



Log In / Cadastre-se

Pesquisar

[HOME](#) [DESENVOLVIMENTO](#) [FRONT-END](#) [BANCO DE DADOS](#) [EM DESTAQUE](#) [TODOS](#)[PUBLIQUE](#)

## Desenvolvimento - C#

# Guia prático para o desenvolvimento de Aplicações C# em Camadas - parte 7

Este é o sétimo artigo da série onde vamos demonstrar passo-a-passo a construção de uma aplicação .Net utilizando o conceito de desenvolvimento em camadas.

por *Carlos Camacho*



## 7 – User Interface – Formulário de Produtos

Neste artigo vamos criar o Formulário de Produtos do nosso projeto.

- Abra o Microsoft Visual Studio;
- Clique em **File > Open > Project/Solution...**
- Na janela *Open Project*, selecione o arquivo da nossa Solution (**C:\LojaModelos\Loja.sln**) e clique em **Open** para abrir a solução.

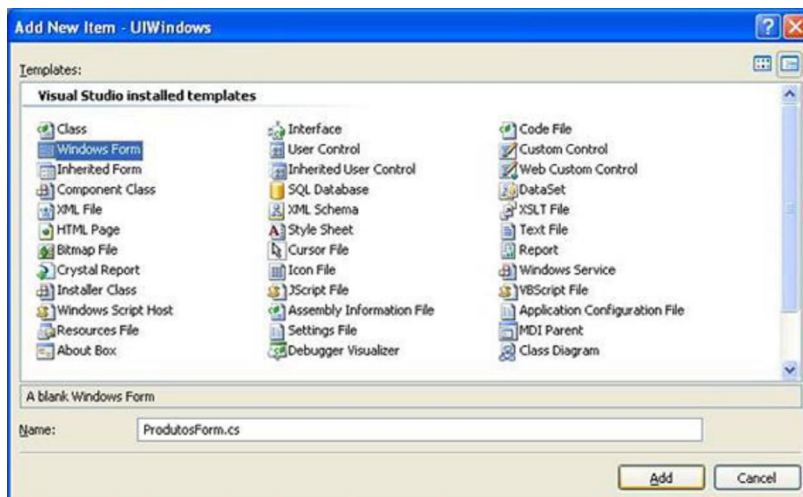
Vamos adicionar um formulário chamado **ProdutosForm.cs**.

- Clique com o botão direito no projeto **UIWindows** e escolha **Add > Windows Form...**

- Na janela "Add New Item" vamos informar os seguintes dados:

Template: **Windows Form**

Name: **ProdutosForm**



- Clique em **Add** para adicionar o formulário.

Será exibido o formulário **ProdutosForm**:



Publicidade

**Compuware dynaTrace**  
É muito simples monitorar **100%** das transações do seu App

Experimente a  
Versão **FREE TRIAL**

Compuware  
**APM**

### REVISTAS DEVMEDIA



.net Mag 116



Easy .net mag 37

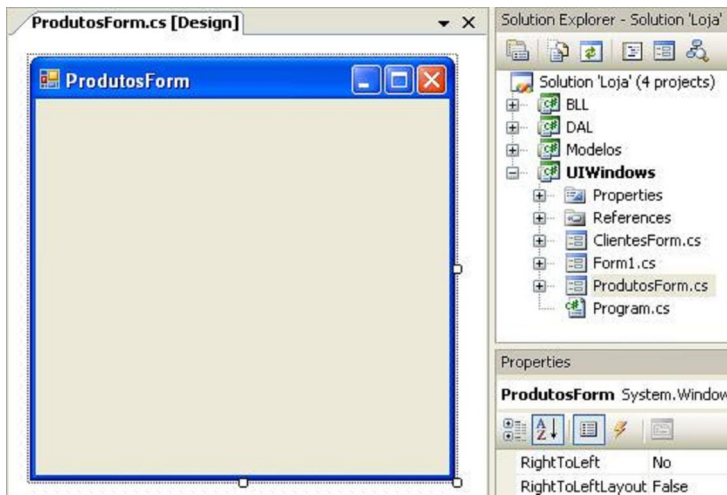
[VER TODAS](#)[ASSINE](#)

### TOP 10 - ARTIGOS

### TOP 10 - AUTORES

- 1 HTML Básico
- 2 Menu em CSS - Menu dropdown horizontal com HTML5 e CSS3
- 3 Comandos básicos em SQL - insert, update, delete e select
- 4 Criando um sistema de cadastro e login com PHP e MySQL
- 5 Código para background HTML e CSS
- 6 Copiando dados com o Robocopy
- 7 Trabalhando com Div em HTML
- 8 Excel: Como verificar se existe valores duplicados
- 9 Botão com CSS 3: Como criar um botão sem imagens
- 10 HTML Avançado

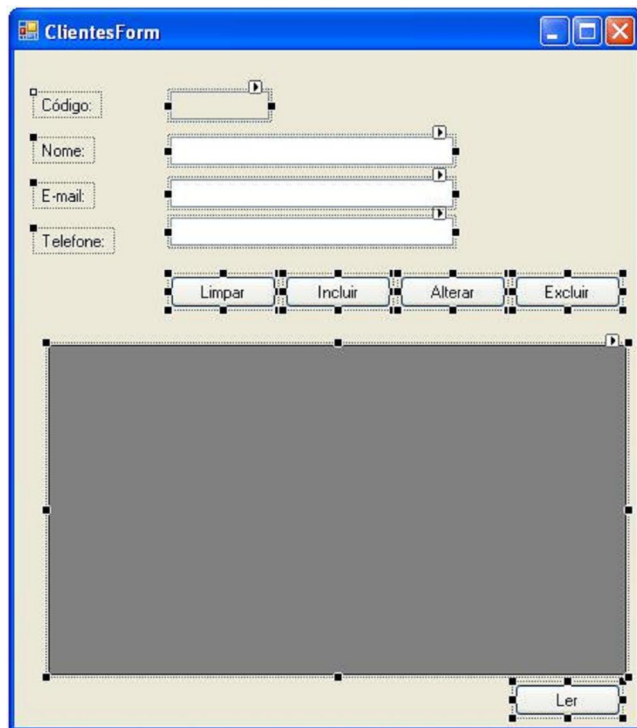
[VER TODOS](#)



- Altere a propriedade **Size** para **448; 512**

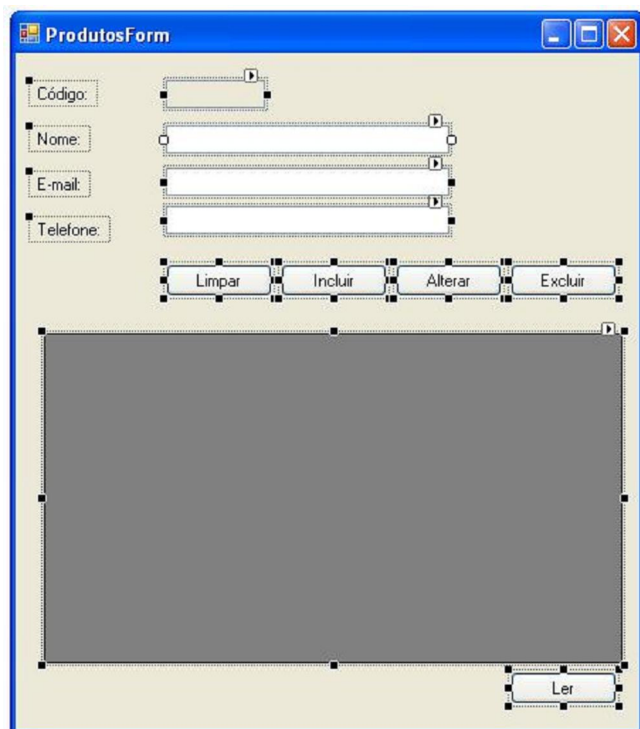
Como as interfaces são semelhantes, vamos copiar os objetos do formulário de clientes para o formulário de produtos.

- Abra o formulário **ClientesForm** no modo de design e digite **<Ctrl> + <A>** para selecionar todos os objetos;



- Digite **<Ctrl> + <C>** para copiar os objetos para a área de transferência;

- Abra o formulário **ProdutosForm** em modo de design e digite **<Ctrl> + <V>** para colar os objetos.



- No objeto **emailLabel** altere os valores das seguintes propriedades:

Name: <b>precoLabel</b>
Text: <b>Preço:</b>

- No objeto **emailTextBox** altere a propriedade **Name** para **precoTextBox**

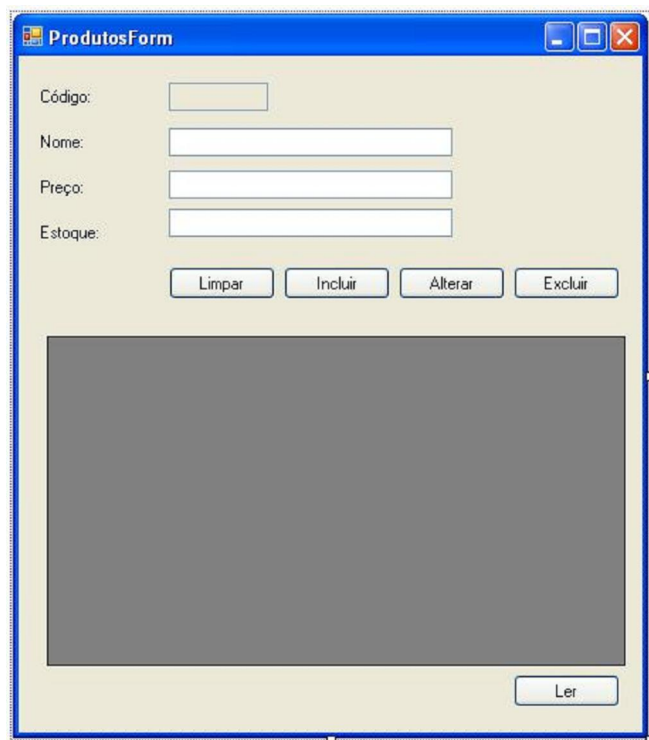
- No objeto **telefoneLabel** altere os valores das seguintes propriedades:

Name: <b>estoqueTextBox</b>
Text: <b>Estoque:</b>

- No objeto **telefoneTextBox** altere a propriedade **Name** para **estoqueTextBox**

- No objeto **clientesDataGridView** altere a propriedade **Name** para **produtosDataGridView**

Neste momento, nosso formulário de produtos estará com esta aparência:



Aqui também vamos criar o método `AtualizaGrid()` para fazer a comunicação com a camada de regras de negócio com o objetivo de preencher o objeto `DataGridView` com a lista de produtos existentes.

Dentro do método `ProdutosForm_Load()` vamos fazer uma chamada para o método `AtualizaGrid()` e também posicionaremos o cursor no primeiro campo do formulário.

- Copie e cole o trecho de código para o code behind do formulário `ProdutosForm`:

```
/*  
  
* File....: ProdutosForm.cs  
  
* Author.: Carlos Camacho - www.camachojunior.com.br  
  
* Content: Formulário de Produtos  
  
* Subject: Guia prático para o desenvolvimento de Aplicações C# em Camadas  
  
*/  
  
using System;  
  
using System.Collections.Generic;  
  
using System.ComponentModel;  
  
using System.Data;  
  
using System.Drawing;  
  
using System.Text;  
  
using System.Windows.Forms;  
  
using Loja.BLL;  
  
using Loja.DAL;  
  
using Loja.Modelos;  
  
namespace Loja.UIWindows  
{  
  
    public partial class ProdutosForm : Form  
  
    {
```

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

public void AtualizaGrid()
{
    // Comunicação com a Camada BLL

    ProdutosBLL obj = new ProdutosBLL();

    produtosDataGridView.DataSource = obj.Listagem();

    // Atualizando os objetos TextBox

    codigoTextBox.Text = produtosDataGridView[0,
produtosDataGridView.CurrentRow.Index].Value.ToString();

    nomeTextBox.Text = produtosDataGridView[1,
produtosDataGridView.CurrentRow.Index].Value.ToString();

    precoTextBox.Text = produtosDataGridView[2,
produtosDataGridView.CurrentRow.Index].Value.ToString();

    estoqueTextBox.Text = produtosDataGridView[3,
produtosDataGridView.CurrentRow.Index].Value.ToString();
}

private void ProdutosForm_Load(object sender, EventArgs e)
{
    AtualizaGrid();

    nomeTextBox.Focus();
}
}
}
```

Agora vamos codificar as funcionalidades dos botões do formulário de produtos.

- Dê um **duplo clique** no **botão Limpar** para codificarmos o evento click;
- Copie e cole o código abaixo:

```
codigoTextBox.Text = "";
nomeTextBox.Text = "";
precoTextBox.Text = "";
estoqueTextBox.Text = "";
```

- Dê um **duplo clique** no botão **Incluir** para codificarmos o evento click;
- Copie e cole o código abaixo:

```
try
{
    ProdutoInformation produto = new ProdutoInformation();

    produto.Nome = nomeTextBox.Text;

    produto.Preco = Convert.ToDecimal(precoTextBox.Text);
```

```
        produto.Estoque = Convert.ToInt32(estoqueTextBox.Text);

        ProdutosBLL obj = new ProdutosBLL();

        obj.Incluir(produto);

        MessageBox.Show("O produto foi incluído com sucesso!");

        codigoTextBox.Text = Convert.ToString(produto.Codigo);
    }

    catch (Exception ex)

    {

        MessageBox.Show("Erro: " + ex.Message);

    }

    AtualizaGrid();
```

- Dê um **duplo clique** no **botão Alterar** para codificarmos o evento click;

- Copie e cole o código abaixo:

```
    if (codigoTextBox.Text == "")

    {

        MessageBox.Show("Um produto precisa ser selecionado para alteração.");

    }

    else

    try

    {

        ProdutoInformation produto = new ProdutoInformation();

        produto.Codigo = int.Parse(codigoTextBox.Text);

        produto.Nome = nomeTextBox.Text;

        produto.Preco = Convert.ToDecimal(precoTextBox.Text);

        produto.Estoque = Convert.ToInt32(estoqueTextBox.Text);

        ProdutosBLL obj = new ProdutosBLL();

        obj.Alterar(produto);

        MessageBox.Show("O produto foi atualizado com sucesso!");

    }

    catch (Exception ex)

    {

        MessageBox.Show("Erro: " + ex.Message);

    }

    AtualizaGrid();
```

- Dê um **duplo clique** no **botão Excluir** para codificarmos o evento click;

- Copie e cole o código abaixo:

```
    if (codigoTextBox.Text.Length == 0)

    {

        MessageBox.Show("Um produto deve ser selecionado antes da exclusão.");

    }
```

```
}  
  
else  
  
    try  
  
    {  
  
        int codigo = Convert.ToInt32(codigoTextBox.Text);  
  
        ProdutosBLL obj = new ProdutosBLL();  
  
        obj.Excluir(codigo);  
  
        MessageBox.Show("O produto foi excluído com sucesso!");  
  
        AtualizaGrid();  
  
    }  
  
    catch (Exception ex)  
  
    {  
  
        MessageBox.Show(ex.Message);  
  
    }  
  
    AtualizaGrid();
```

- Dê um **duplo clique** no **botão Ler** para codificarmos o evento click;

- Copie e cole o código abaixo:

```
AtualizaGrid();
```

- Clique sobre o dataGridView;

- Na janela de propriedades, clique no ícone que tem um relâmpago para exibir a lista de eventos;

- Dê um duplo clique no evento **CellClick** para inserirmos o código para esse evento:

- Dentro do evento **CellClick**, copie e cole o código abaixo:

```
// Atualizando os objetos TextBox
```

```
codigoTextBox.Text = produtosDataGridView[0,  
produtosDataGridView.CurrentRow.Index].Value.ToString();
```

```
nomeTextBox.Text = produtosDataGridView[1,  
produtosDataGridView.CurrentRow.Index].Value.ToString();
```

```
precoTextBox.Text = produtosDataGridView[2,  
produtosDataGridView.CurrentRow.Index].Value.ToString();
```

```
estoqueTextBox.Text = produtosDataGridView[3,  
produtosDataGridView.CurrentRow.Index].Value.ToString();
```

- O code behind completo do nosso ProdutosForm.cs ficou assim:

```
/*  
  
* File...: ProdutosForm.cs  
  
* Author.: Carlos Camacho - www.camachojunior.com.br  
  
* Content: Formulário de Produtos  
  
* Subject: Guia prático para o desenvolvimento de Aplicações C# em Camadas  
  
*/  
  
using System;  
  
using System.Collections.Generic;
```

```
using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using Loja.BLL;

using Loja.DAL;

using Loja.Modelos;

namespace Loja.UIWindows

{

    public partial class ProdutosForm : Form

    {

        public ProdutosForm()

        {

            InitializeComponent();

        }

        public void AtualizaGrid()

        {

            // Comunicação com a Camada BLL

            ProdutosBLL obj = new ProdutosBLL();

            produtosDataGridView.DataSource = obj.Listagem();

            // Atualizando os objetos TextBox

            codigoTextBox.Text = produtosDataGridView[0,

produtosDataGridView.CurrentRow.Index].Value.ToString();

            nomeTextBox.Text = produtosDataGridView[1,

produtosDataGridView.CurrentRow.Index].Value.ToString();

            precoTextBox.Text = produtosDataGridView[2,

produtosDataGridView.CurrentRow.Index].Value.ToString();

            estoqueTextBox.Text = produtosDataGridView[3,

produtosDataGridView.CurrentRow.Index].Value.ToString();

        }

        private void ProdutosForm_Load(object sender, EventArgs e)

        {

            AtualizaGrid();

            nomeTextBox.Focus();

        }

        private void limparButton_Click(object sender, EventArgs e)

        {

            codigoTextBox.Text = "";

            nomeTextBox.Text = "";

        }

    }

}
```



```
precoTextBox.Text = "";

estoqueTextBox.Text = "";

}

private void incluirButton_Click(object sender, EventArgs e)
{
    try
    {
        ProdutoInformation produto = new ProdutoInformation();

        produto.Nome = nomeTextBox.Text;

        produto.Preco = Convert.ToDecimal(precoTextBox.Text);

        produto.Estoque = Convert.ToInt32(estoqueTextBox.Text);

        ProdutosBLL obj = new ProdutosBLL();

        obj.Incluir(produto);

        MessageBox.Show("O produto foi incluído com sucesso!");

        codigoTextBox.Text = Convert.ToString(produto.Codigo);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Erro: " + ex.Message);
    }

    AtualizaGrid();
}

private void alterarButton_Click(object sender, EventArgs e)
{
    if (codigoTextBox.Text == "")
    {
        MessageBox.Show("Um produto precisa ser selecionado para alteração.");
    }
    else
    {
        try
        {
            ProdutoInformation produto = new ProdutoInformation();

            produto.Codigo = int.Parse(codigoTextBox.Text);

            produto.Nome = nomeTextBox.Text;

            produto.Preco = Convert.ToDecimal(precoTextBox.Text);

            produto.Estoque = Convert.ToInt32(estoqueTextBox.Text);

            ProdutosBLL obj = new ProdutosBLL();

            obj.Alterar(produto);
        }
        catch (Exception ex)
        {
            MessageBox.Show("Erro: " + ex.Message);
        }
    }
}
```

```
        MessageBox.Show("O produto foi atualizado com sucesso!");
    }

    catch (Exception ex)
    {
        MessageBox.Show("Erro: " + ex.Message);
    }

    AtualizaGrid();
}

private void excluirButton_Click(object sender, EventArgs e)
{
    if (codigoTextBox.Text.Length == 0)
    {
        MessageBox.Show("Um produto deve ser selecionado antes da exclusão.");
    }
    else
    {
        try
        {
            int codigo = Convert.ToInt32(codigoTextBox.Text);

            ProdutosBLL obj = new ProdutosBLL();

            obj.Excluir(codigo);

            MessageBox.Show("O produto foi excluído com sucesso!");

            AtualizaGrid();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void lerButton_Click(object sender, EventArgs e)
    {
        AtualizaGrid();
    }

    private void produtosDataGridView_CellClick(object sender, DataGridViewCellEventArgs e)
    {
        // Atualizando os objetos TextBox

        codigoTextBox.Text = produtosDataGridView[0,
produtosDataGridView.CurrentRow.Index].Value.ToString();

        nomeTextBox.Text = produtosDataGridView[1,
produtosDataGridView.CurrentRow.Index].Value.ToString();
    }
}
```

```
precoTextBox.Text = produtosDataGridView[2,
produtosDataGridView.CurrentRow.Index].Value.ToString();

estoqueTextBox.Text = produtosDataGridView[3,
produtosDataGridView.CurrentRow.Index].Value.ToString();

}

}

}
```

- Abra o formulário principal Form1 no modo de design e dê um duplo clique na opção produtos do menu para codificarmos a chamada para este formulário com o código abaixo:

```
ProdutosForm obj = new ProdutosForm();

obj.MdiParent = this;

obj.Show();
```

Acabamos de implementar o Formulário de Produtos do nosso projeto Loja .Net.

Execute o projeto e teste as funcionalidades que implementamos neste artigo.

No próximo artigo vamos implementar o formulário de Vendas.

Até o próximo artigo!



Carlos Camacho - Carlos Olavo de Azevedo Camacho Júnior é mestrando em Tecnologias da Inteligência e Design Digital pela Pontifícia Universidade Católica de São Paulo PUCSP. Pós-graduado em Análise e Projeto de Sistemas pela Universidade Paulista UNIP. Bacharel em Ciência da Computação pela Universidade Paulista UNIP e possui Licenciatura Plena em Matemática pelas Faculdades Oswaldo Cruz.  
MCP .Net, MCP SQL Server, Carlos Camacho leciona disciplinas técnicas na área de Ciências Exatas e é Consultor em Tecnologia da Informação para Instituições Financeiras.



## Leia também

[Lambda Expressions x SQL: Comparando a sintaxe de consultas comuns](#)  
C#


[List: trabalhando com listas genéricas em C#](#)  
C#

[Criando Gráficos usando C# e API do Google](#)  
C#


[Imprimindo um Panel com C#](#)  
C#

[Consumindo um Web API em C#](#)  
C#

Estamos aqui: [f](#) [s](#) [t](#) [g+](#)

**Linha de Código**  
**Like**

9,981 people like [Linha de Código](#).



Facebook social plugin

© 2014 Linha de Código. Todos os direitos reservados