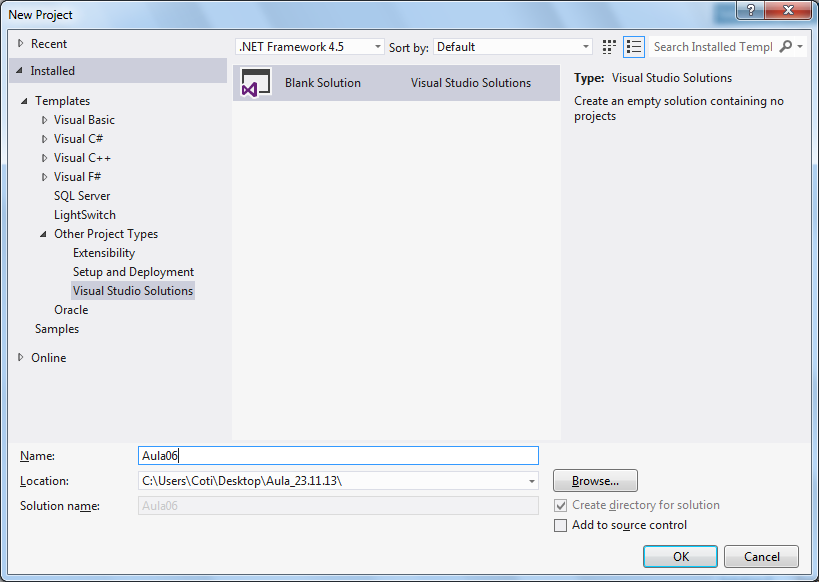
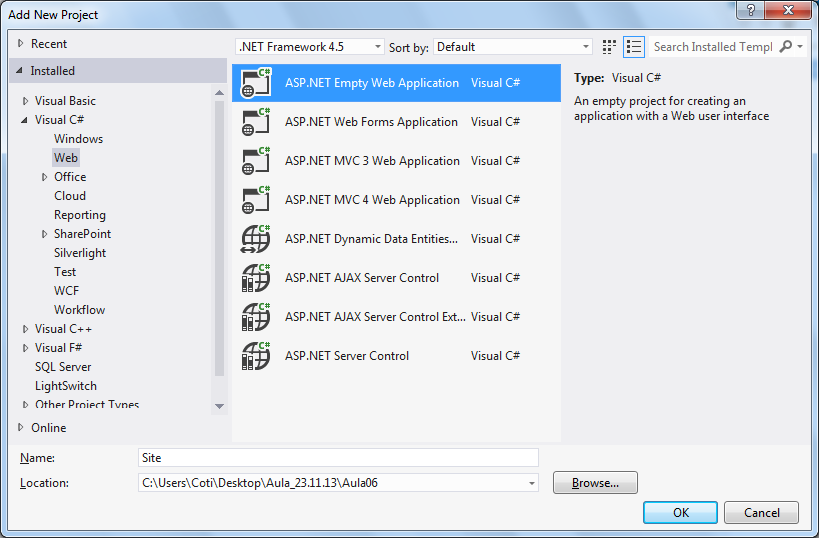
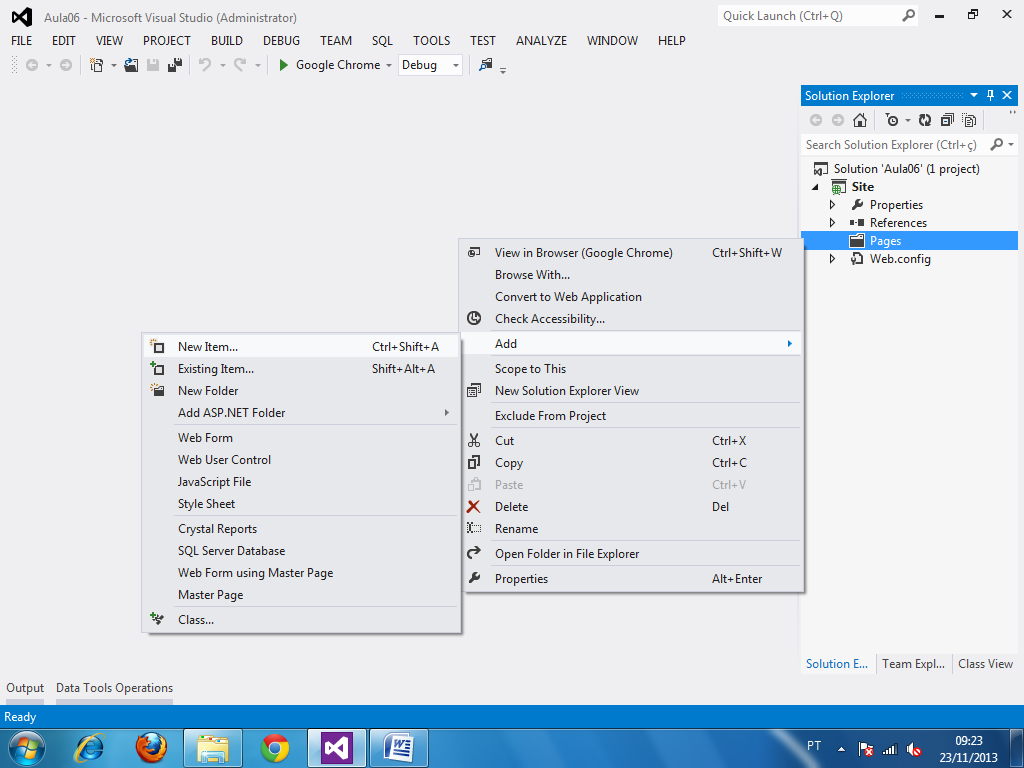
Criação da Solution



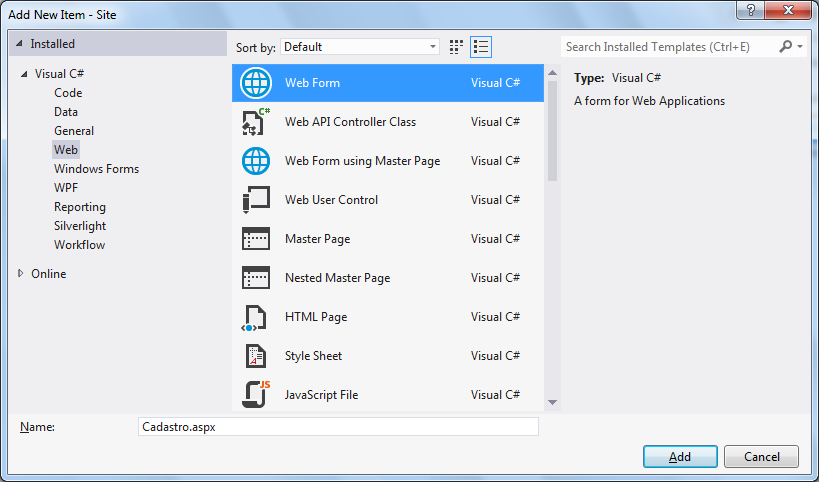
Criação do projeto Asp.NET

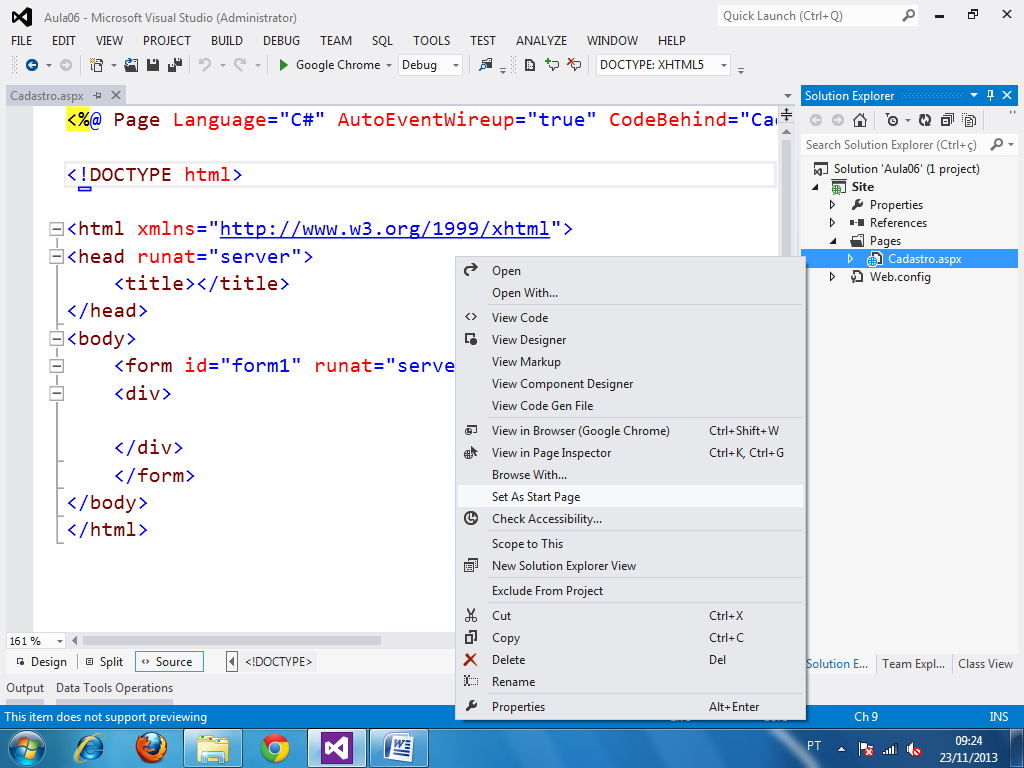


Criação da página .aspx



Criando o WebForm (Página aspx)





<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Cadastro.aspx.cs" Inherits="Site.Pages.Cadastro" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<h3>Cadastro de Produtos</h3>

<hr />

Nome do Produto: <br />

<asp:TextBox ID="txtNome" runat="server" />

<br /><br />

Preço: <br />

<asp:TextBox ID="txtPreco" runat="server" />

<br /><br />

Selecione o Fornecedor: <br />

<asp:DropDownList ID="ddlFornecedor" runat="server" />

<br /><br />

<asp:Button ID="btnCadastro" runat="server" Text="Cadastrar Produto" />

<p>

<asp:Label ID="lblMensagem" runat="server" />

</p>

</div>

</form>

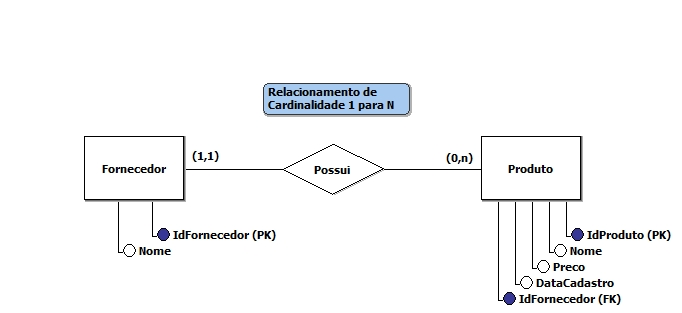
</body>

</html>



Modelagem de dados

Diagrama de entidade / relacionamento

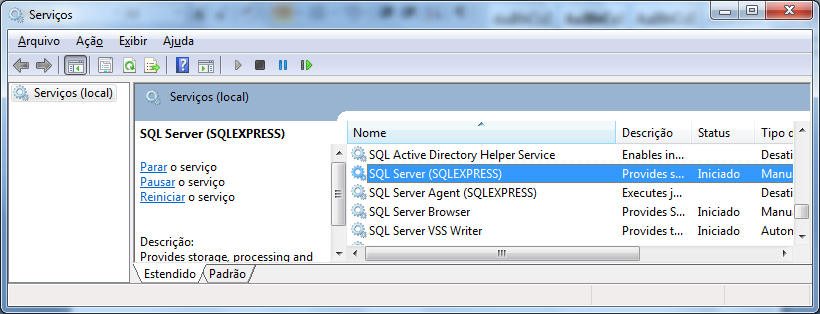


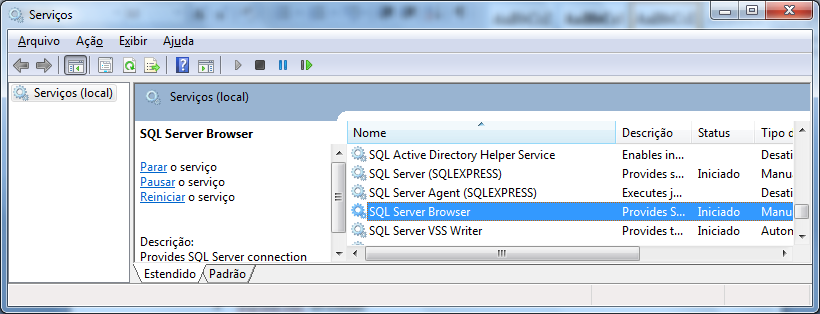
Arquivo .mdf

Microsoft Database File

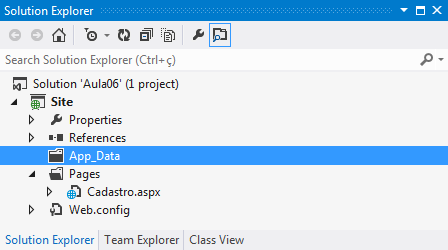
\*\* Iniciar os serviços do SqlServer

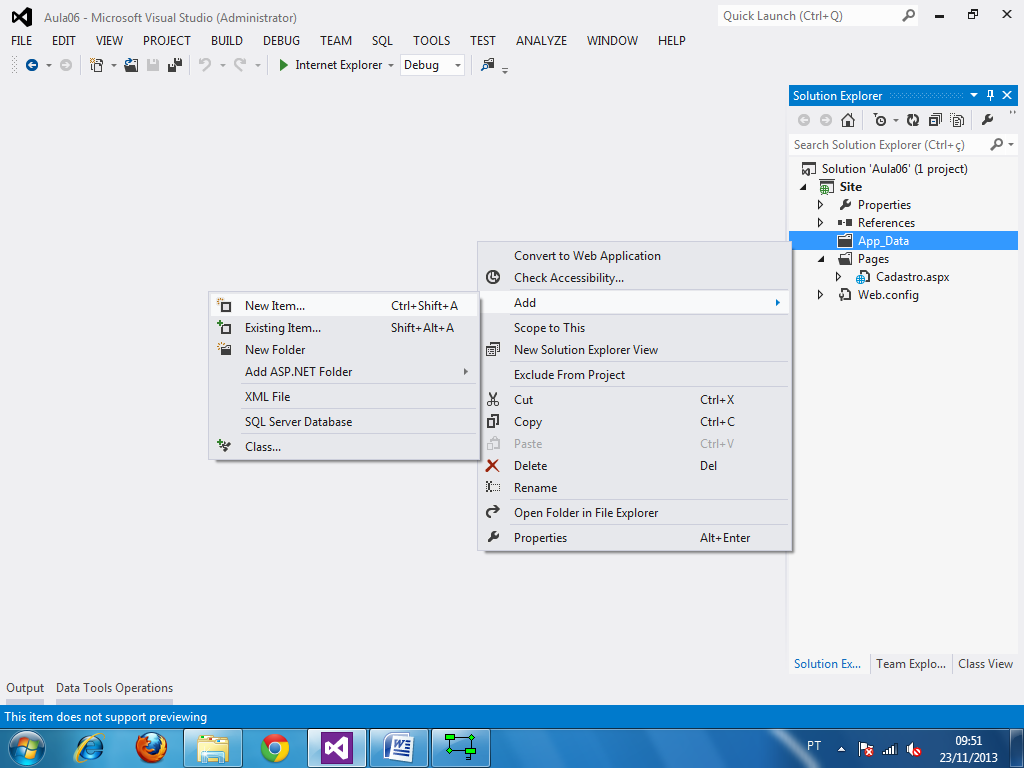
* SqlServer SQLEXPRESS
* SqlServer Browser

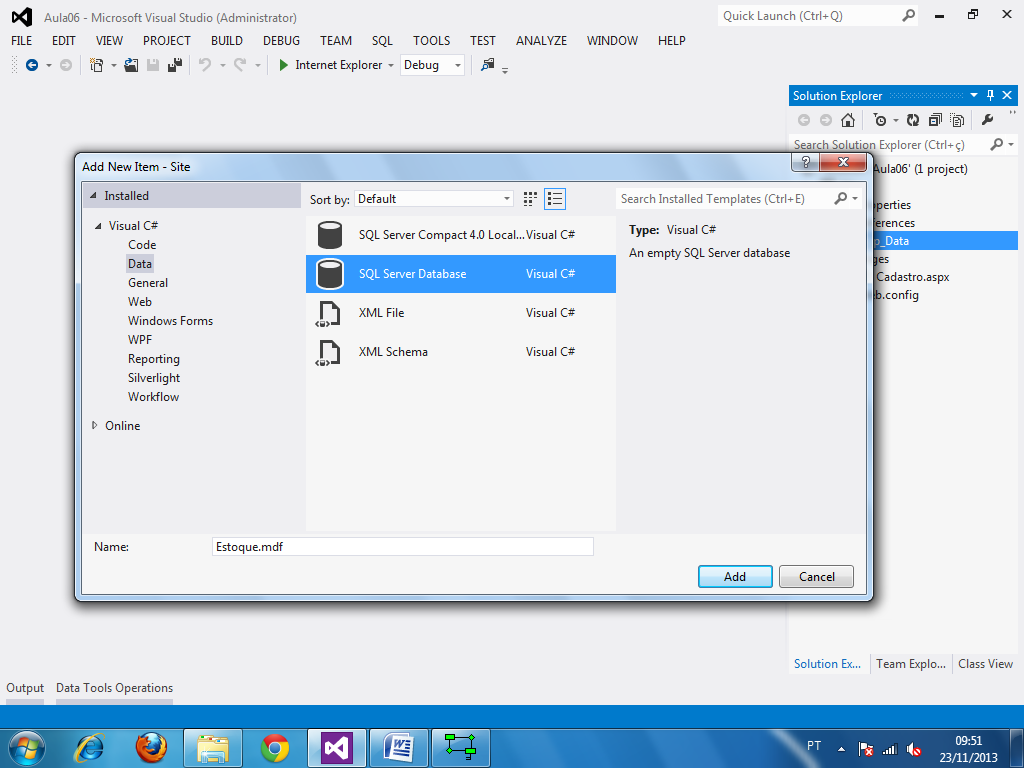




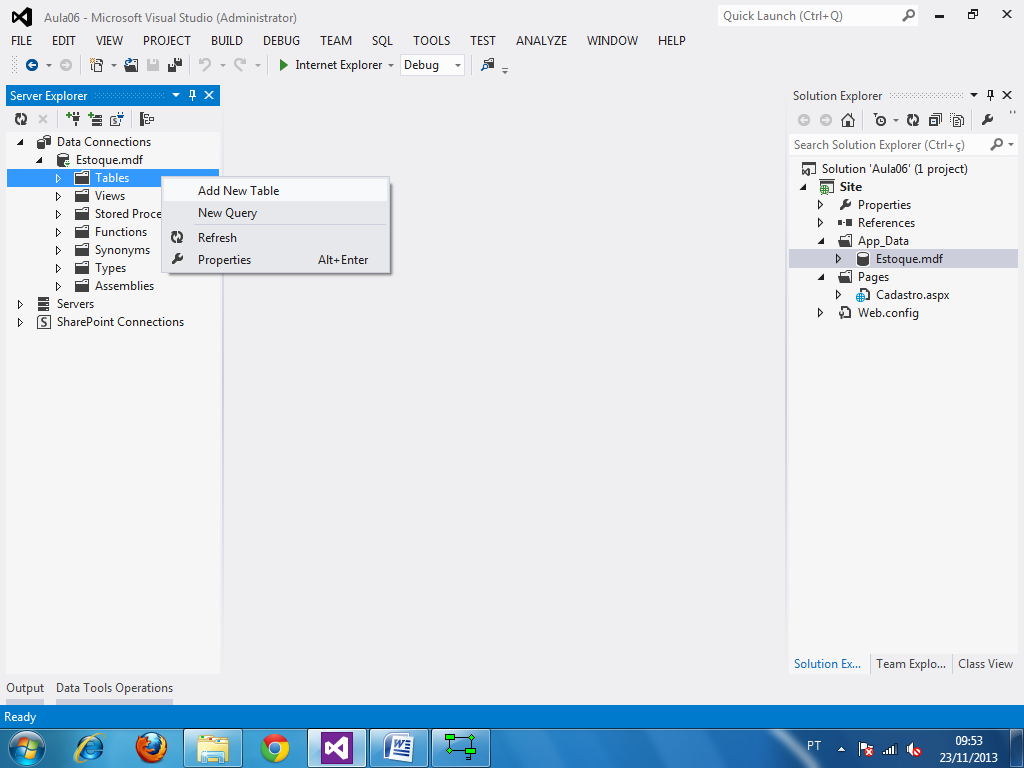
\*\* Para criar este arquivo **.mdf** dentro do projeto Asp.NET, por padrão, o arquivo deverá ficar dentro de uma pasta App\_Data







Criando as tabelas...



go

create table Fornecedor(

IdFornecedor integer identity,

Nome nvarchar(50) not null,

primary key(IdFornecedor))

go

create table Produto(

IdProduto integer identity,

Nome nvarchar(50) not null,

Preco float not null,

DataCadastro datetime not null,

IdFornecedor integer not null,

primary key(IdProduto))

go --regra de chave estrangeira => cardinalidade 1pN

alter table Produto add constraint FkProdutoFornecedor

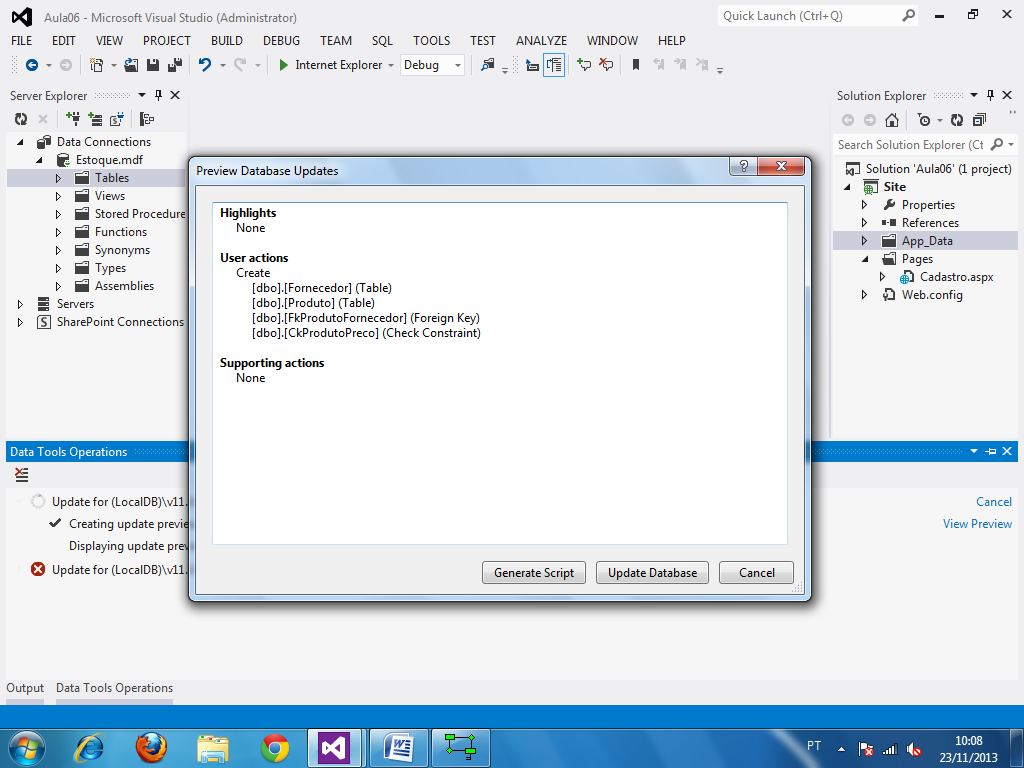
foreign key(IdFornecedor) references Fornecedor(IdFornecedor)

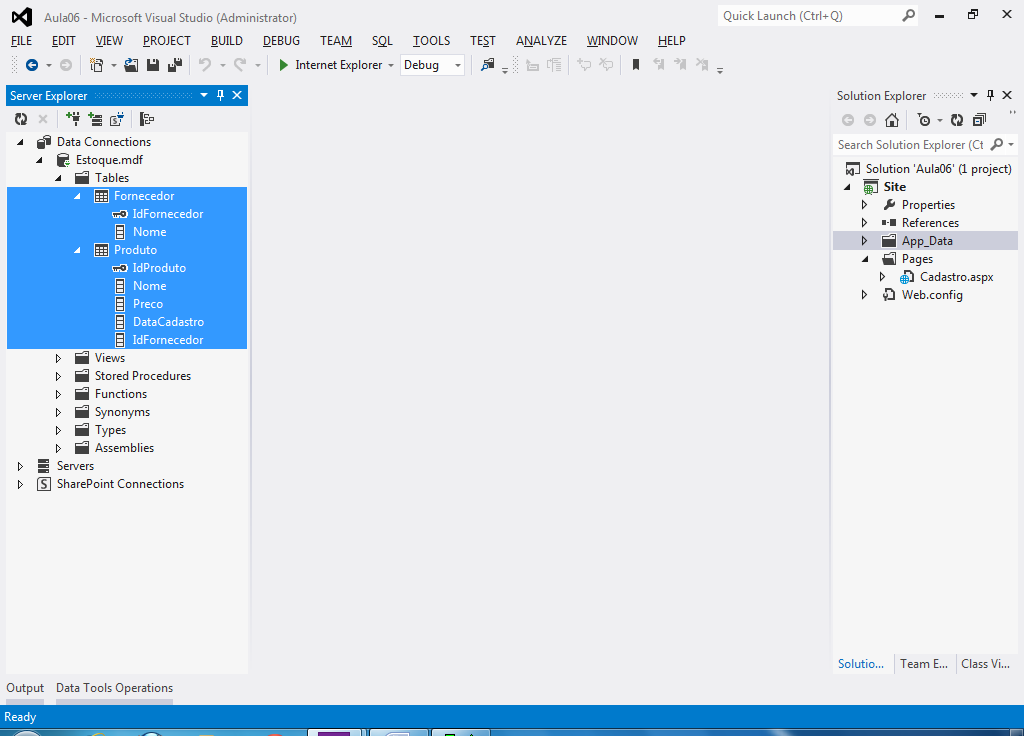
go --regra para limite do campo Preco do Produto

alter table Produto add constraint CkProdutoPreco

check(Preco > 0 and Preco <= 10000)

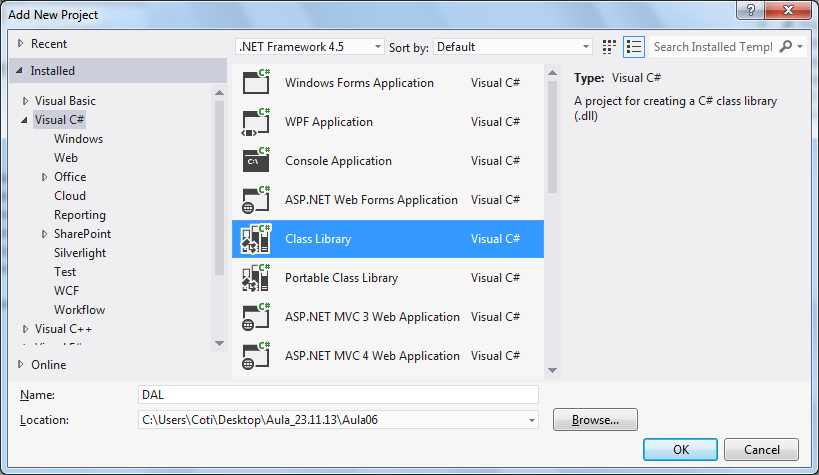
Update Database...





Data Access Layer

Camada de acesso a dados



Classes para realizar a modelagem das entidades...

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DAL.Model

{

public class Produto

{

//propriedades [prop] + 2x[tab]

public int IdProduto { get; set; }

public string Nome { get; set; }

public double Preco { get; set; }

public DateTime DataCadastro { get; set; }

//Relacionamento (Associação -> TER-1)

public Fornecedor Fornecedor { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DAL.Model

{

public class Fornecedor

{

//propriedades [prop] + 2x[tab]

public int IdFornecedor { get; set; }

public string Nome { get; set; }

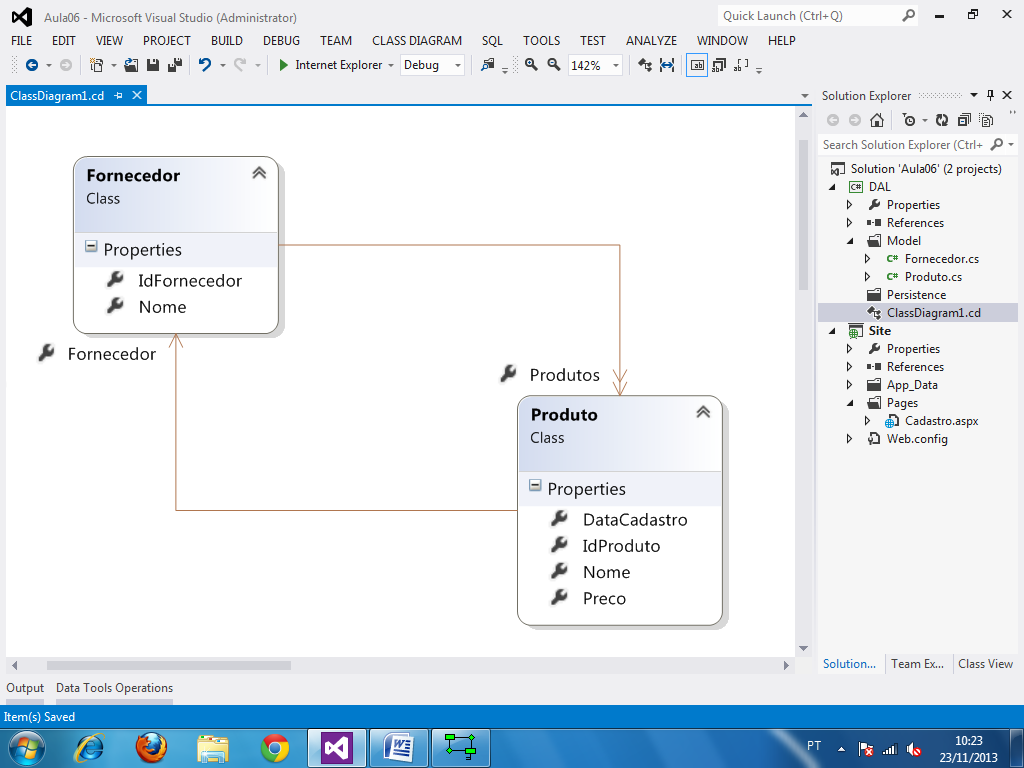
//Relacionamento (Associação -> TER-N)

public List<Produto> Produtos { get; set; }

}

}

Relacionamento bidirecional de   
multiplicidade 1 para Muitos



Conexão com o banco de dados

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient; //API padrão para o SqlServer

namespace DAL.Persistence

{

public class Conexao

{

protected SqlConnection Con;

protected SqlCommand Cmd;

protected SqlDataReader Dr;

protected void AbrirConexao()

{

Con = new SqlConnection();

Con.Open();

}

protected void FecharConexao()

{

Con.Close();

}

}

}

Mapear a String de Conexão

\Web.Config.xml

<?xml version="1.0"?>

<configuration>

<!-- Mapeamento da String de Conexão com o BD -->

**<connectionStrings>**

**<add**

**name="ESTOQUE"**

**connectionString="Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Coti\Desktop\Aula\_23.11.13\Aula06\Site\App\_Data\Estoque.mdf;Integrated Security=True"**

**/>**

**</connectionStrings>**

<system.web>

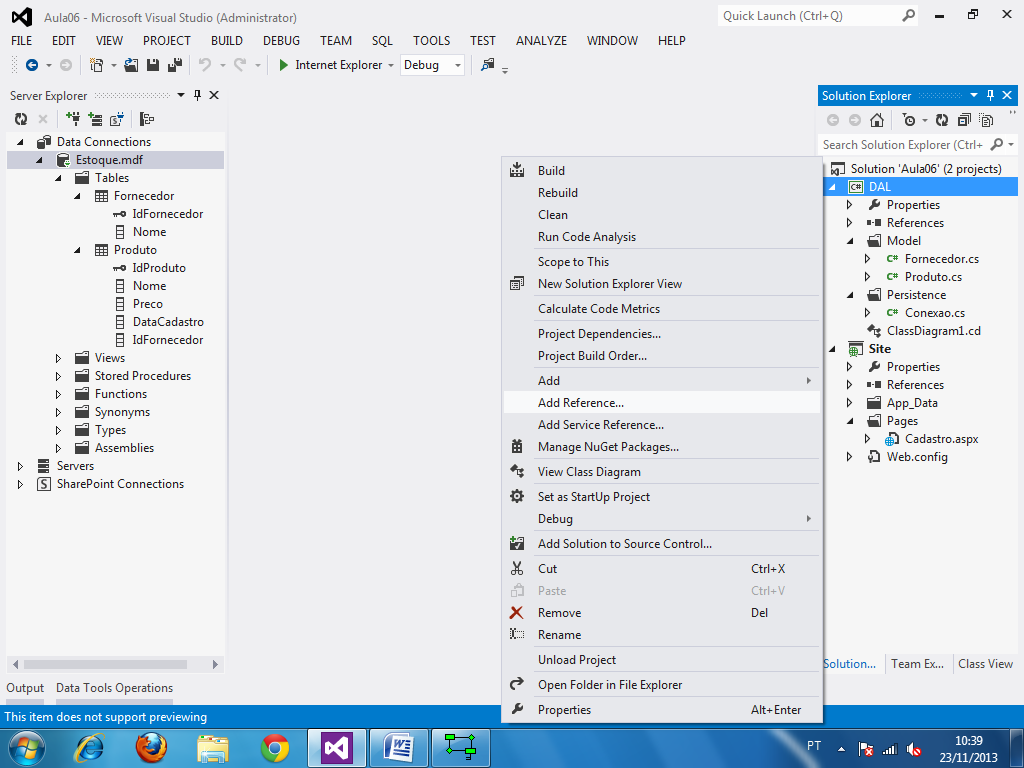
<compilation debug="true" targetFramework="4.5" />

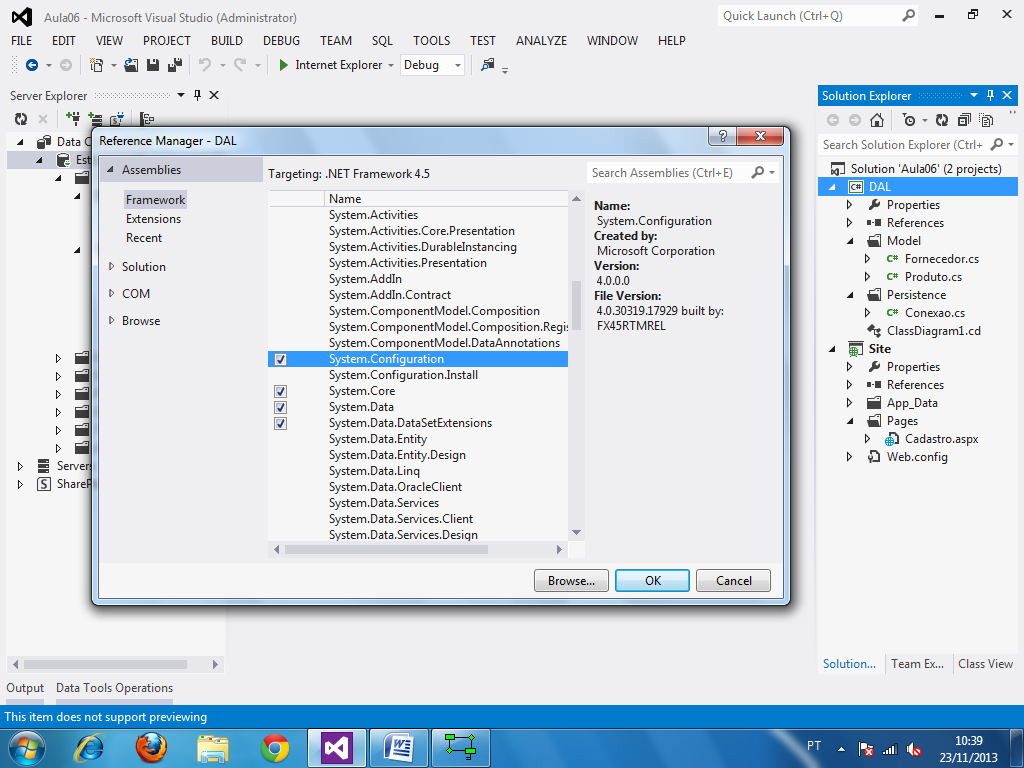
<httpRuntime targetFramework="4.5" />

</system.web>

</configuration>

Para que a biblioteca DAL possa referenciar a conexão mapeada no Web.config.xml, é necessário incluir a referência da biblioteca: System.Configuration





using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient; //API padrão para o SqlServer

using System.Configuration; //referenciar a ConnectionString mapeada

namespace DAL.Persistence

{

public class Conexao

{

protected SqlConnection Con;

protected SqlCommand Cmd;

protected SqlDataReader Dr;

protected void AbrirConexao()

{

Con = new SqlConnection();

Con.ConnectionString = ConfigurationManager.ConnectionStrings

["ESTOQUE"].ConnectionString;

Con.Open(); //conecta

}

protected void FecharConexao()

{

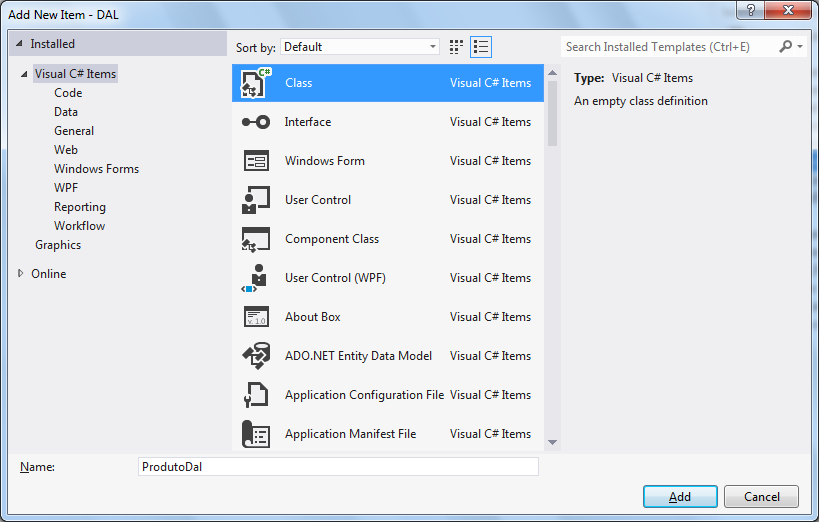
Con.Close(); //desconecta

}

}

}

Classes de persistência...



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient; //API do SqlServer

using DAL.Model; //Camada de modelo (entidades)

namespace DAL.Persistence

{

public class ProdutoDal : Conexao

{

//Método para gravar um registro na tabela Produto

public void Salvar(Produto p)

{

try

{

AbrirConexao();

Cmd = new SqlCommand("insert into Produto(Nome, Preco,

DataCadastro, IdFornecedor) values(@v1, @v2, @v3, @v4)", Con);

Cmd.Parameters.AddWithValue("@v1", p.Nome);

Cmd.Parameters.AddWithValue("@v2", p.Preco);

Cmd.Parameters.AddWithValue("@v3", p.DataCadastro);

Cmd.Parameters.AddWithValue("@v4", p.Fornecedor.IdFornecedor);

Cmd.ExecuteNonQuery();

}

catch (Exception e)

{

throw new Exception("Erro ao cadastrar Produto: " + e.Message);

}

finally

{

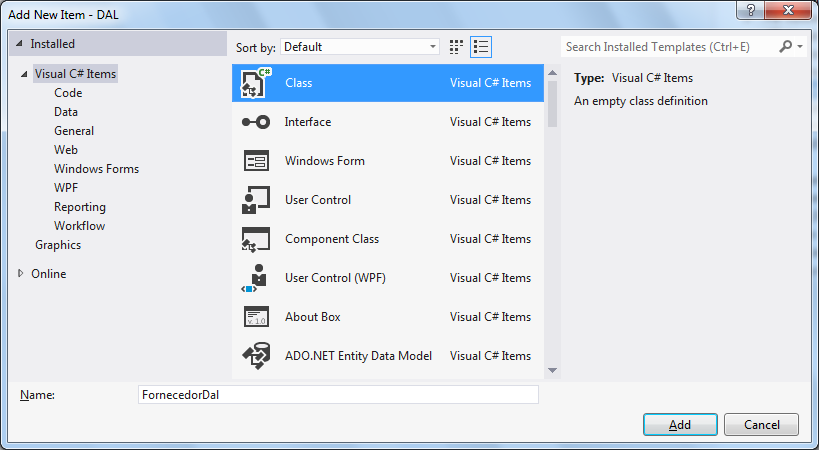
FecharConexao();

}

}

}

}



Classe para listar todos os   
fornecedores cadastrados...

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient; //API para o SqlServer

using DAL.Model; //Camada de modelo (entidades)

namespace DAL.Persistence

{

public class FornecedorDal : Conexao

{

//Método para listar todos os fornecedores cadastrados

**public List<Fornecedor> ListarTodos()**

{

try

{

AbrirConexao();

Cmd = new SqlCommand("select \* from Fornecedor", Con);

Dr = Cmd.ExecuteReader();

//executa a consulta e captura os registros

List<Fornecedor> lista = new List<Fornecedor>();

//lista vazia

while(Dr.Read())

//percorrendo cada registro obtido pelo DataReader

{

Fornecedor f = new Fornecedor();

f.IdFornecedor = Convert.ToInt32(Dr["IdFornecedor"]);

f.Nome = Convert.ToString(Dr["Nome"]);

lista.Add(f); //adiciono o objeto dentro da lista

}

return lista; //retornar a lista

}

catch(Exception e)

{

throw new Exception("Erro ao consultar fornecedores: "

+ e.Message);

}

finally

{

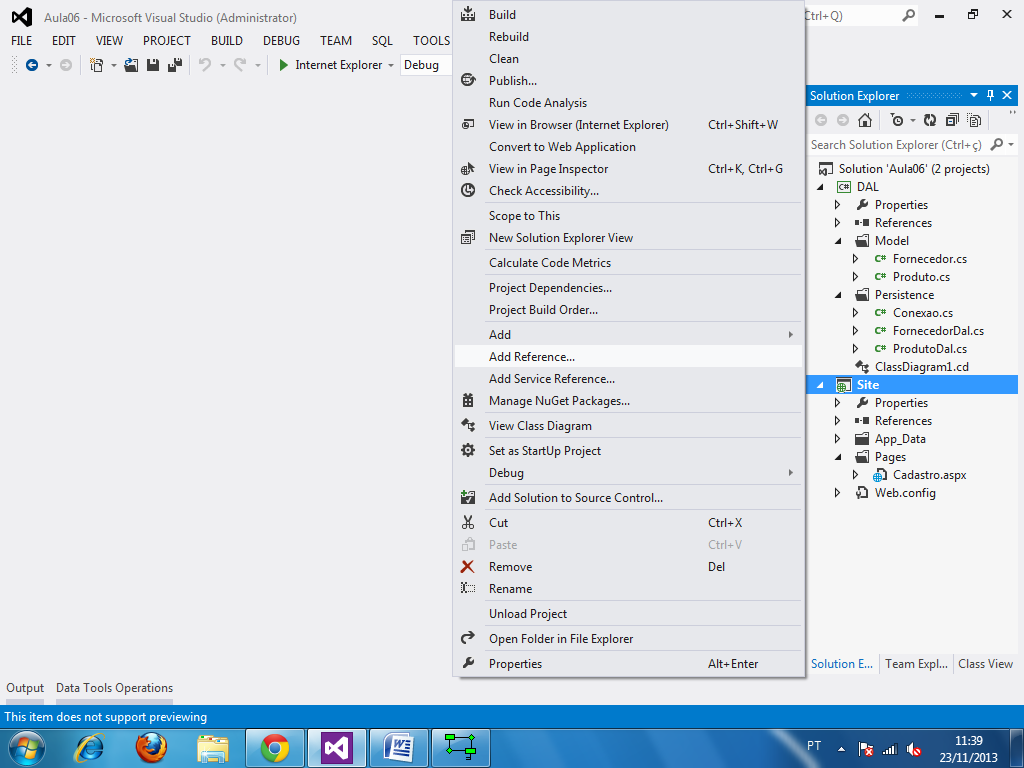
FecharConexao();

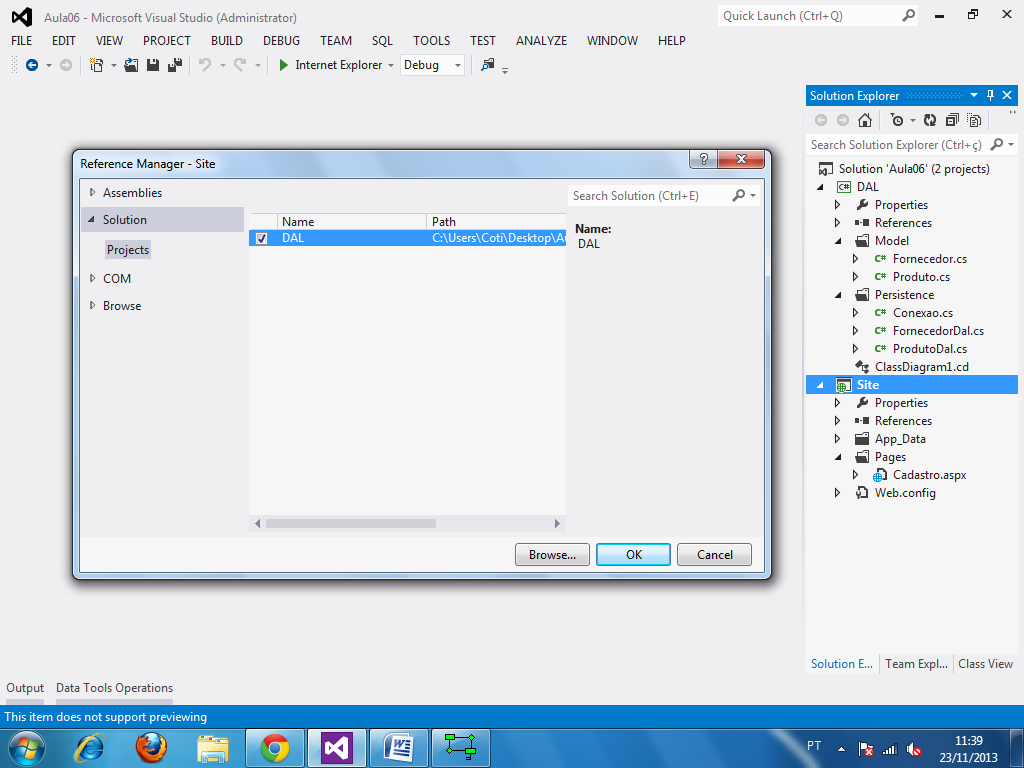
}

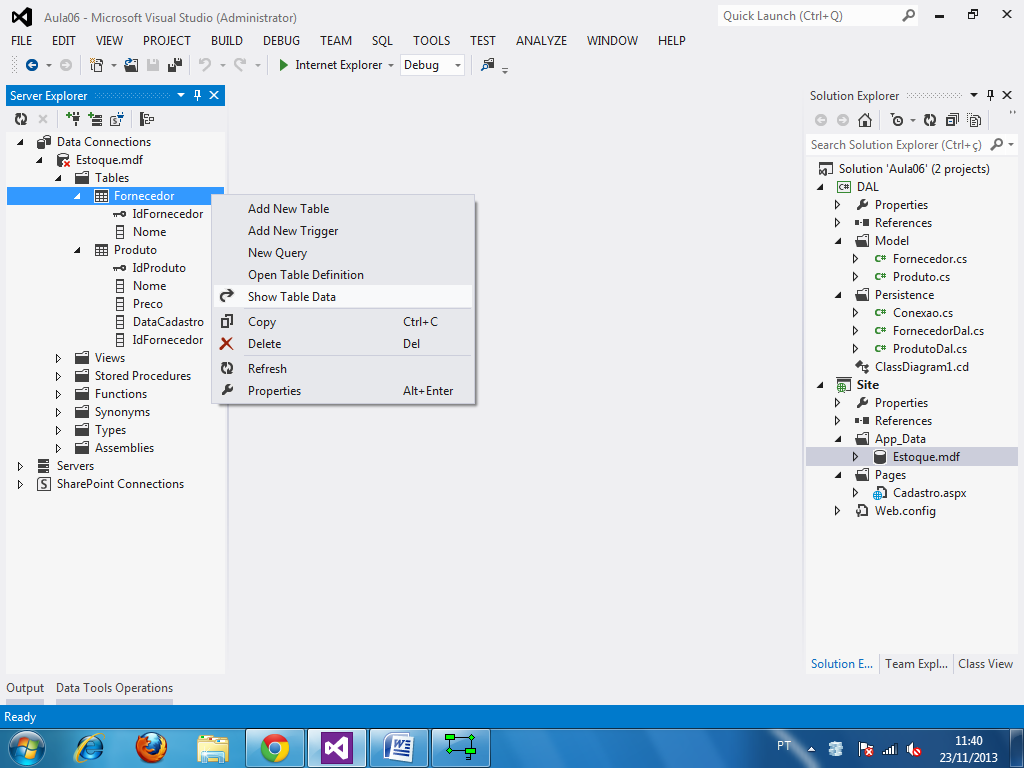
}

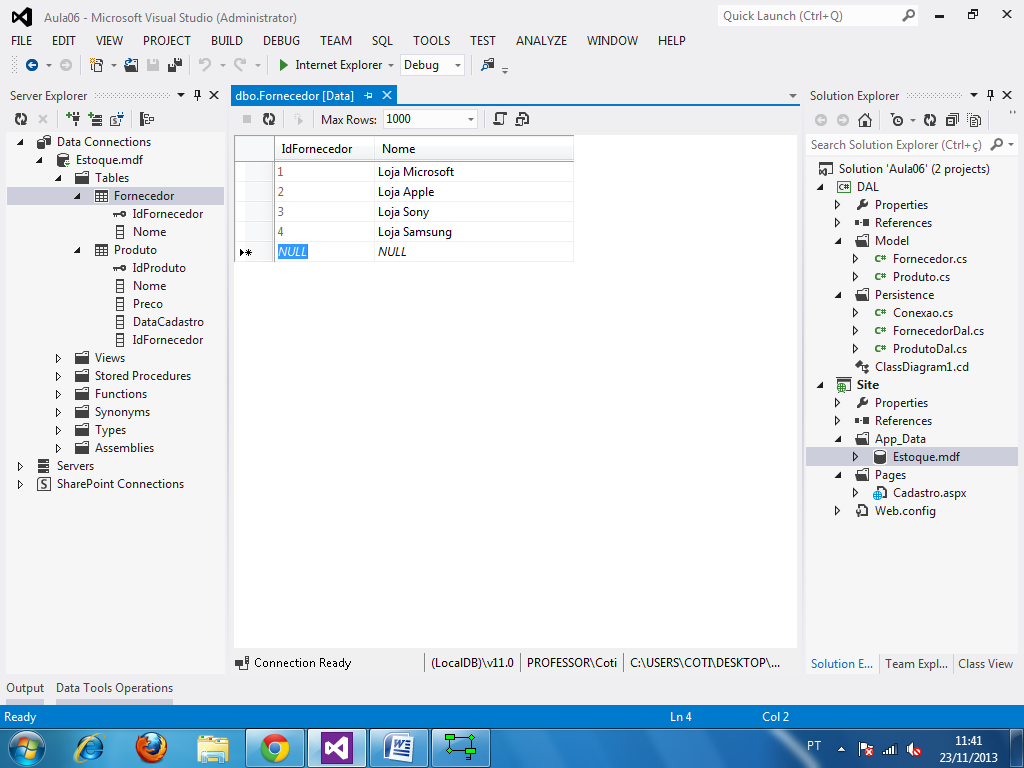
}

}









CodeBehind [F7]

Programação “lado-servidor” de uma página .aspx

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using DAL.Model;

using DAL.Persistence;

namespace Site.Pages

{

public partial class Cadastro : System.Web.UI.Page

{

//Evento executado toda vez que a página é carregada

protected void Page\_Load(object sender, EventArgs e)

{

try

{

FornecedorDal d = new FornecedorDal(); //Classe de Persistência

//DataSource -> Popular o componente com conteudo

//DataBind -> Exibir o conteudo do DataSource

//dentro do componente

ddlFornecedor.DataSource = d.ListarTodos(); //Popular

ddlFornecedor.DataValueField = "IdFornecedor"; //valor do campo

ddlFornecedor.DataTextField = "Nome"; //texto exibido no campo

ddlFornecedor.DataBind(); //Exibir

//Incluir um elemento default no campo DropDownList

ddlFornecedor.Items.Insert(0,

new ListItem("- Escolha uma Opção -", "0"));

}

catch(Exception ex)

{

//Escrever a mensagem de erro dentro do Label

lblMensagem.Text = ex.Message;

}

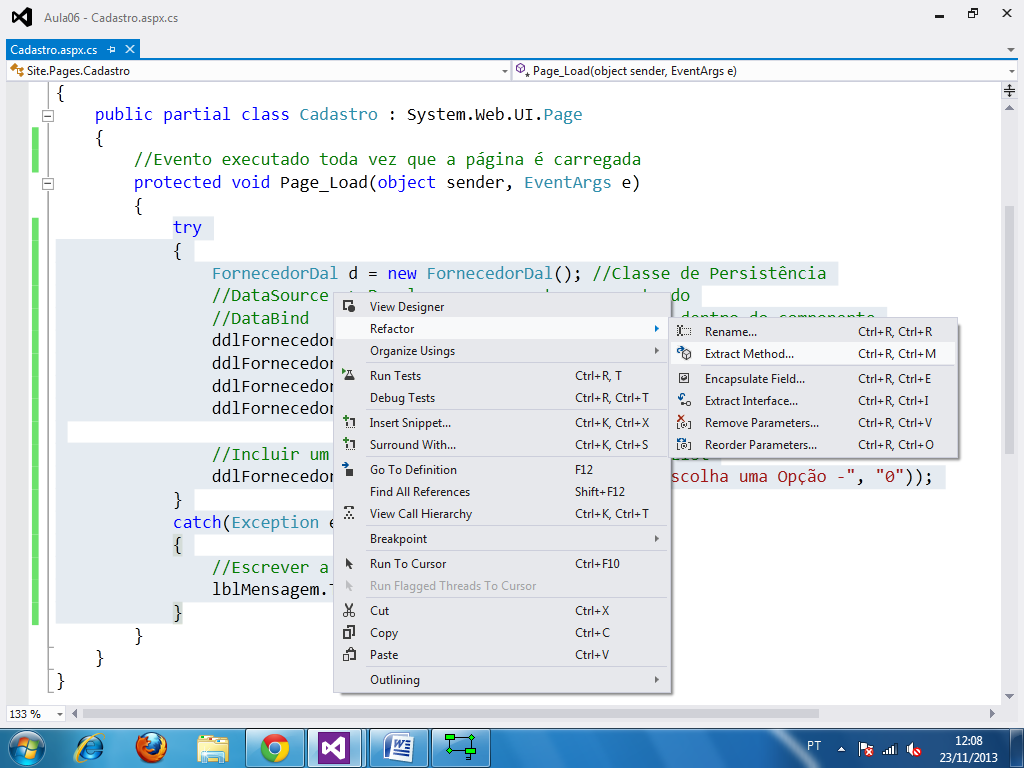
}

}

}

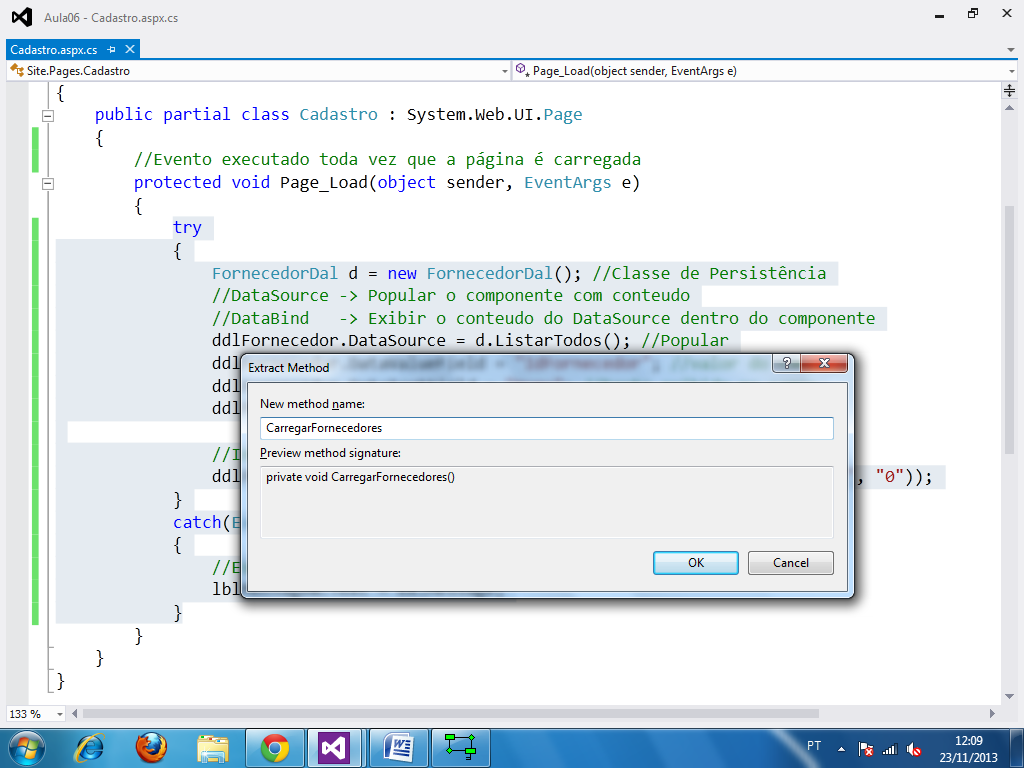
Refatoração de Código

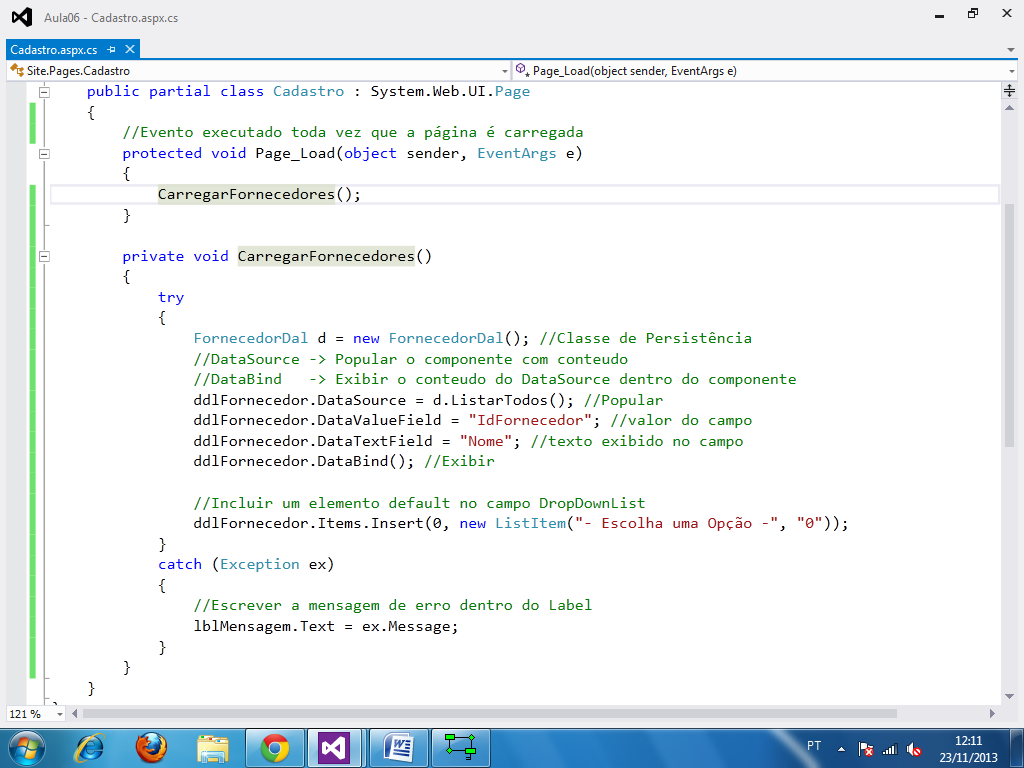
Otimização (Melhoria de codificação)



Extract Method

Técnica de refatoração de retira um bloco de código e o isola em uma método separado...





using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using DAL.Model;

using DAL.Persistence;

namespace Site.Pages

{

public partial class Cadastro : System.Web.UI.Page

{

//Evento executado toda vez que a página é carregada

protected void Page\_Load(object sender, EventArgs e)

{

**CarregarFornecedores();**

}

**private void CarregarFornecedores()**

{

try

{

FornecedorDal d = new FornecedorDal(); //Classe de Persistência

//DataSource -> Popular o componente com conteudo

//DataBind -> Exibir o conteudo do DataSource

//dentro do componente

ddlFornecedor.DataSource = d.ListarTodos(); //Popular

ddlFornecedor.DataValueField = "IdFornecedor"; //valor do campo

ddlFornecedor.DataTextField = "Nome"; //texto exibido no campo

ddlFornecedor.DataBind(); //Exibir

//Incluir um elemento default no campo DropDownList

ddlFornecedor.Items.Insert(0,

new ListItem("- Escolha uma Opção -", "0"));

}

catch (Exception ex)

{

//Escrever a mensagem de erro dentro do Label

lblMensagem.Text = ex.Message;

}

}

}

}

O Método Page\_Load sempre é executado toda vez que a página for carregada pelo navegador, inclusive quando por exemplo, um botão for clicado na página.

Podemos condicionar a execução do Page\_Load para que seja realizado somente na 1º vez que a página abrir

IsPostBack

**true** 🡪 Indica que a página já foi previamente carregada

**false** 🡪 Indica que a página está sendo carrega pela 1º vez

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using DAL.Model;

using DAL.Persistence;

namespace Site.Pages

{

public partial class Cadastro : System.Web.UI.Page

{

//Evento executado toda vez que a página é carregada

protected void Page\_Load(object sender, EventArgs e)

{

//Verificar se a página está sendo carregada pela 1º vez

//Se o IsPostBack é falso

**if( ! IsPostBack ) //A página não foi carregada**

**{**

**CarregarFornecedores();**

**}**

}

private void CarregarFornecedores()

{

try

{

FornecedorDal d = new FornecedorDal(); //Classe de Persistência

//DataSource -> Popular o componente com conteudo

//DataBind -> Exibir o conteudo do DataSource

//dentro do componente

ddlFornecedor.DataSource = d.ListarTodos(); //Popular

ddlFornecedor.DataValueField = "IdFornecedor"; //valor do campo

ddlFornecedor.DataTextField = "Nome"; //texto exibido no campo

ddlFornecedor.DataBind(); //Exibir

//Incluir um elemento default no campo DropDownList

ddlFornecedor.Items.Insert(0,

new ListItem("- Escolha uma Opção -", "0"));

}

catch (Exception ex)

{

//Escrever a mensagem de erro dentro do Label

lblMensagem.Text = ex.Message;

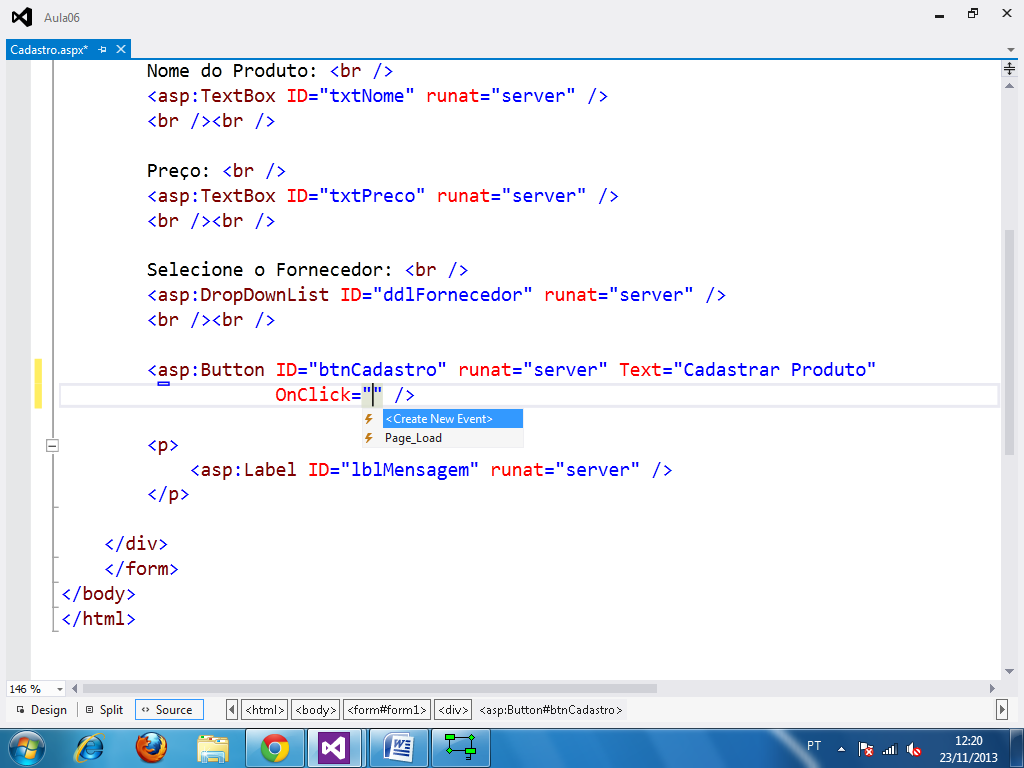
}

}

}

}

Criar o evento para realizar a   
gravação dos dados do Produto...



<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Cadastro.aspx.cs" Inherits="Site.Pages.Cadastro" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<h3>Cadastro de Produtos</h3>

<hr />

Nome do Produto: <br />

<asp:TextBox ID="txtNome" runat="server" />

<br /><br />

Preço: <br />

<asp:TextBox ID="txtPreco" runat="server" />

<br /><br />

Selecione o Fornecedor: <br />

<asp:DropDownList ID="ddlFornecedor" runat="server" />

<br /><br />

<asp:Button ID="btnCadastro" runat="server" Text="Cadastrar Produto"

**OnClick="btnCadastro\_Click"** />

<p>

<asp:Label ID="lblMensagem" runat="server" />

</p>

</div>

</form>

</body>

</html>

CodeBehind

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using DAL.Model;

using DAL.Persistence;

namespace Site.Pages

{

public partial class Cadastro : System.Web.UI.Page

{

//Evento executado toda vez que a página é carregada

protected void Page\_Load(object sender, EventArgs e)

{

//Verificar se a página está sendo carregada pela 1º vez

//Se o IsPostBack é falso

if( ! IsPostBack ) //A página não foi carregada

{

CarregarFornecedores();

}

}

private void CarregarFornecedores()

{

try

{

FornecedorDal d = new FornecedorDal(); //Classe de Persistência

//DataSource -> Popular o componente com conteudo

//DataBind -> Exibir o conteudo do DataSource

//dentro do componente

ddlFornecedor.DataSource = d.ListarTodos(); //Popular

ddlFornecedor.DataValueField = "IdFornecedor"; //valor do campo

ddlFornecedor.DataTextField = "Nome"; //texto exibido no campo

ddlFornecedor.DataBind(); //Exibir

//Incluir um elemento default no campo DropDownList

ddlFornecedor.Items.Insert(0,

new ListItem("- Escolha uma Opção -", "0"));

}

catch (Exception ex)

{

//Escrever a mensagem de erro dentro do Label

lblMensagem.Text = ex.Message;

}

}

//Método executado quando o botão for clicado

protected void btnCadastro\_Click(object sender, EventArgs e)

{

try

{

Produto p = new Produto(); //Instanciar a Classe Produto

p.Fornecedor = new Fornecedor(); //Instanciar a Classe Fornecedor

//Capturar o conteúdo dos campos do formulário

p.Nome = txtNome.Text;

p.Preco = Convert.ToDouble(txtPreco.Text);

p.DataCadastro = DateTime.Now; //captura a data do sistema

p.Fornecedor.IdFornecedor = Convert.ToInt32

(ddlFornecedor.SelectedValue);

ProdutoDal d = new ProdutoDal();

d.Salvar(p); //Salvar o Objeto

lblMensagem.Text = "Produto " + p.Nome

+ ", cadastrado com sucesso.";

}

catch(Exception ex)

{

//Escrever mensagem de erro dentro do Label

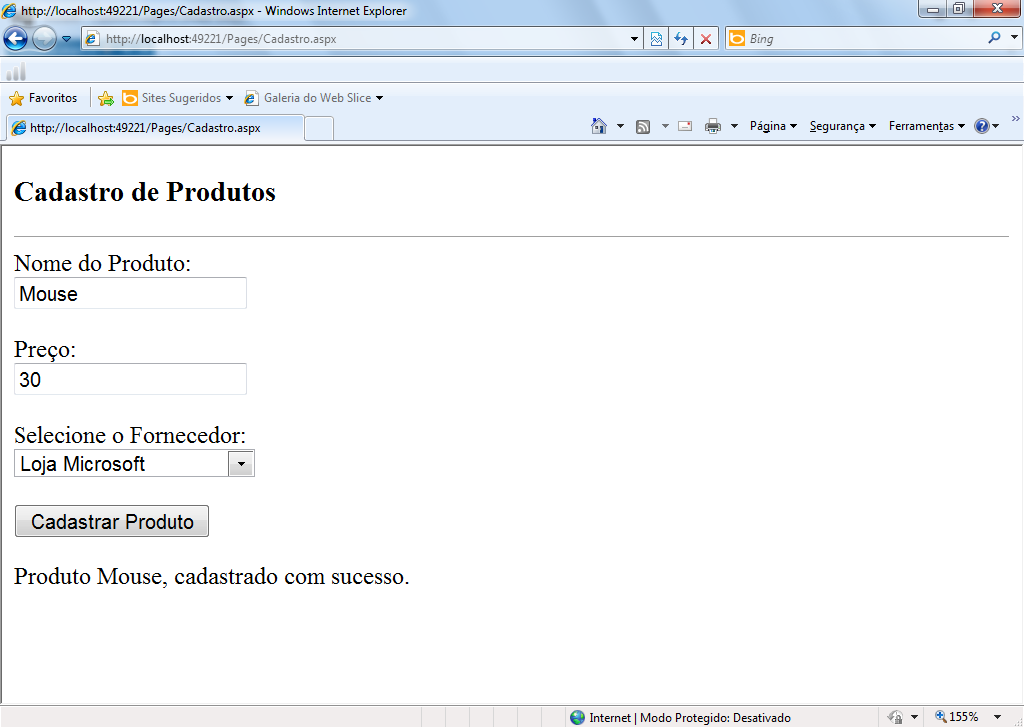
lblMensagem.Text = ex.Message;

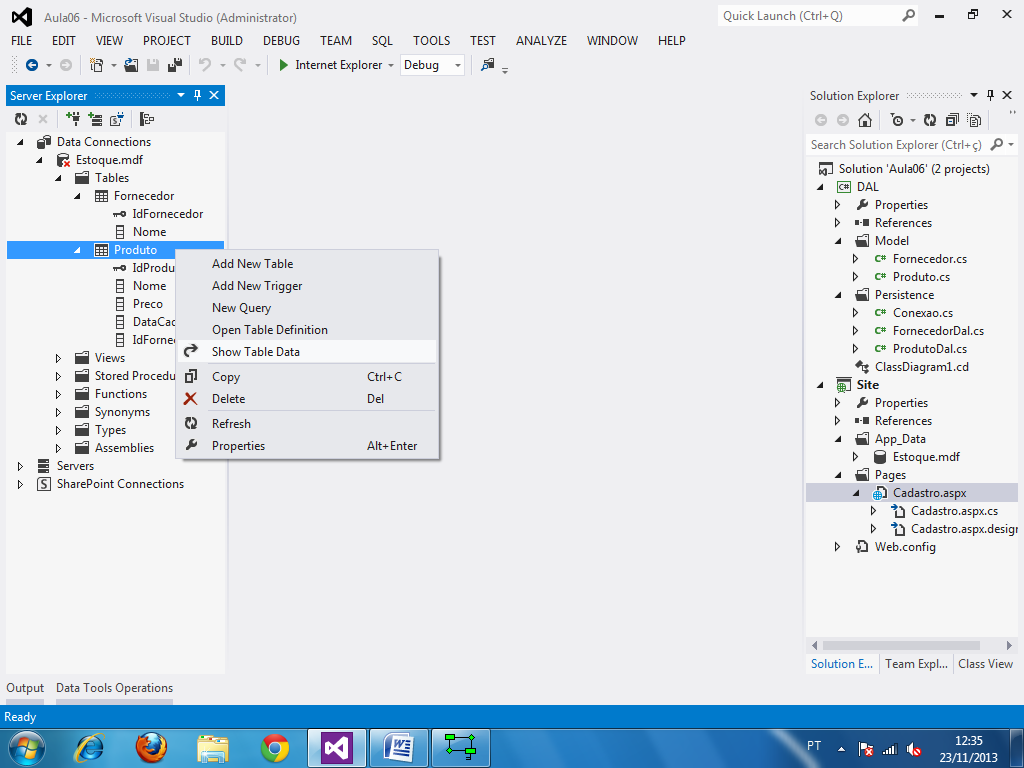
}

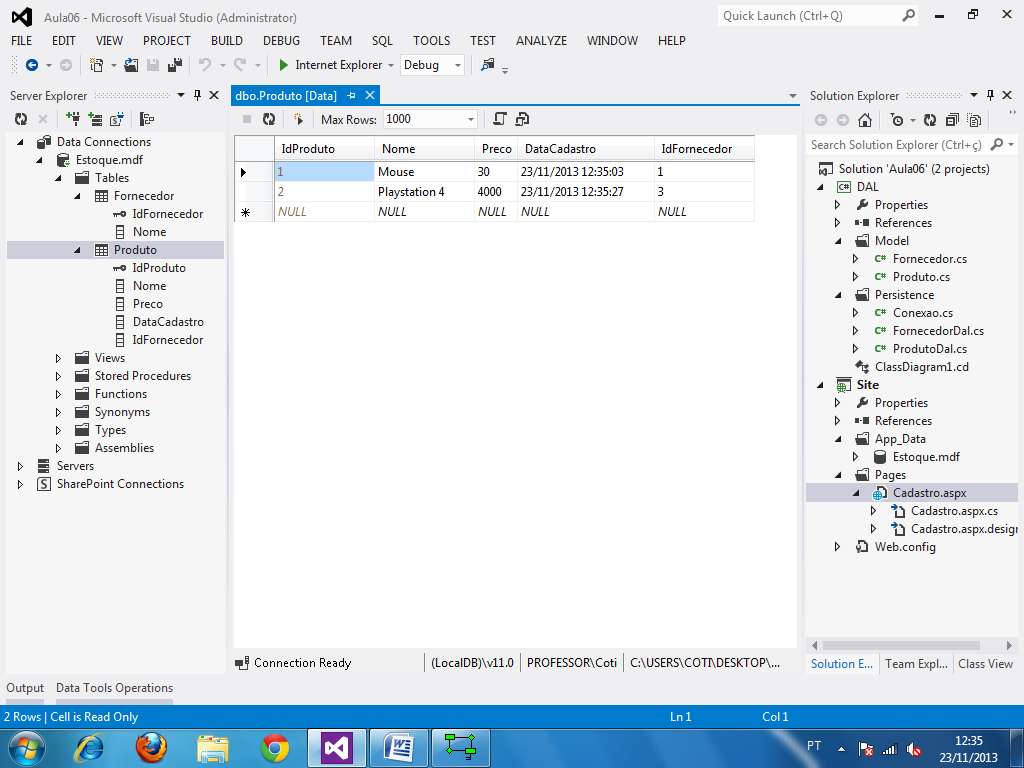
}

}

}







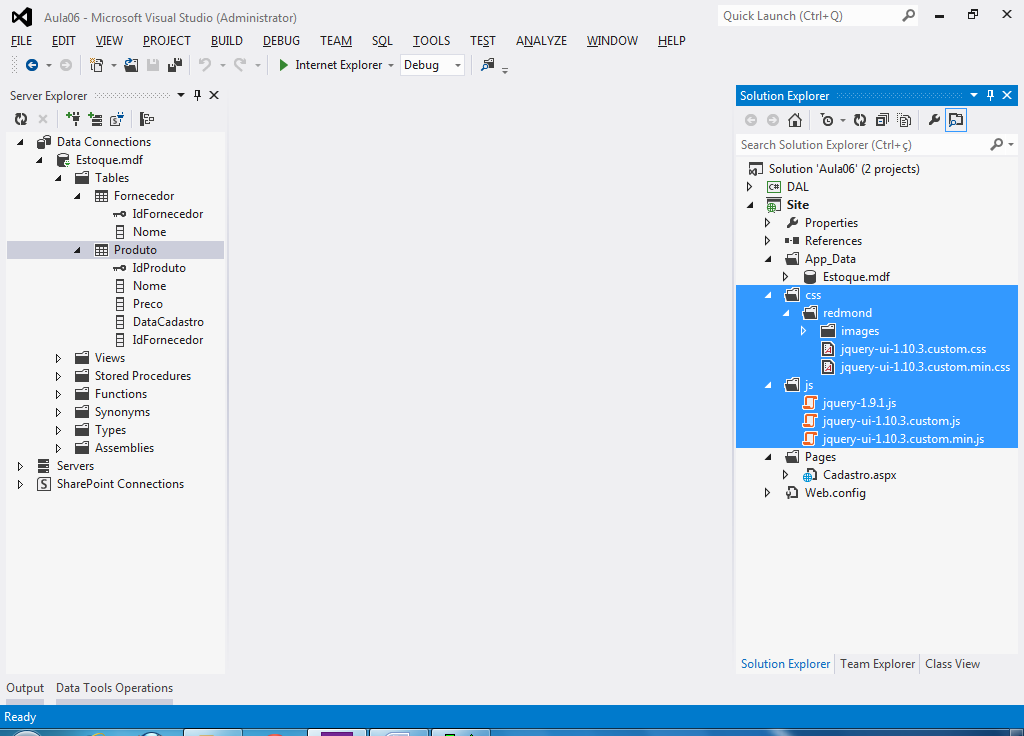
JQuery

Conjunto de bibliotecas JavaScript utilizado para otimização de desenvolvimento voltado para frontend de uma aplicação web.

* JQuery User Interface

Biblioteca JQuery voltado para componentes de interface web

<http://www.jqueryui.com/>



<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Cadastro.aspx.cs" Inherits="Site.Pages.Cadastro" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

**<link href="../css/redmond/jquery-ui-1.10.3.custom.css" rel="stylesheet" />**

**<script src="../js/jquery-1.9.1.js"></script>**

**<script src="../js/jquery-ui-1.10.3.custom.js"></script>**

**<script type="text/javascript">**

**//Executar quando a página estiver pronta**

**$(document).ready(**

**function () { //faça...**

**//transformar o conteudo da div em uma janela de díalogo**

**$("#janela").dialog(); //[#] captura o elemento pelo id**

**$("#btnCadastro").button();**

**}**

**);**

**</script>**

</head>

<body>

<div id="janela" title="Formulário de Inclusão">

<form id="form1" runat="server">

<div>

<h3>Cadastro de Produtos</h3>

<hr />

Nome do Produto: <br />

<asp:TextBox ID="txtNome" runat="server" />

<br /><br />

Preço: <br />

<asp:TextBox ID="txtPreco" runat="server" />

<br /><br />

Selecione o Fornecedor: <br />

<asp:DropDownList ID="ddlFornecedor" runat="server" />

<br /><br />

<asp:Button ID="btnCadastro" runat="server" Text="Cadastrar Produto"

OnClick="btnCadastro\_Click" />

<p>

<asp:Label ID="lblMensagem" runat="server" />

</p>

</div>

</form>

</div>

</body>

</html>

