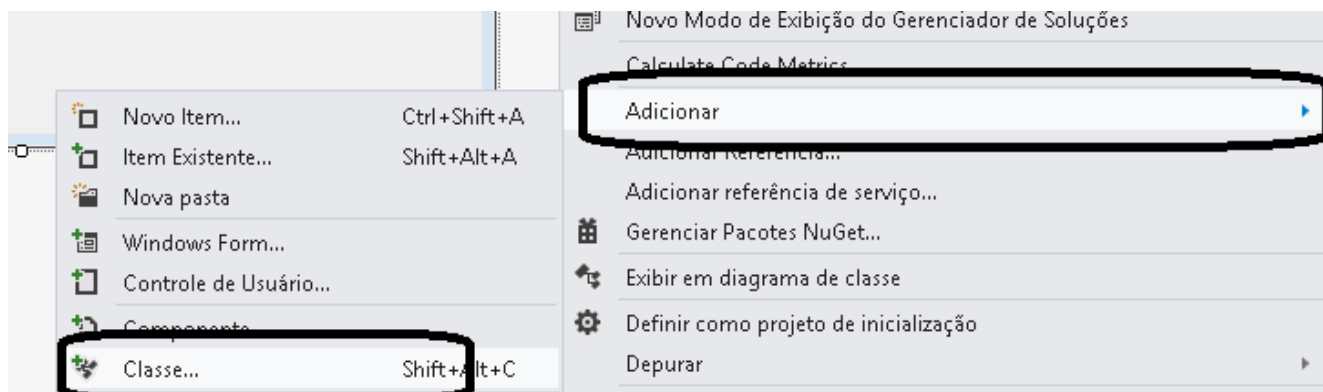


REVISÃO: LISTAS DE OBJETOS

Em nosso exemplo, vamos modelar a seguinte classe que controle um sistema de carrinho de compras. Em nosso caso iniciaremos coma modelagem da classe Produto.



Codifique a classe:

```
class Produto
{
    private int codigo;
    private string nome;
    private double precoUnitario;
}
```

Crie o construtor:

```
class Produto
{
    private int codigo;
    private string nome;
    private double precoUnitario;

    public Produto(int codigo, string nome, double precoUnitario)
    {
        this.codigo = codigo;
        this.nome = nome;
        this.precoUnitario = precoUnitario;
    }
}
```

Criar os encapsulamentos:

```

class Produto
{
    private int codigo;

    public int Codigo
    {
        get { return codigo; }
        set { codigo = value; }
    }

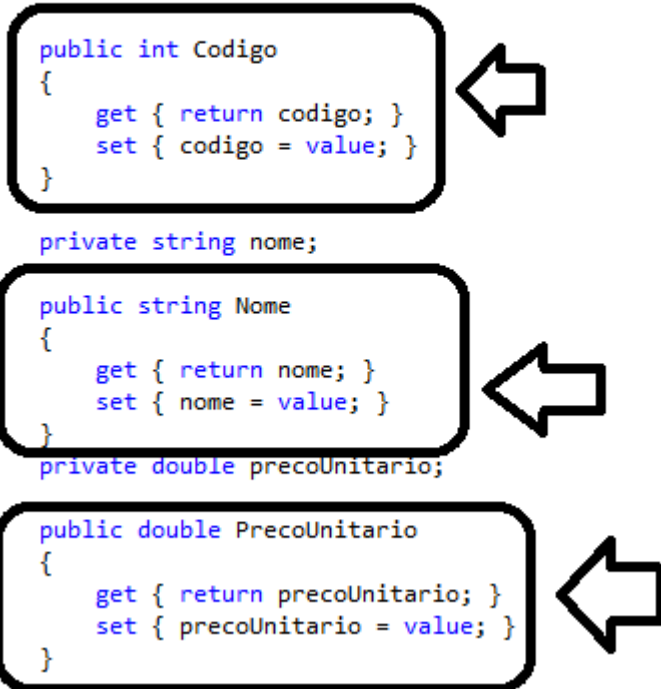
    private string nome;

    public string Nome
    {
        get { return nome; }
        set { nome = value; }
    }

    private double precoUnitario;

    public double PrecoUnitario
    {
        get { return precoUnitario; }
        set { precoUnitario = value; }
    }
}

```



Construir a classe carrinho:

```

class Carrinho
{
    private BindingList<Produto> produtos;

    internal BindingList<Produto> Produtos
    {
        get { return produtos; }
        set { produtos = value; }
    }

    private string cliente;

    public string Cliente
    {
        get { return cliente; }
        set { cliente = value; }
    }

    public Carrinho()
    {
        Produtos = new BindingList<Produto>();
        Cliente = "";
    }
}

```

Considera acrescentar os métodos de adicionar e remover produtos do carrinho:

```

public Carrinho()
{
    Produtos = new BindingList<Produto>();
    Cliente = "";
}

public void adicionar(Produto p) {
    Produtos.Add(p);
}

public void remover(Produto p)
{
    Produtos.Remove(p);
}
}

```

Desenhar agora o formulário:

The image shows a Windows application window titled "Carrinho - Merceria". Inside the window, there is a label "Código do Produto:" followed by a text input field containing the number "0". To the right of the input field are two buttons: "Adicionar" (represented by a shopping basket icon with a green plus sign) and "Remover" (represented by a shopping basket icon with a red minus sign). Below these buttons is a large, empty gray rectangular area. At the bottom of the window, there is a white rectangular box containing the text "TOTAL: R\$0,00" in a large, bold, black font.

Nome dos campos:

```
.BindingSource bs
.DataGridView dgLista
.Label lbPrecoTotal
.Button btnAdd
.Label label1
.TextBox txtCodigo
.Button btnDel
```

Criar o carrinho no formulário, a lista de produtos padrão e relaciona-las no **bindingSource** e **datagridview no evento load**:

```
public partial class Form1 : Form
{
    // crio o carrinho:
    Carrinho carrinho = new Carrinho();

    // lista de produtos da mercearia
    BindingList<Produto> lista = new BindingList<Produto>();

    public Form1()
    {
        InitializeComponent();
    }
}
```

Crie o evento load:

```
private void Form1_Load(object sender, EventArgs e)
{
    // lista de Produtos da mercearia - exemplo
    lista.Add(new Produto(1, "PCT ARROZ 1 KG", 4.3));
    lista.Add(new Produto(2, "VINHO SUAVE CHAPINHA 1 LT", 5.78));
    lista.Add(new Produto(3, "FARINHA TRIGO TIPO 2 1 KG", 7.8));
    lista.Add(new Produto(4, "LEITE LONGA VIDA DESNATADO 1 LT", 8.9));
    lista.Add(new Produto(5, "SOBRECOXA FRANGO 1 KG", 12.7));
    lista.Add(new Produto(6, "COCA COLA 2 LT", 4.5));
    lista.Add(new Produto(7, "MACA ARGENTINA PCT", 3.2));

    bs.DataSource = carrinho.Produtos;
    dgLista.DataSource = bs;
}
```

Codifique o evento do botão de adicionar itens ao carrinho de compras:

```
private void btnAdd_Click(object sender, EventArgs e)
{
    // código do produto
    int cod = Int16.Parse(txtCodigo.Text);

    // procura o pedido na lista
    Produto p = (from item in lista
                  where item.Codigo == cod
                  select item).FirstOrDefault();

    // achou?
    if (p != null)
    {
        // poe o produto no carrinho
        carrinho.adicionar(p);
        bs.ResetBindings(false);
    }
    else
        MessageBox.Show("Produto não cadastrado");
}
```

O resultado visual será:

	Codigo	Nome	PrecoUnitario
▶	1	PCT ARROZ 1 KG	4,3
	4	LEITE LONGA VIDA DESNATADO 1 LT	8,9
	5	SOBRECOXA FRANGO 1 KG	12,7

Vamos automatizar agora o totalizador do carrinho de compras, que irá mostrar quantos produtos foram comprados. Crie o método totalizador na classe Carrinho:

```

public void adicionar(Produto p) {
    Produtos.Add(p);
}

public void remover(Produto p)
{
    Produtos.Remove(p);
}

public string TotalCarrinho {

    get {

        return "TOTAL" +
            String.Format("{0:C2}", produtos.Sum(item=> item.PrecoUnitario));
    }

}
}

```

Vincule este método ao label do formulário no evento load:

```

private void Form1_Load(object sender, EventArgs e)
{
    // lista de Produtos da mercearia - exemplo
    lista.Add(new Produto(1, "PCT ARROZ 1 KG", 4.3));
    lista.Add(new Produto(2, "VINHO SUAVE CHAPINHA 1 LT", 5.78));
    lista.Add(new Produto(3, "FARINHA TRIGO TIPO 2 1 KG", 7.8));
    lista.Add(new Produto(4, "LEITE LONGA VIDA DESNATADO 1 LT", 8.9));
    lista.Add(new Produto(5, "SOBRECOXA FRANGO 1 KG", 12.7));
    lista.Add(new Produto(6, "COCA COLA 2 LT", 4.5));
    lista.Add(new Produto(7, "MACA ARGENTINA PCT", 3.2));

    bs.DataSource = carrinho.Produtos;
    dgLista.DataSource = bs;

    lbPrecoTotal.Text = carrinho.TotalCarrinho;
}

```



Faça a mesma coisa no botão adicionar:

```


private void btnAdd_Click(object sender, EventArgs e)
{
    // código do produto
    int cod = Int16.Parse(txtCodigo.Text);

    // procura o pedido na lista
    Produto p = (from item in lista
                 where item.Codigo == cod
                 select item).FirstOrDefault();

    // achou?
    if (p != null)
    {
        // poe o produto no carrinho
        carrinho.adicionar(p);
        bs.ResetBindings(false);
    }
    else
        MessageBox.Show("Produto não cadastrado");

    lbPrecoTotal.Text = carrinho.TotalCarrinho;
}

```



CODIFICANDO O BOTÃO EXCLUIR

Abaixo o código do botão remover:

```

private void btnDel_Click(object sender, EventArgs e)
{
    // se não há registro
    if (bs.Count == 0) {

        // sair
        return;
    }

    // qual é o produto selecionado?

    Produto atual = (Produto) bs.Current;

    // procura o pedido na lista
    Produto p = (from item in lista
                 where item.Codigo == atual.Codigo
                 select item).FirstOrDefault();

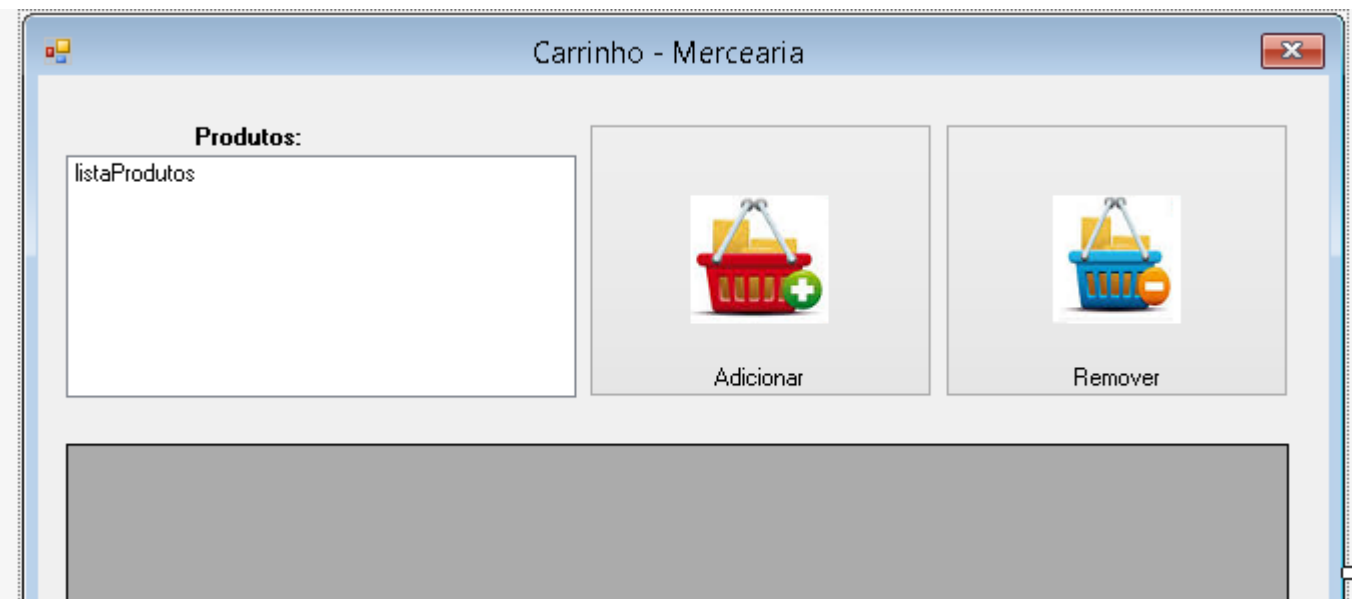
    // achou?
    if (p != null)
    {
        // tira o produto do carrinho
        carrinho.remover(p);
        bs.ResetBindings(false);
    }

    lbPrecoTotal.Text = carrinho.TotalCarrinho;
}

```

Teste o código e verifique a remoção do produto.

O próximo passo é exibir a lista de produtos em um listbox. Para isso vamos excluir o textbox e substituir por um listbox:



Modifique o evento load do formulário:

```
private void Form1_Load(object sender, EventArgs e)
{
    // lista de Produtos da mercearia - exemplo
    lista.Add(new Produto(1, "PCT ARROZ 1 KG", 4.3));
    lista.Add(new Produto(2, "VINHO SUAVE CHAPINHA 1 LT", 5.78));
    lista.Add(new Produto(3, "FARINHA TRIGO TIPO 2 1 KG", 7.8));
    lista.Add(new Produto(4, "LEITE LONGA VIDA DESNATADO 1 LT", 8.9));
    lista.Add(new Produto(5, "SOBRECOXA FRANGO 1 KG", 12.7));
    lista.Add(new Produto(6, "COCA COLA 2 LT", 4.5));
    lista.Add(new Produto(7, "MACA ARGENTINA PCT", 3.2));

    // as três linhas abaixo mostram os produtos vendidos
    listaProdutos.DataSource = lista;
    listaProdutos.DisplayMember = "nome";
    listaProdutos.ValueMember = "codigo";

    bs.DataSource = carrinho.Produtos;
    dgLista.DataSource = bs;

    lbPrecoTotal.Text = carrinho.TotalCarrinho;
}
```

Altere o botão adicionar para que pegue agora o código a partir do item selecionado no listbox:


```
private void btnAdd_Click(object sender, EventArgs e)
{
    // código do produto
    int cod = Int16.Parse(listaProdutos.SelectedValue.ToString());

    // procura o pedido na lista
    Produto p = (from item in lista
                  where item.Codigo == cod
                  select item).FirstOrDefault();

    // achou?
    if (p != null)
    {
        // poe o produto no carrinho
        carrinho.adicionar(p);
        bs.ResetBindings(false);
    }
    else
        MessageBox.Show("Produto não cadastrado");

    lbPrecoTotal.Text = carrinho.TotalCarrinho;
}
}
```



Agora você poderá usar o carrinho selecionando o produto.

Carrinho - Merceria

Produtos:

PCT ARROZ 1 KG
VINHO SUAVE CHAPINHA 1 LT
FARINHA TRIGO TIPO 2 1 KG
LEITE LONGA VIDA DESNATADO 1 LT
SOBRECOXA FRANGO 1 KG
COCA COLA 2 LT
MACA ARGENTINA PCT

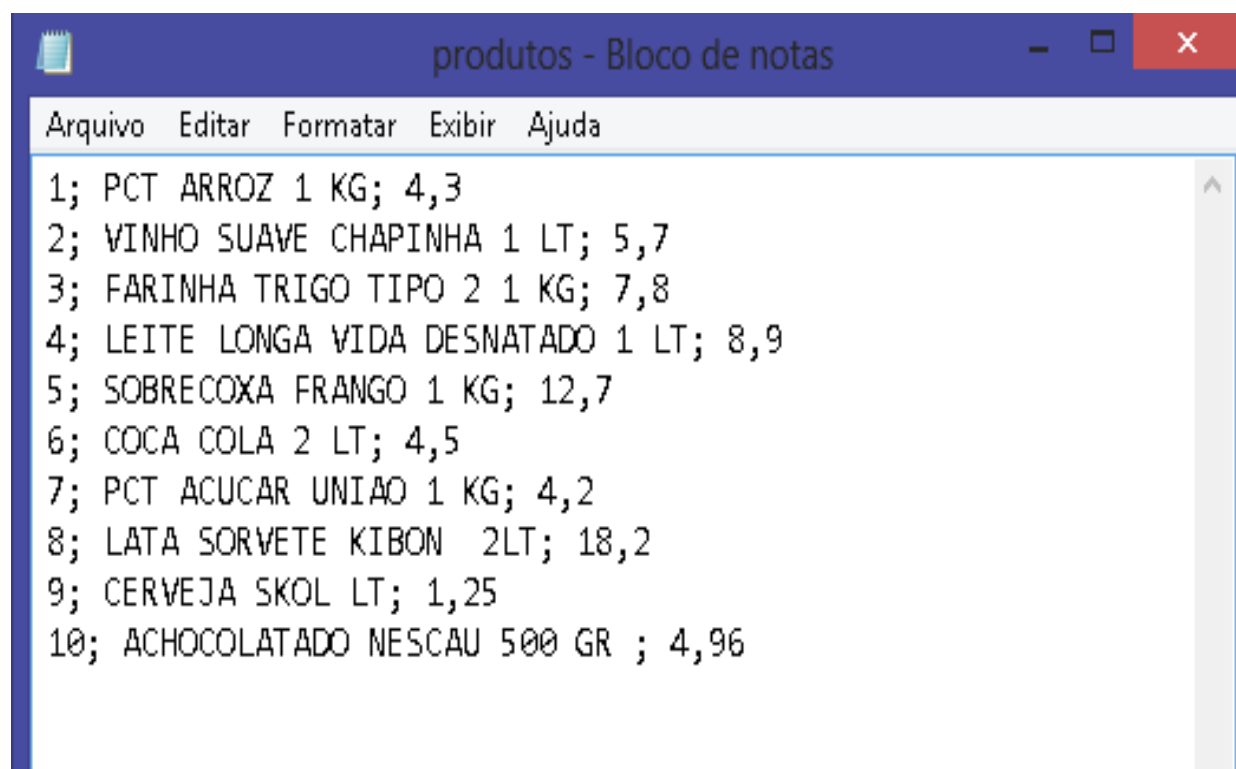
Adicionar

Remover

	Codigo	Nome	PrecoUnitario
	4	LEITE LONGA VIDA DESNATADO 1 LT	8,9
	7	MACA ARGENTINA PCT	3,2
▶	4	LEITE LONGA VIDA DESNATADO 1 LT	8,9
	5	SOBRECOXA FRANGO 1 KG	12,7
	7	MACA ARGENTINA PCT	3,2

TOTALR\$ 36,90

Nosso projeto lê os dados da lista considerando os produtos adicionados no código. O passo seguinte é permitir que a lista de produtos seja lida em um arquivo texto. Em nosso exemplo, crie um arquivo texto chamado **“produtos.txt”** com o seguinte conteúdo:



Perceba que os valores correspondem ao campos da nossa classe produto. Nossa intenção é ler os dados presentes no arquivo e preenche-los. Para isso separamos os dados com ponto e vírgula para facilitar a leitura do arquivo. Modifique a classe load do formulário conforme a seguir:

```
private void Form1_Load(object sender, EventArgs e)
{
    // lista de Produtos da mercearia - exemplo
    preencherLista();

    // as três linhas abaixo mostram os produtos vendidos
    listaProdutos.DataSource = lista;
    listaProdutos.DisplayMember = "nome";
    listaProdutos.ValueMember = "codigo";

    bs.DataSource = carrinho.Produtos;
    dgLista.DataSource = bs;

    lbPrecoTotal.Text = carrinho.TotalCarrinho;
}
```

Criar a rotina preencher lista:

```

private void preencherLista()
{
    // abre o arquivo para leitura
    System.IO.StreamReader arquivo = new System.IO.StreamReader(@"D:\produtos.txt");

    // enquanto não é fim de arquivo
    while (arquivo.Peek() >= 0) {
        //Le a linha e passa para a seguinte
        string linha = arquivo.ReadLine();
        // separa os campos e coloca em uma matriz
        string[] campos = linha.Split(';');
        // joga os valor dos campos no objeto p
        Produto p = new Produto(Int16.Parse(campos[0]),
                                campos[1],
                                Double.Parse(campos[2])
                                );

        // adiciona o produto a lista
        lista.Add(p);
    }

    arquivo.Close();
    // fecha o arquivo
}
}

```

O código acima percorre cada linha do arquivo e coloca os valores que estão separados por ponto e vírgula em uma matriz de campos. Criamos um objeto p da classe produto para receber os dados do produto, ou seja, seu código, nome e preço. No fim adicionamos o objeto p para a lista que será exibida pelo listBox. No fim fechamos o arquivo para leitura. O resultado será:

