyendo una clase que permita probarlo.
2. Proponer otra solución utilizando el patrón *Observer* proporcionado en el API de Java, manteniendo la filosofía PULL.
 - Dibujar el diagrama de clases.
 - Implementarlo en Java, incluyendo una clase que permita probarlo.

