

# Universidad Tecnológica Nacional Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

**Materia: Laboratorio de Programación II**

Apellido:		Fecha:	19/12/2019
Nombre:		Docente <sup>(2)</sup> :	
División:	2º	Nota <sup>(2)</sup> :	
Legajo:		Firma <sup>(2)</sup> :	
Instancia <sup>(1)</sup> :	<div style="display: flex; justify-content: space-around; align-items: center;"> <div>PP</div> <div></div> <div>RPP</div> <div>X</div> <div>SP</div> <div></div> <div>RSP</div> <div></div> <div>FIN</div> <div></div> </div>		

(1) Las instancias validas son: 1<sup>er</sup> Parcial (PP), Recuperatorio 1<sup>er</sup> Parcial (RPP), 2<sup>do</sup> Parcial (SP), Recuperatorio 2<sup>do</sup> Parcial (RSP), Final (FIN). Marque con una cruz.

(2) Campos a ser completados por el docente.

### IMPORTANTE:

- **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 del proyecto principal, colocando: Apellido.Nombre.Departamento. Ej: Pérez.Juan.2D. No se corregirán proyectos que no sea identificable su autor.
- **TODAS** las clases deberán ir en una Biblioteca de Clases llamada Entidades.
- 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.

---

*TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 90 MINUTOS.*

---

1. Crear una solución con el nombre en el siguiente formato: [APELLIDO].[NOMBRE]
2. Dentro crear 3 proyectos: *Entidades* (Class Library), *VistaConsola* (Console) y *VistaForm* (WindowsForms).
3. Dentro del **Program**, en el **Main** de *VistaConsola*, colocar el siguiente código para probar las entidades:

```
Factura f = new Factura("UTN", "20-12345678-2");
Producto p1 = new ProductoA("111", "Producto1", 100, new DateTime(2018, 07, 01));
Producto pd = new ProductoA("111", "Duplicado1", 100, new DateTime(2018, 07, 01));
Producto p2 = new ProductoB("112", "Producto2", 200, Tamanios.CHICO);
Producto p3 = new ProductoB("113", "Producto3", 250, Tamanios.MEDIANO);
Producto p4 = new ProductoB("114", "Producto4", 300, Tamanios.GRANDE);
Producto p5 = new ProductoB("115", "Producto5", 400);

f += p1;
f += pd;
f += p2;
f += p3;
f += p4;
f += p5;

try
{
    Producto p6 = new ProductoA("116", "Producto6", 100, new DateTime(2018, 06, 15));
```

```

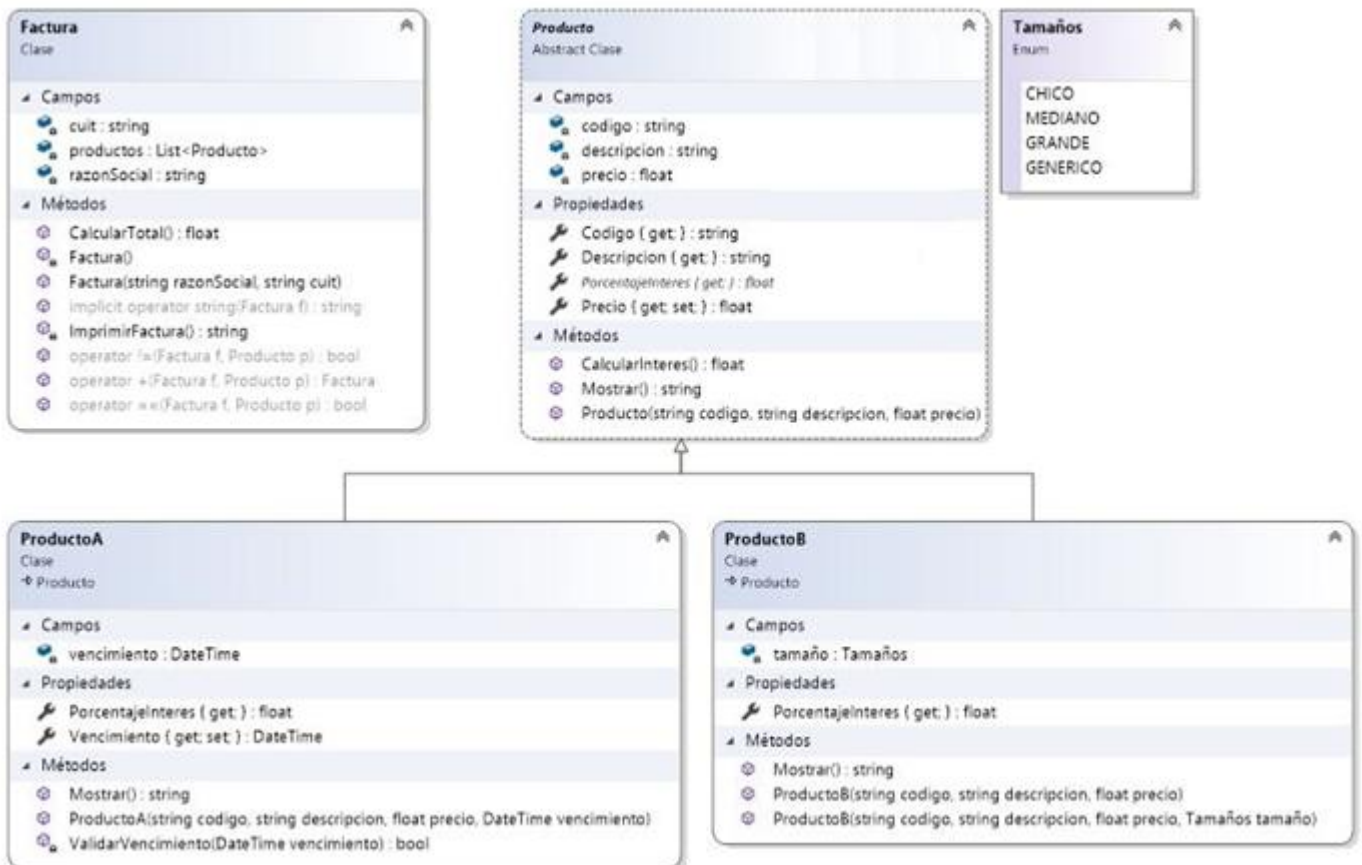
        f += p6;
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }

    Console.WriteLine((string)f);

    Console.ReadKey();

```

4. Dentro de Entidades, diagramar las siguientes clases:



5. Tener en cuenta:
  - a. *Mostrar* retornará los datos de la clase dónde se lo coloque, utilizando `StringBuilder` para compilar dicha información.
6. Tener en cuenta dentro de **Producto**:
  - a. *Mostrar* tendrá implementación en todas las clases y podrá ser sobreescrita en las clases derivadas. El precio del producto se deberá mostrar con el interés incluido, también se deberá mostrar el porcentaje de interés aplicado.
  - b. La **propiedad** *PorcentajeInteres* no tendrá implementación dentro de **Producto**.
  - c. *CalcularInteres* calculará y retornará el **monto** del interés que se aplicará al producto.
7. Tener en cuenta dentro de **ProductoA**:
  - a. La **propiedad** *PorcentajeInteres* retornará un porcentaje del 21% de interés.
  - b. *ValidarVencimiento* validará que el vencimiento sea posterior a 30 días a partir de la fecha actual.
  - c. La propiedad *Vencimiento* sólo asignará vencimientos válidos, caso contrario lanzará una excepción genérica con el mensaje "Vencimiento inválido"
8. Tener en cuenta dentro de **ProductoB**:
  - a. En caso de no indicarse un tamaño, se tomará por defecto el genérico.
  - b. *PorcentajeInteres* retornará un 5% para productos chicos, un 10% para medianos, un 15% para grandes y un 21% para genéricos.

9. Tener en cuenta dentro de Factura:
  - a. El **constructor privado** será el único lugar donde se instanciará la lista de productos.
  - b. El **operador explícito** retornará los datos de la factura y todos sus productos, utilizando StringBuilder para compilar dicha información.
  - c. El **operador ==** entre Factura y Producto informará true si ya hay un producto cargado en la factura con el mismo código.
  - d. El **operador +** entre Factura y Producto agregará el producto a la factura siempre y cuando no haya uno cargado con su mismo código.
10. Tamaños tendrá su propio archivo, siendo también parte del namespace Entidades.
11. Por último, generar el siguiente **formulario** dentro de *VistaForm*:

12. Siendo los elementos a utilizar GroupBox, Button, ComboBox, RichTextBox, Label, DateTimePicker y TextBox.
13. El formulario tendrá un atributo del tipo Factura el cual se instanciará en el evento Load del mismo.
14. Con el botón **btnImprimir** se mostrará en el RichTextBox **rtbFactura** todos los datos de la Factura.
15. Para agregar productos a la factura del formulario se utilizarán los botones **btnAgregarA** y **btnAgregarB**.
16. Se deberá mostrar el mensaje “Vencimiento inválido” al usuario al intentar agregar un productoA con un vencimiento anterior a 30 días a partir de la fecha actual.
17. Para cargar el ComboBox **cmbTamaño** utilizar el siguiente código:

```
cmbTamano.DataSource = Enum.GetValues(typeof(Tamaños));
```

18. Para leer el elemento enumerado del combo, utilizar el siguiente código:

```
Tamaños tam;  
Enum.TryParse<Tamaños>(cmbTamaño.SelectedValue.ToString(), out tam);
```

19. Respetar los nombres de todos los elementos.
20. El diagrama de clases del formulario será:



Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP que deberá tener como nombre Apellido.Nombre.division.zip y dejar este último en el Escritorio de la máquina.

Luego presionar el botón  de la barra superior, colocar un mensaje y apretar **Aceptar**.

Finalmente retirarse del aula y aguardar por la corrección.