



La librería

Instancia de evaluación	Cuatrimestre
Primer Parcial	1ero 2021

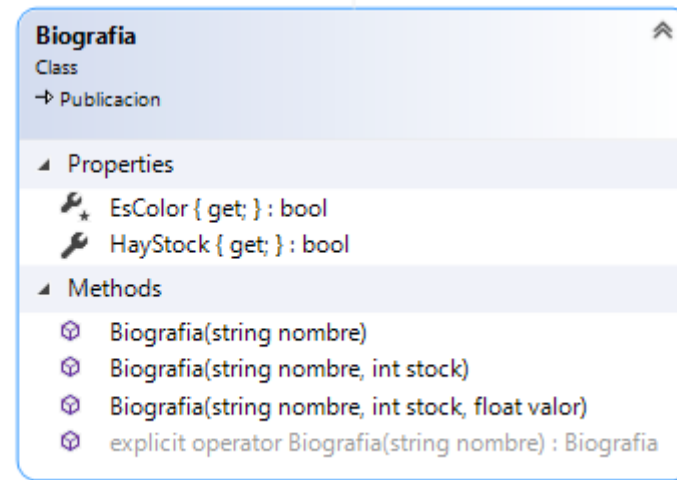
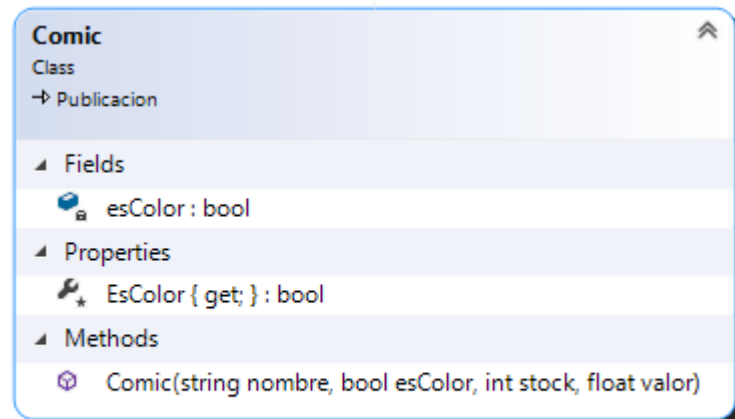
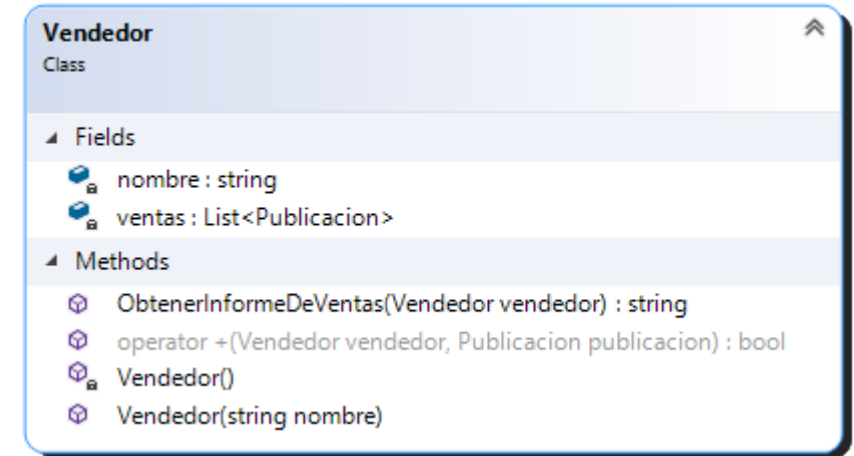
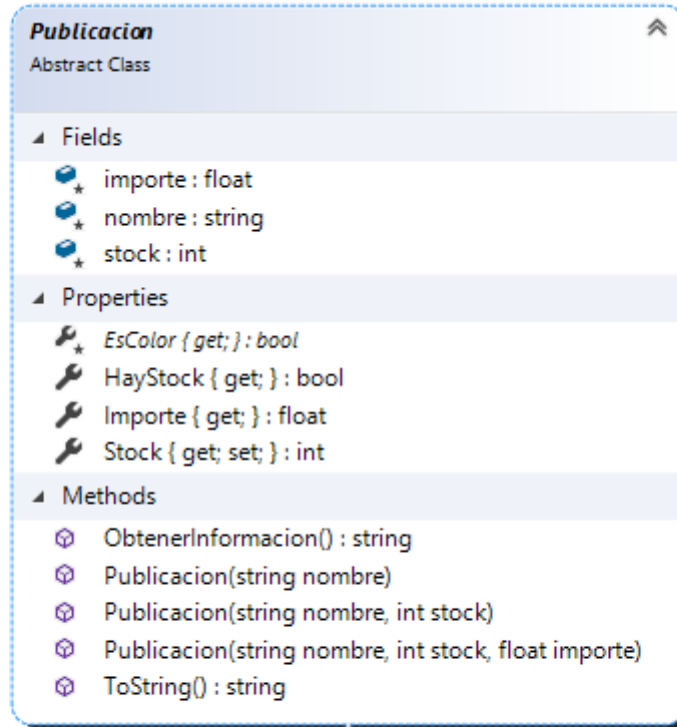
Criterios de evaluación

- **2 (dos) errores en el mismo tema anulan su puntaje.**
- La correcta documentación y reglas de estilo de la cátedra serán evaluadas.
- Colocar sus datos personales en el nombre de la carpeta principal y la solución: *Apellido.Nombre.Div. Ej: Pérez.Juan.2D*. No sé corregirán proyectos que no sea identificable su autor.
- No se corregirán exámenes que no compilen.
- **Reutilizar** tanto código como crean necesario.
- Colocar nombre de la clase (en estáticos), `this` o `base` en todos los casos que corresponda.
- Aplicar los principios de los 4 pilares de la POO.

Consigna

Crear un proyecto del tipo **Biblioteca de Clases** y colocar el siguiente esquema de clases, sin agregar nada extra:





Clase Publicacion

- Los métodos y propiedades marcados en cursiva son abstractos.

- La propiedad `HayStock` será virtual, retornará `true` siempre y cuando haya stock y el importe sea mayor a 0 (cero).
- La propiedad `Stock` solo asignará si lo recibido es mayor a 0 (cero).
- El método `ToString` retornará el nombre (servirá para el formulario).
- El método `ObtenerInformacion` retornará los datos de la `Publicacion` utilizando `StringBuilder` (deberá utilizar todos los métodos `Append` disponibles). Completar las palabras en mayúscula con los datos de cada elemento según corresponda:

```
Nombre: NOMBRE  
Stock: STOCK  
Es color: SI/NO  
Valor: $IMPORTE
```

Clase `Biografia`

- La propiedad `HayStock` controlará solo que haya stock, sin importar el valor de venta.
- La propiedad `EsColor` siempre será que NO.
- La conversión explícita generará un objeto `Biografia`.

Clase `Comic`

- Sin particularidades.

Clase `Vendedor`

- La lista se inicializará en el constructor privado.
- El operador `+` agregará siempre y cuando haya stock suficiente para realizar una venta del producto, agregando en ese caso la información de la venta a la lista y descontando stock de la publicación.
- El método `ObtenerInformeDeVentas` será de clase y retornará la información del vendedor, sus ventas y la ganancia total de las mismas con el siguiente formato:

```
NOMBRE
-----

PUBLICACION: INFORMACION
-----

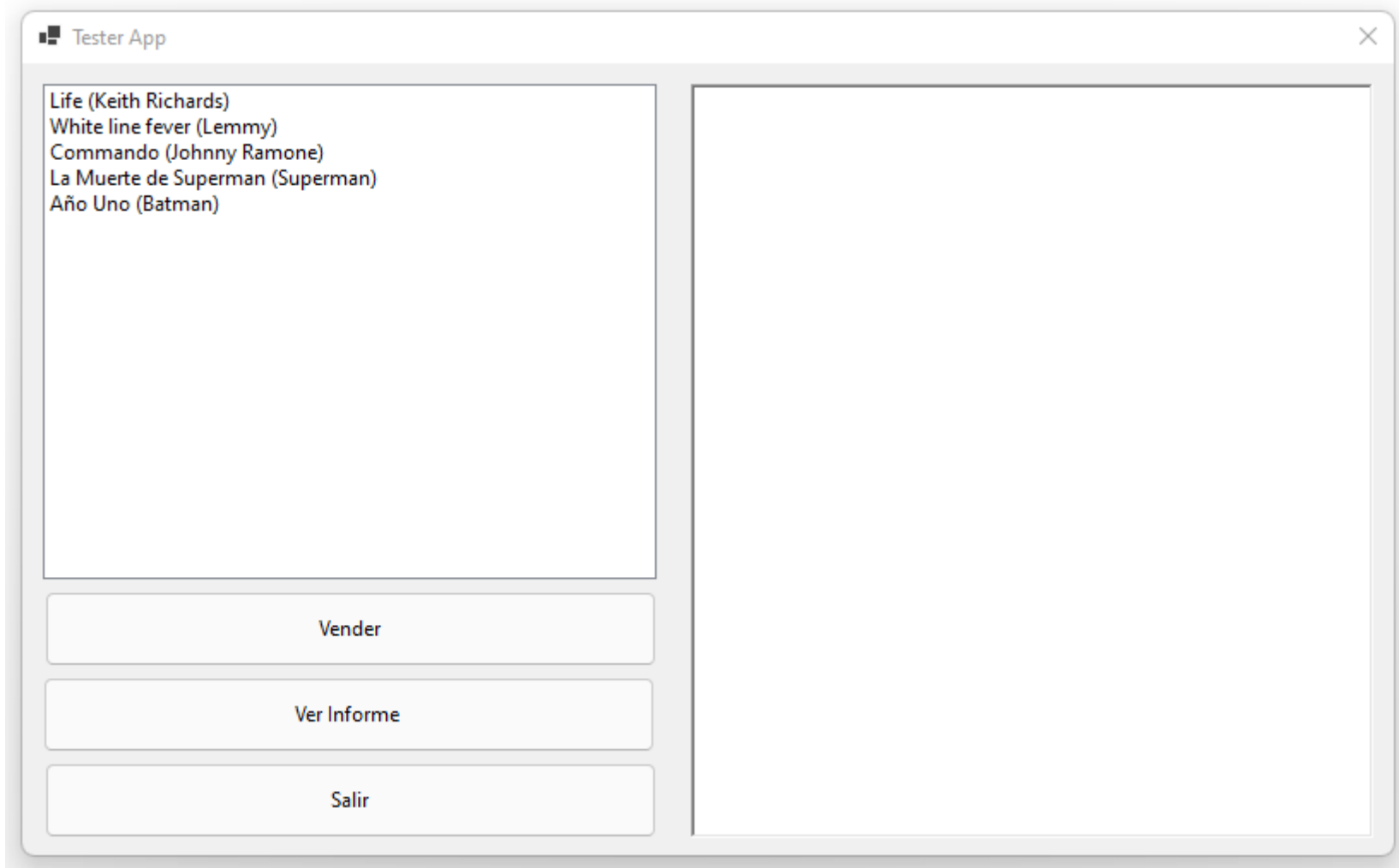
PUBLICACION: INFORMACION

Ganancia Total: $GANANCIA
```

Clase

FrmTest

- El formulario debe iniciar centrado en la pantalla con el siguiente formato:

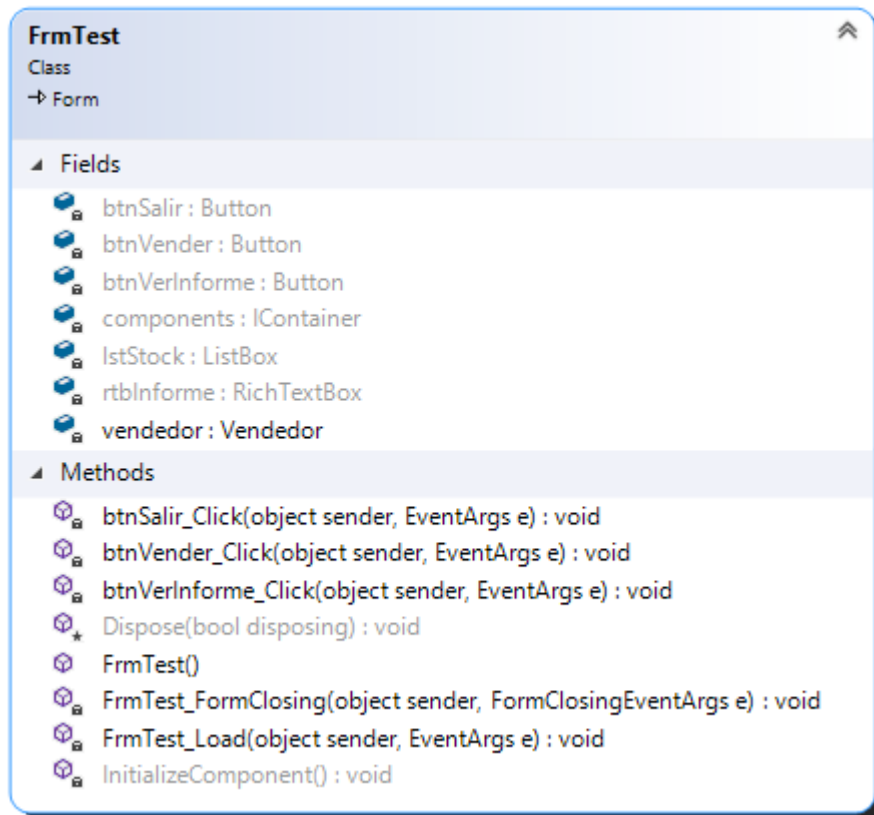


- Quitar los botones de maximizar y minimizar.
- `FormBorderStyle` debe ser `FixedSingle`.
- El nombre del formulario debe ser *Tester App de Nombre Apellido (comision)*.
- El formulario debe iniciar centrado en la pantalla.
- Agregar el siguiente código al evento `Load` del formulario:

```
Biografia p1 = (Biografia)"Life (Keith Richards)";  
Biografia p2 = new Biografia("White line fever (Lemmy)", 5);  
Biografia p3 = new Biografia("Commando (Johnny Ramone)", 2, 5000);
```

```
Comic p4 = new Comic("La Muerte de Superman (Superman)", true, 1, 1850);  
Comic p5 = new Comic("Año Uno (Batman)", false, 3, 1270);
```

- El formulario deberá contar con los siguientes elementos:



- Agregar las publicaciones del evento `Load` al `ListBox lstStock`.
- El vendedor solo podrá ser instanciado en el constructor.
- `btnVender` será el encargado de realizar ventas, informando por pantalla si la venta pudo realizarse o no.
- Al presionar el botón `btnVerInforme` se mostrará en el `RichTextBox rtbInforme` los datos de las ventas del vendedor.
- Al intentar cerrar el formulario por cualquier método debe preguntar si está seguro de querer salir. Si la respuesta es NO, deberá permanecer abierto.

Resolución



Video



Código

*Last updated on **9/26/2021** by **mauriciocerizza***