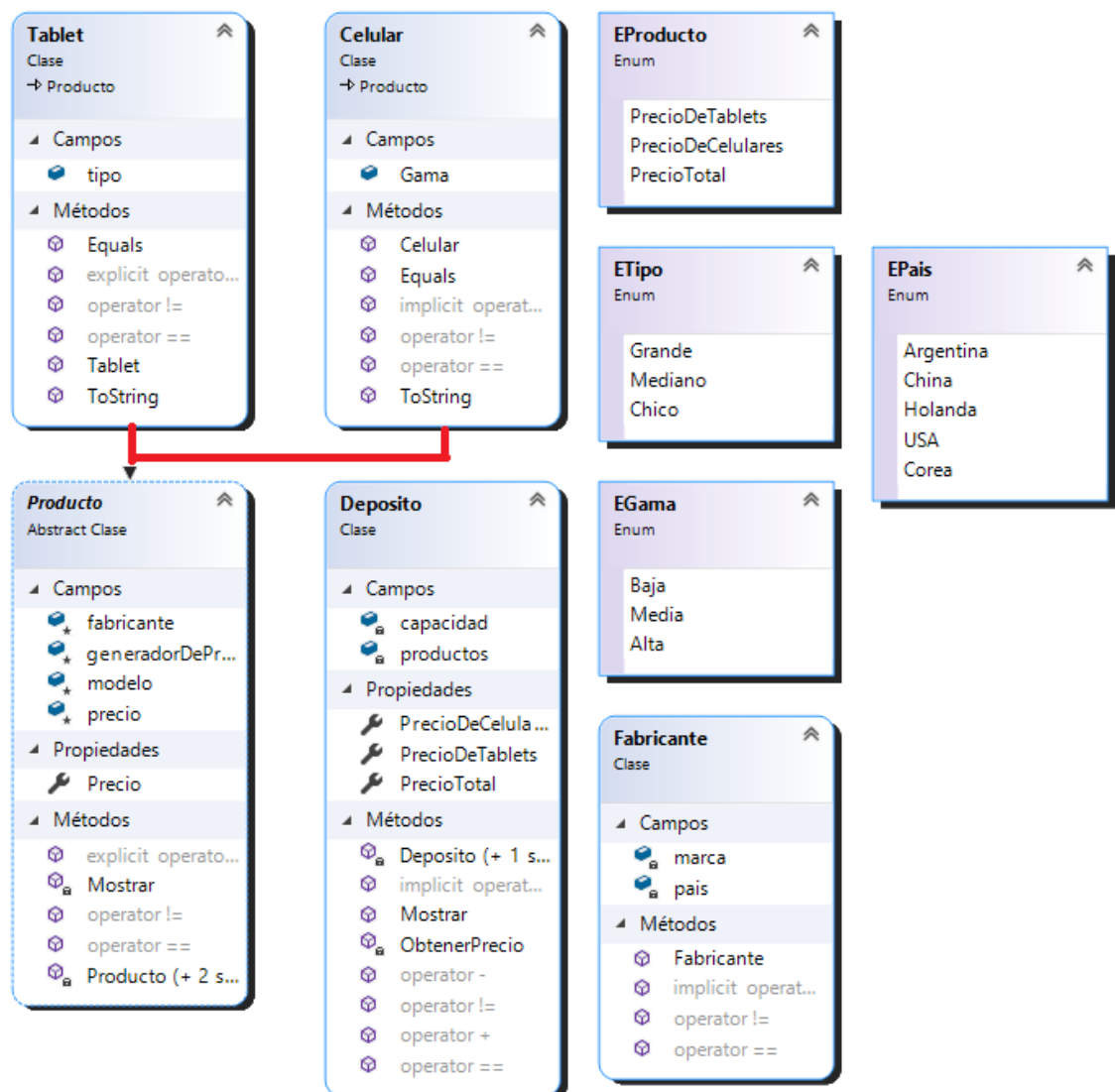


• Diagrama en Bloques.



Introducción

Este programa se basa en la venta y fabricación de Celulares y Tablets, cada uno con su gama, lugar de fabricación, tamaño, precio, marca y modelo con un límite del depósito de 200 productos, al hacer mas no dejara más entrar productos. A su vez sus precios varían (con un random que hice) de 8500 a 150000.

Clases

Contemos con varias clases (Tablet, Producto, Depósito, Celular, tipo, gama, Fabricante, país)

P.O.O

Creación de objetos (Tablet y celular), constructor para cada uno.

Sobrecargas

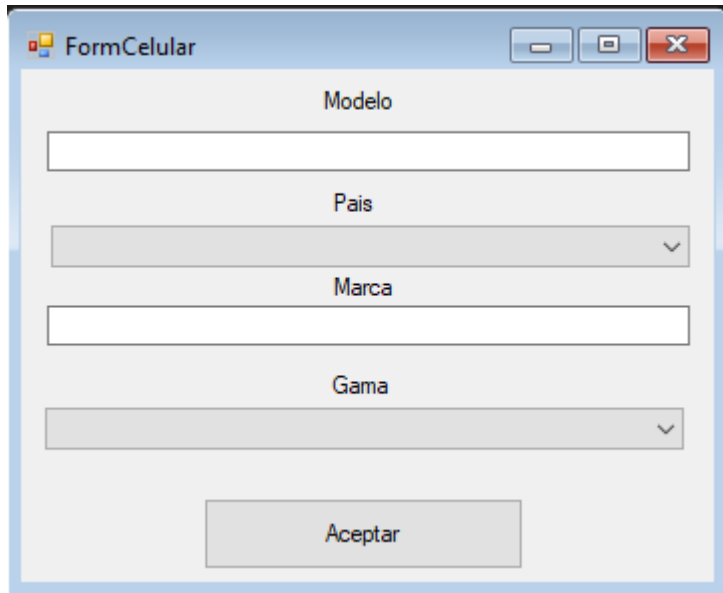
```
3 referencias
public static bool operator ==(Producto p1, Producto p2)
{
    return (p1.modelo == p2.modelo && p1.fabricante == p2.fabricante);
}
0 referencias
public static bool operator !=(Producto p1, Producto p2)
{
    return !(p1 == p2);
}
public static explicit operator string(Producto p)
{
    return Producto.Mostrar(p);
}
#endregion
```

Cada producto tiene su sobrecarga, tanto distinto, igualdad e explícito (también para Tablet y celulares). Ejemplo en Tablet.

```
3 referencias
public static bool operator ==(Tablet t1, Tablet t2)
{
    return (Producto)t1 == (Producto)t2 && t1.tipo == t2.tipo;
}
0 referencias
public static bool operator !=(Tablet t1, Tablet t2)
{
    return !(t1 == t2);
}
public static explicit operator double(Tablet t)
{
    return t.precio;
}
0 referencias
public override bool Equals(object obj)
{
    return ((Tablet)obj) == this;
}
0 referencias
public override string ToString()
{
    StringBuilder sb = new StringBuilder();
    sb.AppendLine("Tablet");
    sb.AppendFormat($"{{(string)(Producto)this}}");
    sb.AppendFormat($"TIPO: {this.tipo}\n");
}
```

Windows Forms

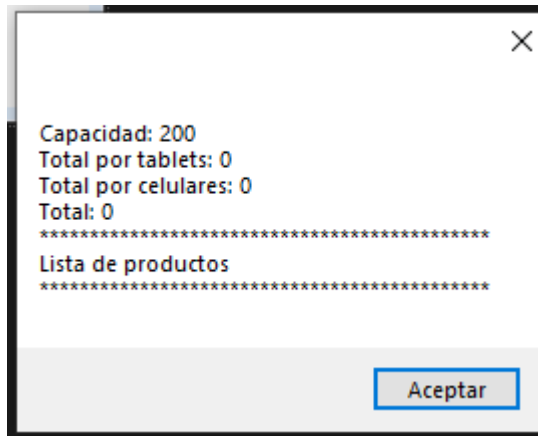
Hay tres tanto para el menú: Contiene Tablet, Celulares (para fabricar la Tablet o Celular) y el muestreo que muestra los productos que se hicieron.



The screenshot shows a Windows Form titled "FormCelular". It contains four input fields: "Modelo" (a text box), "Pais" (a dropdown menu), "Marca" (a text box), and "Gama" (a dropdown menu). Below these fields is a button labeled "Aceptar".

Un ejemplo a la hora de fabricar uno de los dos productos.

A la hora de mostrar.



The screenshot shows a Windows Form with a close button (X) in the top right corner. The form displays the following text:

```
Capacidad: 200
Total por tablets: 0
Total por celulares: 0
Total: 0
*****
Lista de productos
*****
```

At the bottom right of the form is a button labeled "Aceptar".

Colecciones

```
1 referencia
private Deposito()
{
    this.productos = new List<Producto>();
}
```

Lo utilizo para generar la lista nueva de productos y evitar que sea nula.

Encapsulamiento

```
1 referencia
public double PrecioDeTablets
{
    get
    {
        return this.ObtenerPrecio(EProducto.PrecioDeTablets);
    }
}
1 referencia
```

Lo utilice para obtener los precios de cada producto.

Herencia

Heredo de Producto, Celulares y Tablet
Ejemplo.

```
public class Tablet:Producto
```

Cada clase que hereda de productos tiene su propio atributo.

Abstracto

Uso como Producto clase abstracta porque de esta deriva dos clases, ósea Tablet y Celulares.

Polimorfismo

Hace una comparación de dos objetos de la misma clase. Un ejemplo con el Equals de Tablet.

```
0 referencias
public override bool Equals(object obj)
{
    return ((Tablet)obj) == this;
}
```