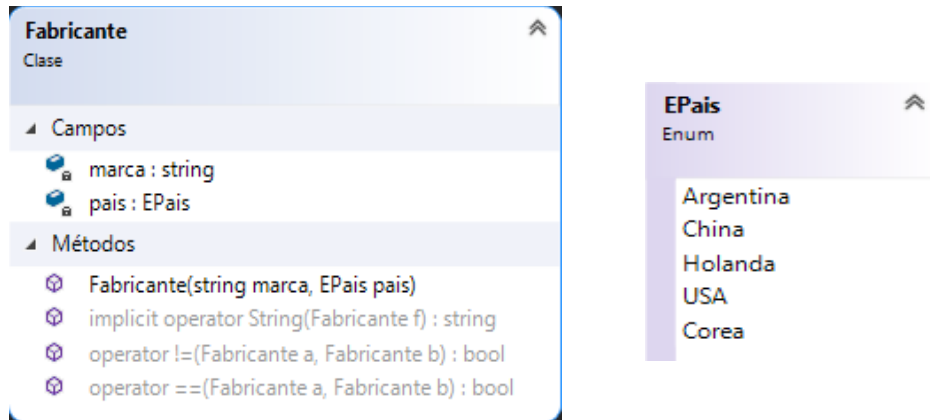


Recuperatorio del PRIMER PARCIAL – LABORATORIO II – 2021

Generar una Solución nombrada como: **Apellido.Nombre.Division**, que contenga un proyecto de tipo *Biblioteca de Clases* (Entidades) el cual tendrá las siguientes clases:

Fabricante posee todos sus atributos privados, un único constructor y sobrecargas de operadores: Igualdad (Fabricante, Fabricante). Retornará *true*, si las marcas y los países son iguales, *false*, caso contrario. Implícito. Retornará la marca y el país del fabricante que recibe como parámetro separado por un guión.



Clase abstracta **Producto**:

Todos sus atributos son protegidos.

Posee un constructor de **clase** y un constructor de instancia sobrecargado. Reutilizar código.

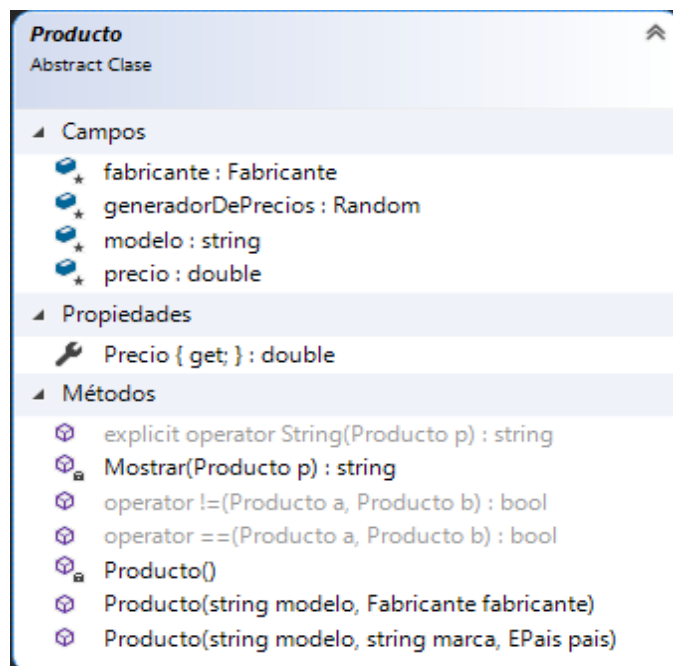
La propiedad (de sólo lectura) `Precio`, retornará el valor correspondiente del atributo `precio`, que se inicializará en dicha propiedad, si y sólo si su valor es cero. Para inicializar dicho atributo, se utilizará el atributo estático `generadorDePrecios` (valores aleatorios entre 10500 y 125000). Inicializarlo dónde corresponda.

El método privado y de **clase** `Mostrar`, retornará una cadena detallando los atributos del objeto que recibe cómo parámetro. Reutilizar código.

Sobrecarga de operadores:

Igualdad (`Producto`, `Producto`). Retornará *true*, si los modelos y los fabricantes son iguales, *false*, caso contrario. Reutilizar código.

Explícito. Retornará el detalle completo del producto que recibe como parámetro. Reutilizar código.



También tendrá las siguientes clases derivadas de Producto:

Televisor posee un único atributo público, que será inicializado por su **único** constructor.

Sobrecarga de operadores:

Igualdad (Televisor, Televisor). Retornará *true*, si los productos y los tipos son iguales, *false*, caso contrario. Reutilizar código.

Explícito. Retornará el precio del televisor que recibe como parámetro. Reutilizar código.

Polimorfismo en Equals. Retornará true, si el parámetro recibido es igual a la instancia actual. Reutilizar código.

Polimorfismo en ToString, retornará una cadena conteniendo la información completa del objeto. Reutilizar código.

The screenshot displays the 'Televisor' class in an IDE. The class is a 'Clase' (Class) that inherits from 'Producto'. It has a single public field 'tipo : ETipo'. The methods listed are: 'Equals(object obj) : bool', 'explicit operator Double(Televisor tv) : double', 'operator !=(Televisor a, Televisor b) : bool', 'operator ==(Televisor a, Televisor b) : bool', 'Televisor(string modelo, string marca, EPais pais, ETipo tipo)', and 'ToString() : string'. To the right, the 'ETipo' enum is shown with three values: 'Tubo', 'Plasma', and 'Led'.

Celular posee un único atributo propio, que será inicializado por su **único** constructor.

Sobrecarga de operadores:

Igualdad (Celular, Celular). Retornará *true*, si los productos y las gamas son iguales, *false*, caso contrario. Reutilizar código.

Implícito. Retornará el precio del celular que recibe como parámetro.

Polimorfismo en Equals. Retornará true, si el parámetro recibido es igual a la instancia actual. Reutilizar código.

Polimorfismo en ToString, retornará una cadena conteniendo la información completa del objeto. Reutilizar código.

The screenshot displays the 'Celular' class in an IDE. The class is a 'Clase' (Class) that inherits from 'Producto'. It has a single public field 'gama : EGama'. The methods listed are: 'Celular(string modelo, Fabricante fabricante, EGama gama)', 'Equals(object obj) : bool', 'implicit operator Double(Celular celu) : double', 'operator !=(Celular a, Celular b) : bool', 'operator ==(Celular a, Celular b) : bool', and 'ToString() : string'. To the right, the 'EGama' enum is shown with three values: 'Baja', 'Media', and 'Alta'.

La última clase que tendrá el proyecto será **Deposito**.

Dicha clase posee dos atributos, ambos privados. Uno indicará la capacidad máxima que tendrá el depósito para almacenar productos. El otro es una colección genérica (lista) de tipo Producto.

El constructor y su sobrecarga son privados. El constructor por defecto será el **único** que inicializará la lista genérica. La sobrecarga, inicializará la capacidad del depósito. Reutilizar código.

El método público de **clase** Mostrar, retornará una cadena con toda la información del depósito, incluyendo el detalle (completo) de cada uno de sus productos. Reutilizar código.

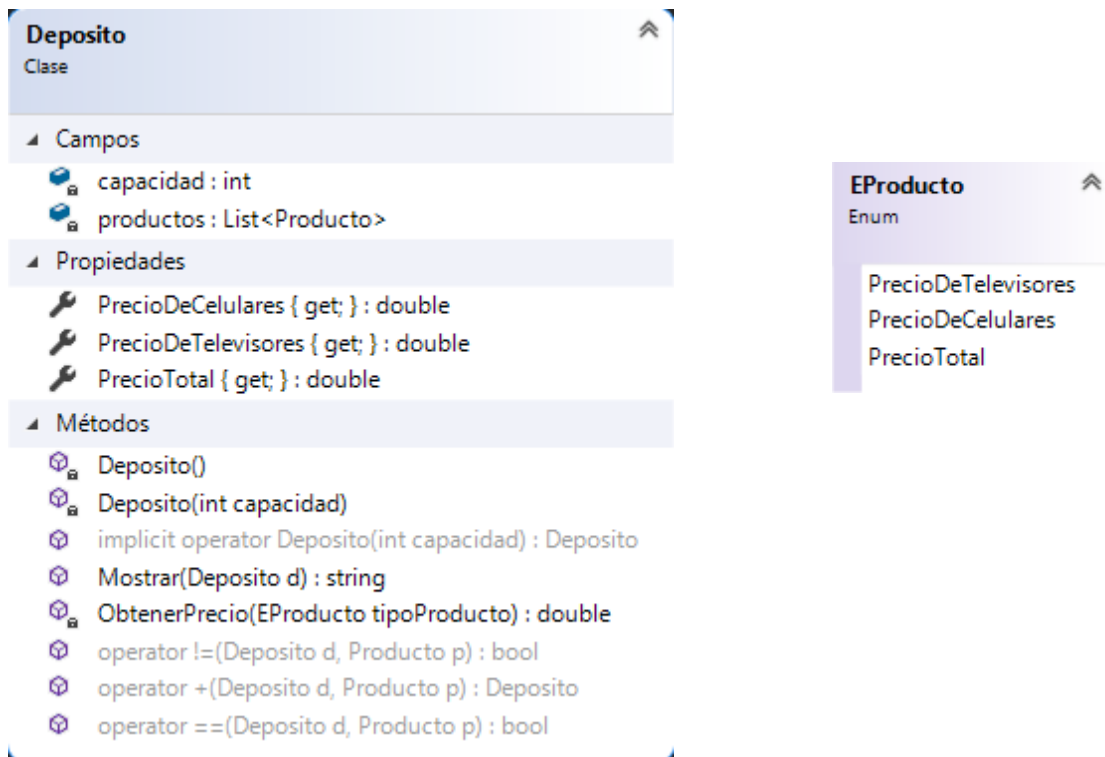
Sobrecarga de operadores:

Implícito, retornará una instancia de tipo Deposito, cuya capacidad coincida con el parámetro recibido.

Igualdad, retornará *true*, si es que el televisor o el celular ya se encuentra en depósito, *false*, caso contrario. Reutilizar código.

Adición, si el depósito posee capacidad de almacenar al menos un producto más y ese televisor o celular no se encuentra en el depósito, lo agregará a la colección, caso contrario, informará lo acontecido. Reutilizar código.

Método privado y de instancia ObtenerPrecio, retornará el valor del depósito de acuerdo con el enumerado **EProducto** que recibe como parámetro. Las propiedades públicas PrecioDeTelevisores, PrecioDeCelulares y PrecioTotal están asociadas al método ObtenerPrecio. Reutilizar código.



Agregar a la solución un proyecto de tipo Aplicación de Consola (Test) y agregar el siguiente Main, sin modificar línea alguna.

NOTA: Los únicos valores que podrán cambiar con respecto a la salida de consola, son los que indican el precio de cada producto, ya que deben ser valores aleatorios de entre 10500 y 125000.

Main:

```
Deposito miDeposito = 7;
Fabricante f1 = new Fabricante("Samsung", EPais.Corea);
Fabricante f2 = new Fabricante("Huawei", EPais.China);

Televisor t1 = new Televisor("LED 4K", "Philips", EPais.Holanda, ETipo.Led);
Televisor t2 = new Televisor("Plasma 2000", "Noblex", EPais.Argentina, ETipo.Plasma);
Televisor t3 = new Televisor("LED 4K", "Philips", EPais.Holanda, ETipo.Tubo);
Televisor t4 = new Televisor("LED 4K", "Philips", EPais.Holanda, ETipo.Led);
Celular c1 = new Celular("A6", f1, EGama.Media);
Celular c2 = new Celular("A20", f2, EGama.Alta);
Celular c3 = new Celular("A20", f2, EGama.Baja);
Celular c4 = new Celular("Iphone 12", new Fabricante("Apple", EPais.USA), EGama.Alta);
Celular c5 = new Celular("A20", f1, EGama.Alta);
```

```
miDeposito += c1;
```

```
//YA INGRESADO
```

```
miDeposito += c1;
```

```
miDeposito += t1;
```

```
miDeposito += t2;
```

```
miDeposito += t3;
```

```
miDeposito += c2;
```

```
miDeposito += c3;
```

```
//YA INGRESADO
```

```
miDeposito += t4;
```

```
miDeposito += c4;
```

```
//SIN LUGAR
```

```
miDeposito += c5;
```

```
Console.WriteLine(Deposito.Mostrar(miDeposito));
```

```
Console.ReadLine();
```

```
El producto ya se encuentra en el depósito!!!
El producto ya se encuentra en el depósito!!!
No hay mas lugar en el depósito!!!
```

```
Capacidad: 7 (7)
Total por televisores: 128296
Total por celulares: 367009
Total: 495305
```

```
*****
Listado de productos
*****
```

```
FABRICANTE: Samsung - Corea
MODELO: A6
PRECIO: 92949
GAMA: Media
```

```
FABRICANTE: Philips - Holanda
MODELO: LED 4K
PRECIO: 27134
TIPO: Led
```

```
FABRICANTE: Noblex - Argentina
MODELO: Plasma 2000
PRECIO: 18162
TIPO: Plasma
```

```
FABRICANTE: Philips - Holanda
MODELO: LED 4K
PRECIO: 83000
TIPO: Tubo
```

```
FABRICANTE: Huawei - China
MODELO: A20
PRECIO: 114828
GAMA: Alta
```

```
FABRICANTE: Huawei - China
MODELO: A20
PRECIO: 39236
GAMA: Baja
```

```
FABRICANTE: Apple - USA
MODELO: Iphone 12
PRECIO: 119996
GAMA: Alta
```