

```
o public abstract class Producto : IComparable {
```

```
    protected double precioBase;
```

```
    protected double largo;
```

```
    private int codigo;
```

```
    public int Codigo {
```

```
        get {
```

```
            return codigo;
```

```
        } set { value = null;
```

```
    }  
  
    public Producto(double precio, double largo) {
```

```
        this.precioBase = precio;
```

```
    } this.largo = largo;
```

```
    public abstract double Peso();
```

```
    public abstract double Precio();
```

```
    public int CompareTo(object obj) {
```

```
        Producto nuevoProducto = obj as Producto;
```

```
        if (nuevoProducto != null) {
```

```
            return this.Codigo.CompareTo(nuevoProducto.Codigo);
```

```
        } return -1;
```

```
    }  
}
```

```

0 public class Banco : Producto {
    public Banco (double precio, double largo) : base (precio, largo) { }
    public override double Peso() {
        return (this.largo * 0.25) * 0.42;
    }
    public override double Precio() {
        return this.Peso() * this.precioBase * 1.15;
    }
    public override string ToString() {
        return $" {this.Codigo} : Banco de {this.largo:F2} ";
    }
}

```

```

0 public class Mesa : Producto {
    private double ancho;
    private double grosor;
    public Mesa (double precio, double largo, double ancho, double grosor) : base (precio, largo) {
        this.ancho = ancho;
        this.grosor = grosor;
    }
    public override double Peso() {
        return (this.largo * this.ancho * this.grosor) * 0.3;
    }
    public override double Precio() {
        return Peso() * this.precioBase * 1.25;
    }
    public override string ToString() {
        return $" {this.Codigo} : Mesa de {this.largo:F2} x {this.ancho:F2} ";
    }
}

```



```
o public class Cliente {  
    private String nombre;  
    private String direccion;  
    public Cliente (String nombre, String direccion) {  
        this.nombre = nombre;  
        this.direccion = direccion;  
    }  
    public override String ToString() {  
        return "Nombre: {this.nombre} - Dirección: {this.direccion}";  
    }  
}
```

```
0 public class Presupuesto {
```

```
    . public double Precio {
```

```
        private double total = 0;
```

```
        get { double total = 0;
```

```
            foreach (Producto p in listaProductos) {
```

```
                total += p.Precio();
```

```
            } return total;
```

```
        set { total = value; }
```

```
    . private List<Producto> listaProductos = new List<Producto>();
```

```
    . private Cliente solicitante;
```

```
    public Presupuesto (string nombre, string direccion) {
```

```
        solicitante = new Cliente (nombre, direccion);
```

```
    }
```

```
    . public void AgregarProducto (Producto unProducto) {
```

```
        listaProductos.Add (unProducto);
```

```
    . public bool QuitarProducto (int codigo) {
```

```
        bool ret = false;
```

```
        Producto producto = BuscarProducto (codigo);
```

```
        if (producto != null) {
```

```
            listaProductos.Remove (producto);
```

```
            ret = true;
```

```
        } return ret;
```

```
    }
```



```

private Producto BuscarProducto(int codigo) {
    listaProductos.Sort();
    Producto banco = new Banco(0,0);
    banco.Codigo = codigo;
    int idx = listaProductos.BinarySearch(banco);
    Producto p = null;
    if (idx > -1) {
        p = listaProductos[idx]; // es Producto
    }
    return p;
}

```

```

public string[] Resumen() {
    string[] resumen = new string[listaProductos.Count + 2];
    int i = 0;
    resumen[i++] = solicitante.ToString();
    foreach (Producto p in listaProductos) {
        resumen[i++] = $" {p.ToString()} - Precio: ${p.Precio:F2}";
    }
    resumen[i++] = $" Precio Final: ${p.Precio:F2}";
    return resumen;
}

```

```
o public class Form1 {
```

```
Presupuesto presupuesto;  
Cliente cliente;
```

```
• private void btnIniciarPresupuesto_Click (-) {
```

```
string nombre = tbNombre.Text;
```

```
string direccion = tbDir.Text;
```

```
presupuesto = new Presupuesto (nombre, direccion);
```

```
# region Llamado
```

```
tbDir.Enabled = false;
```

```
tbNombre.Enabled = false;
```

```
tbNombre.Enabled = false;
```

```
tbDir.Enabled = false;
```

```
# endregion
```

```
}
```

```
• private void btnAgregar_Click (-) {
```

```
double precio = Convert.ToDouble (tbPrecio.Text);
```

```
int codigo = Convert.ToInt32 (tbCodigo.Text);
```

```
double largo = Convert.ToDouble (tbLargo.Text);
```

```
if (tbAncho.Text != "" && tbGrosor.Text != "") {
```

```
double ancho = Convert.ToDouble (tbAncho.Text);
```

```
double grosor = Convert.ToDouble (tbGrosor.Text);
```

```
Producto p = new Mesa (precio, largo, ancho, grosor);
```

```
{ p.Codigo = codigo;
```

```
else {
```

```
Producto p = new Banco (precio, largo);
```

```
{ p.Codigo = codigo;
```

```
if (presupuesto != null) {
```

```
presupuesto.AgregarProducto (p);
```

```
cmbProductos.Items.Add (p); // Tiene el ToString() Ej: "101: Mesa de 200x100"
```

```
cmbProductos.Text = "";
```

```
}
```

```
else {
```

```
MessageBox.Show ("Debes iniciar un presupuesto con los datos del cliente");
```

```
}
```

```
}
```



```

private void btnBorrar_Click (--) {
    if (presupuesto != null) {
        Productos pSeleccionado = cmbProductos.SelectedItem as Producto;
        if (pSeleccionado != null) {
            presupuesto.QuitarProducto(pSeleccionado.Codigo);
            cmbProductos.Remove(pSeleccionado);
            cmbProductos.Text = "";
        }
        else {
            MessageBox.Show("Debes seleccionar un producto del Combo box");
        }
    }
    else {
        MessageBox.Show("Debes iniciar un Presupuesto");
    }
}

```

```

private void btnCerrarPresupuesto_Click (--) {
    if (presupuesto != null) {
        Form2 FPresupuesto = new Form2();
        FPresupuesto.IsResultado.Items.Add(Ridge(presupuesto.Resumen()));
        FPresupuesto.ShowDialog();
        FPresupuesto.Dispose();
    }
    else {
        MessageBox.Show("Debes iniciar un Presupuesto");
    }
}

```