

# Trabajo Práctico Numero 4

Bustos Gil Felipe División 2A

El fin de mi aplicación es vender libros en una librería. Los tipos de libros que se venden son: Cuentos, novelas y libros de investigación. Los tres poseen cantidad de paginas y tipo de cubierta(cuentos y novelas, tapa blanda y tapa dura los de investigación). Cada uno tiene una característica particular:

- Cuento -> Ilustraciones.
- Novela -> Autografiado.
- Investigación -> Cantidad de tomos.

## La solución cuenta varios proyectos:

Por ejemplo una biblioteca de clases con una clase base que es Libro, de la cual derivan Cuento, Novela e Investigación.

La clase Libro posee la editorial del libro, el título y el precio. Y las clases que derivan aparte de poseer estos atributos por herencia, poseen los atributos mencionados al inicio.

Dos formularios. Entre otras.

## Funcionalidad

<i>id</i>	<i>editorial</i>	<i>titulo</i>	<i>tipo</i>	<i>precio</i>	<i>caracteristica</i>
1	Apagoue	Autonomia cientifica	Investigacion	3500	2 tomos
2	Aranzadi	Economia mundial	Investigacion	4200	4 tomos
3	Imaginante	El patito feo	Cuento	360	Incluye ilustraciones
4	Sigmar	El mago de Oz	Cuento	400	No incluye ilustraciones
5	Entropia	Moby Dick	Novela	1200	Sin autografiar
6	Edelvives	El moustruo perfecto	Novela	1700	Autografiado

Este es el formulario principal, que cuenta con ciertos botones:

- Boton **Agregar** que permite agregar stock a la tabla de libros y a la base de datos(explicado en el apartado hilos), a través del formulario secundario .(Mostrado abajo)

Formulario 'Agregar stock' con los siguientes campos:

- Editorial:
- Título:
- Tipo:
- Precio:
- Característica:
- Botones: Aceptar, Cancelar

Este formulario permite ingresar los datos del libro. Al elegir un tipo de libro producirá un cambio en el label Característica, con la característica adecuada para ese tipo de libro y además modificará el contenido de su comboBox.

Si se hace click en el botón **Cancelar** se cancelará el ingreso del producto, regresando al formulario anterior. En el caso de hacer click en **Aceptar** se agregará a la tabla, se verá el cambio en el form principal. **No se pueden dejar campos en blanco, ni agregar un libro que ya se encuentre en el stock.** La imagen siguiente muestra como se agregó un libro, teniendo como referencia la imagen de muestra del formulario principal.

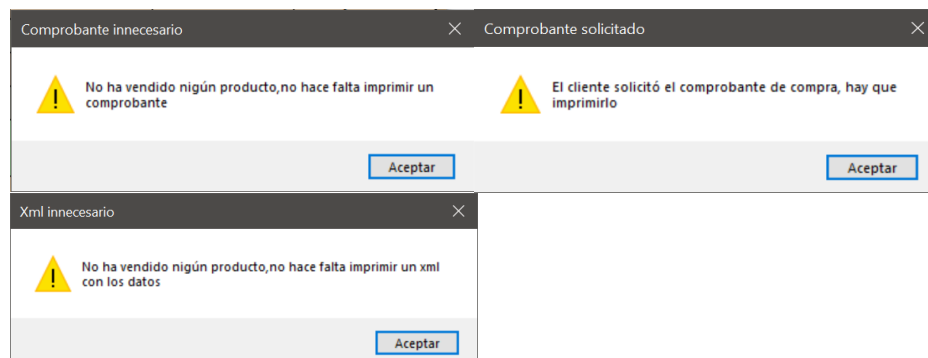
Formulario 'Libreria' con la siguiente tabla de libros:

id	editorial	título	tipo	precio	característica
1	Apagoue	Autonomia cientifica	Investigacion	3500	2 tomos
2	Aranzadi	Economia mundial	Investigacion	4200	4 tomos
3	Imaginante	El patito feo	Cuento	360	Incluye ilustraciones
4	Sigmar	El mago de Oz	Cuento	400	No incluye ilustraciones
5	Entropia	Moby Dick	Novela	1200	Sin autografiar
6	Edelvives	El moustruo perfecto	Novela	1700	Autografiado
7	Limonero	Caperucita Roja	Cuento	340	Incluye ilustraciones

Botones: Cerrar, Agregar, Vender, Venta Xml, Comprobante

- El botón **Vender** venderá el producto seleccionado (la fila seleccionada se visualiza en gris) lo que provocará que se lance un mensaje y se muestra el precio en un label en la parte superior además de esto agregará el producto vendido a una lista de vendidos que será utilizada en los dos botones siguientes.
- El botón **Venta Xml** serializará en Xml la lista de vendidos, y guardará el archivo en el Escritorio, siempre y cuando se haya realizado al menos una venta.

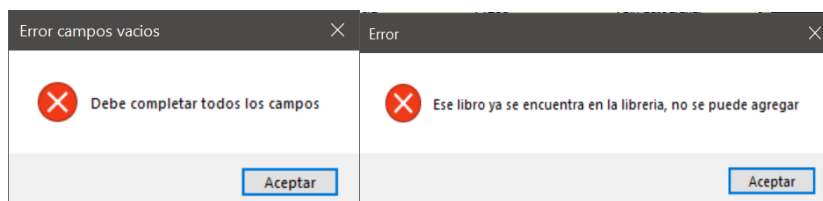
- El botón **Comprobante** con la venta, utilizará la lista de venta para guardar en un txt los datos de la venta (fecha, productos y total). También, siempre y cuando se haya realizado una venta.
- Y por último el botón **Cerrar** que cerrará el formulario.
- Como aclaración, se lanzarán advertencias y errores, algunos relacionados a excepciones creadas por mí (detalladas en la parte de excepciones) y otros por llevar o no a cabo acciones en la aplicación. Ejemplos :



## Detalle de las clases 15 a 25

### Clase 15 “Excepciones”:

El programa controla todo tipo de excepciones y además posee algunas excepciones personalizadas. Ejemplos:



Estos son los mensajes lanzados al producirse alguna de las excepciones creadas por mí.

### Clase 16 “Test unitarios”:

El test unitario prueba tres funcionalidades, que se agregue correctamente un Cuento a una Librería, que la lista de libros de una Librería no sea null y que se haya realizado la conexión a la base de datos correctamente.

### Clase 17 “Tipos genericos”, Clase 18 “Interfaces” y Clase 19 “Archivos”:

Se utiliza una interfaz genérica para serializar en Xml. Y aparte, sin estar en la interfaz, para generar el Comprobante se utiliza el guardado en un archivo de texto.

### **Clase 21 "SQL" y Clase 22 "Base de datos":**

Utilizo base de datos SQL para traer el stock de libros al formulario principal y tambien se pueden generar cambios en esa base de datos (agregar libros).

### **Clase 23 "Hilos":**

Utilizo hilos para ir actualizando la base de datos cada 1 segundo y muestra si se actualizo la base de datos en un label en la parte superior derecha.

### **Clase 24 "Eventos":**

Utilizo un evento para que si al momento de agregar un libro ingresan un precio invalido(letras, o una cadena vacia, o un número que supere el maximo valor de un entero) asigne según el tipo de libro un precio específico.

### **Clase 25 "Metodos de extensión":**

Aplico un metodo de extensión a la excepcion LibroRepetidoException.