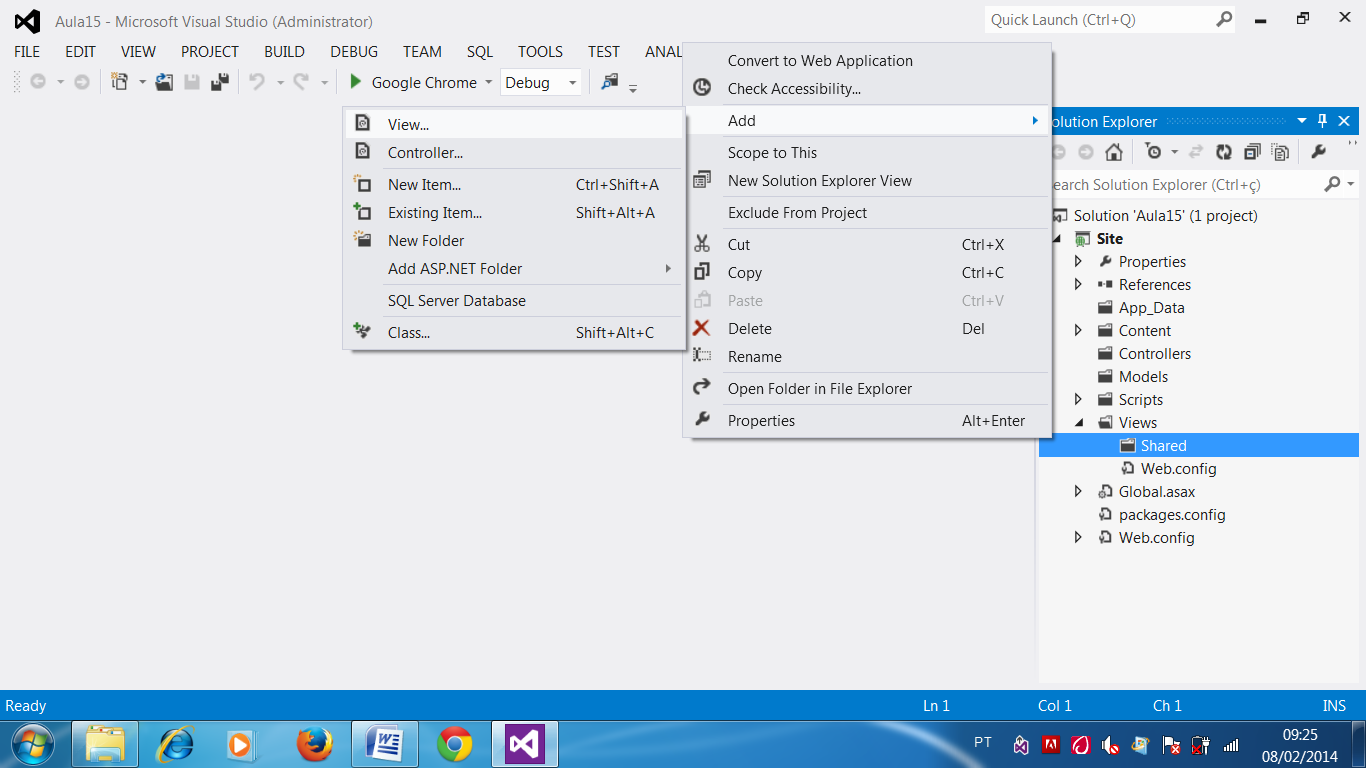
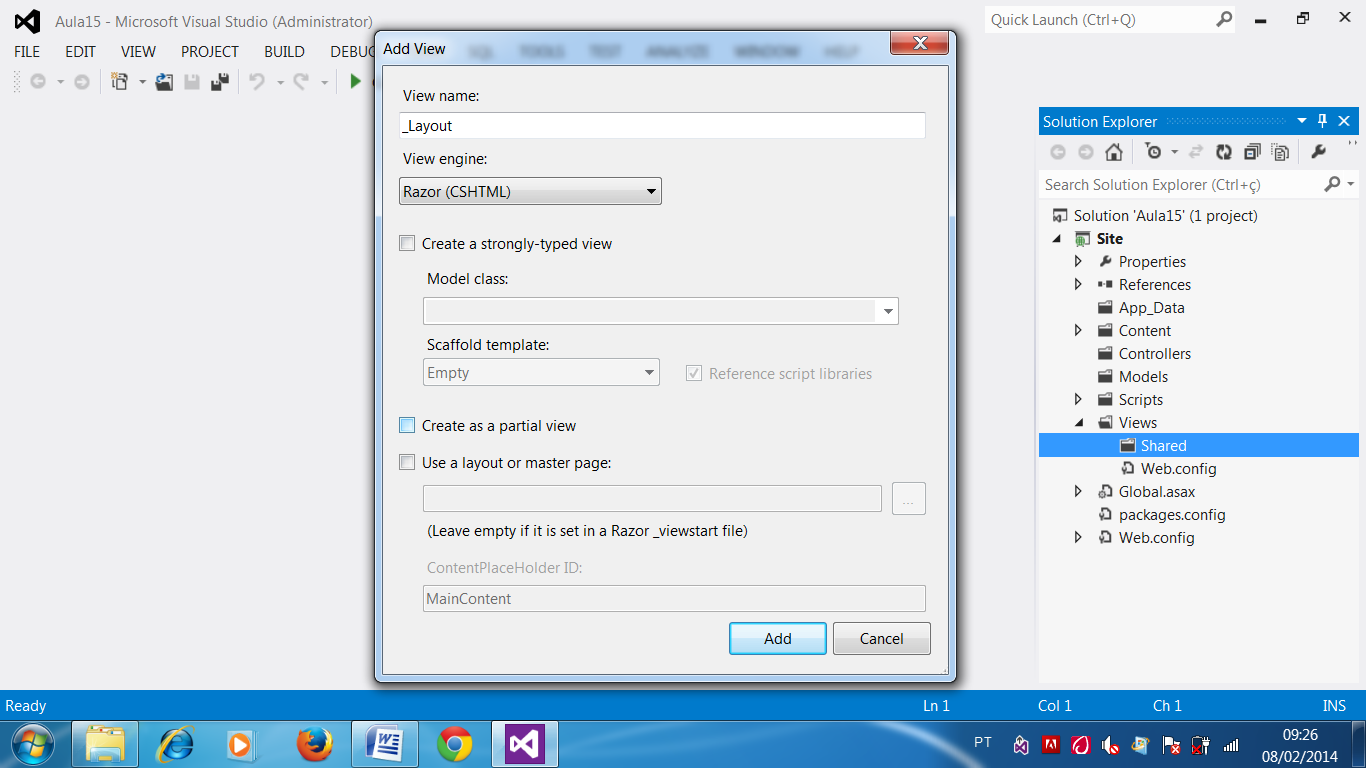


Página de Layout





<!DOCTYPE html>

<html>

<head>

<title>\_Layout</title>

<link href="../../Content/Site.css" rel="stylesheet" />

</head>

<body>

<div>

<h3>Projeto Asp.NET MVC3</h3>

<ul>

<li> @Html.ActionLink("Manter Produtos", "Cadastro", "Produto") </li>

<li> @Html.ActionLink("Pesquisar Produtos", "Pesquisa", "Produto") </li>

</ul>

<hr />

<!-- Área para receber o conteudo das demais páginas que herdarão o layout -->

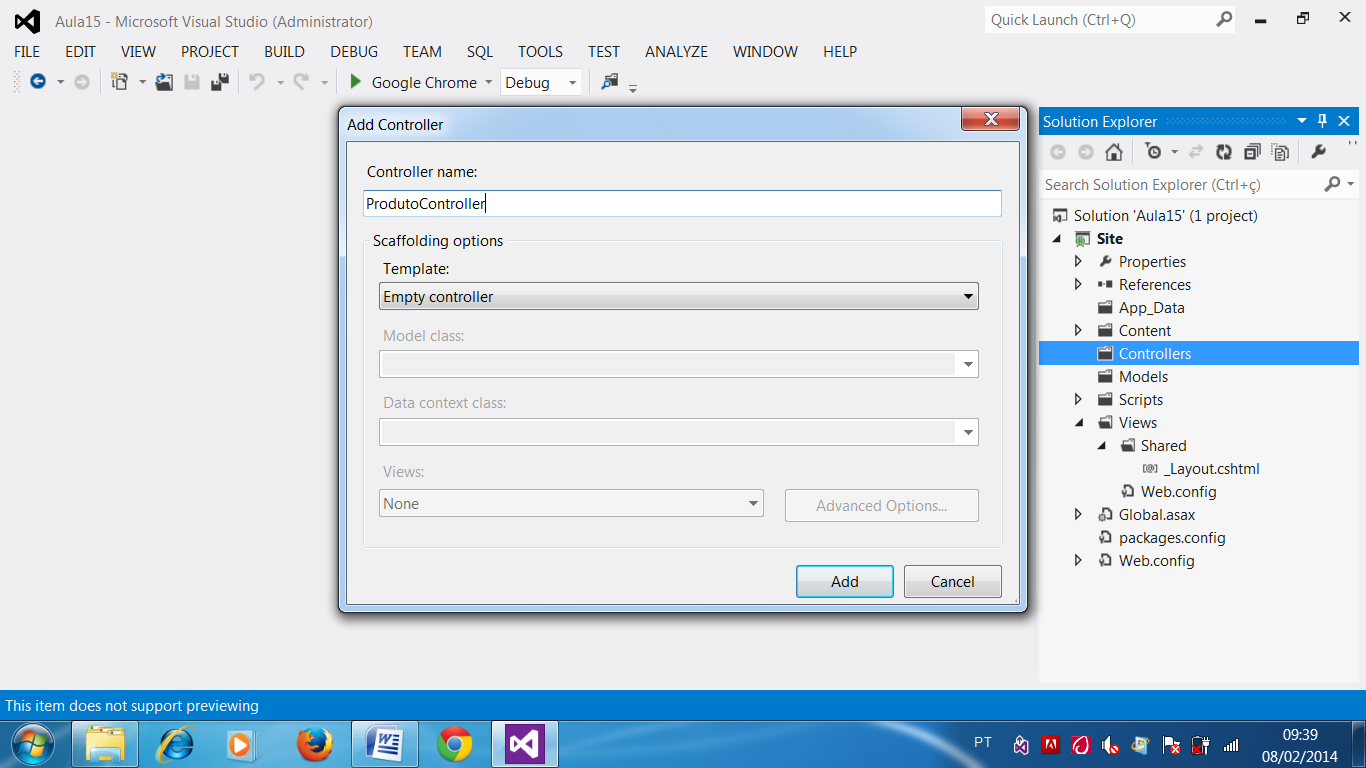
**@RenderBody()**

</div>

</body>

</html>





using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Site.Controllers

{

public class ProdutoController : Controller

{

//Página ser aberta! => /Produto/Cadastro

public ActionResult Cadastro() //Page\_Load

{

return View();

}

public ActionResult Pesquisa() //Page\_Load

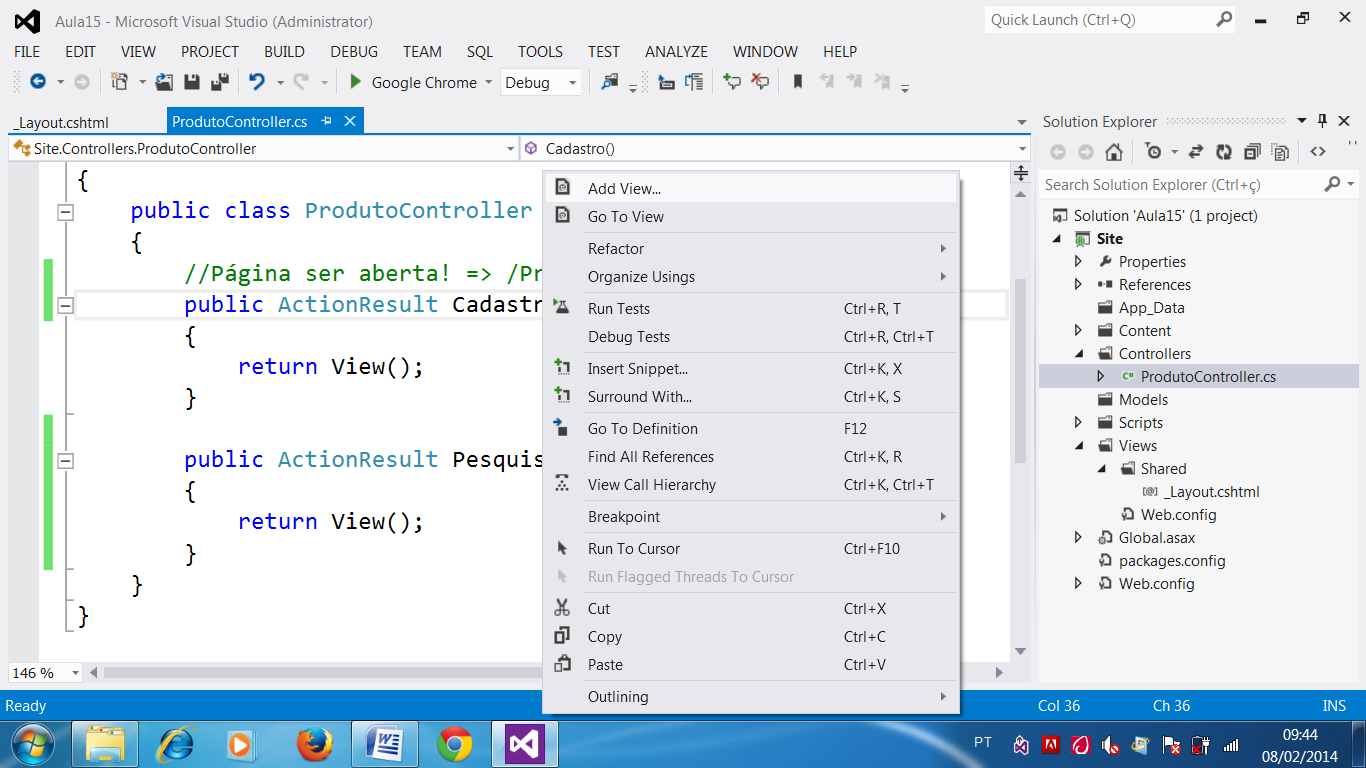
{

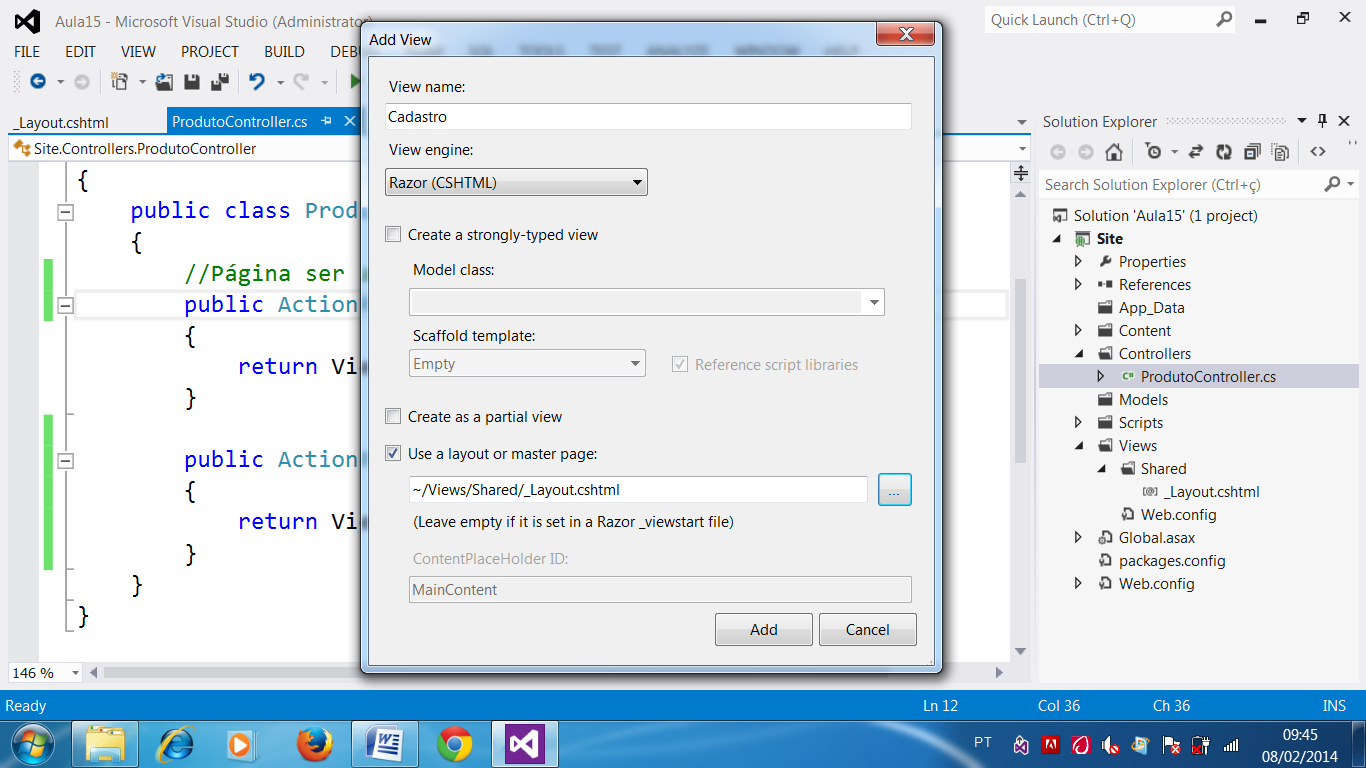
return View();

}

}

}



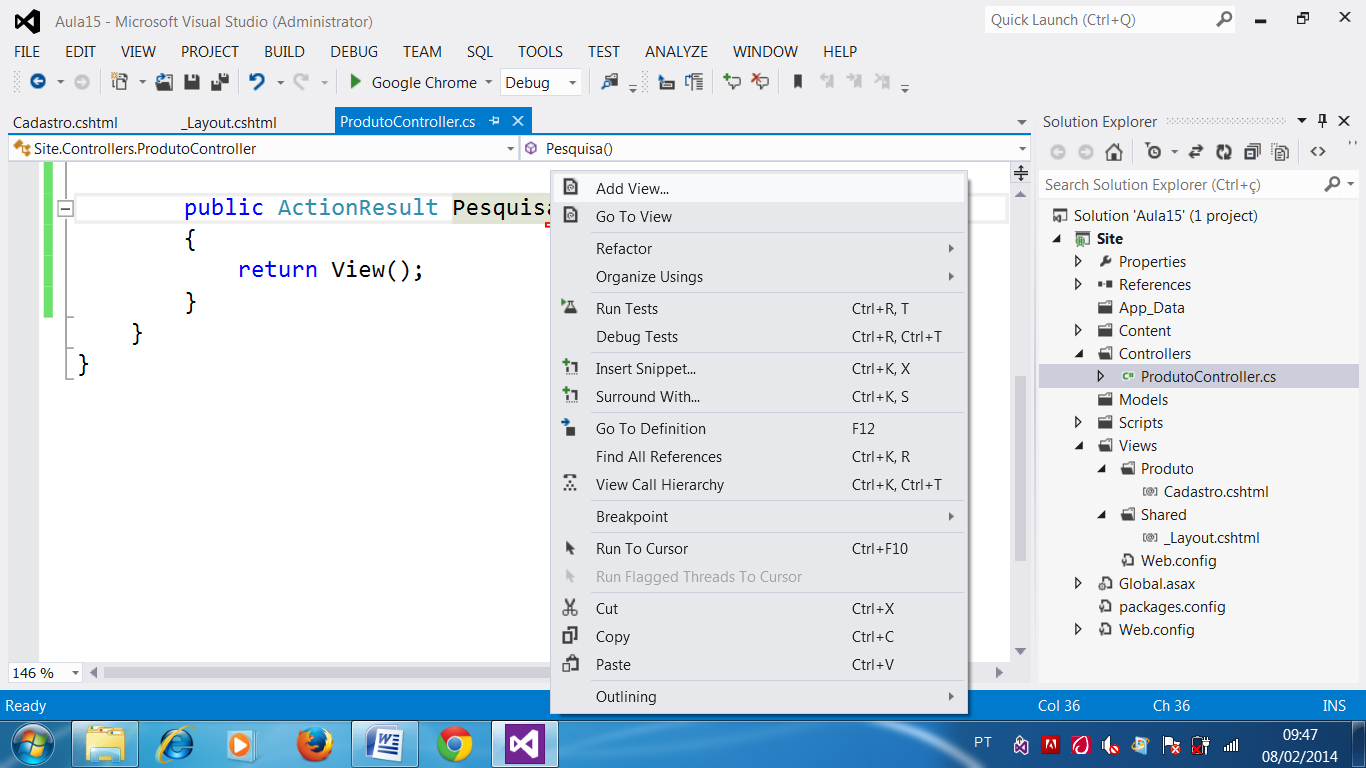


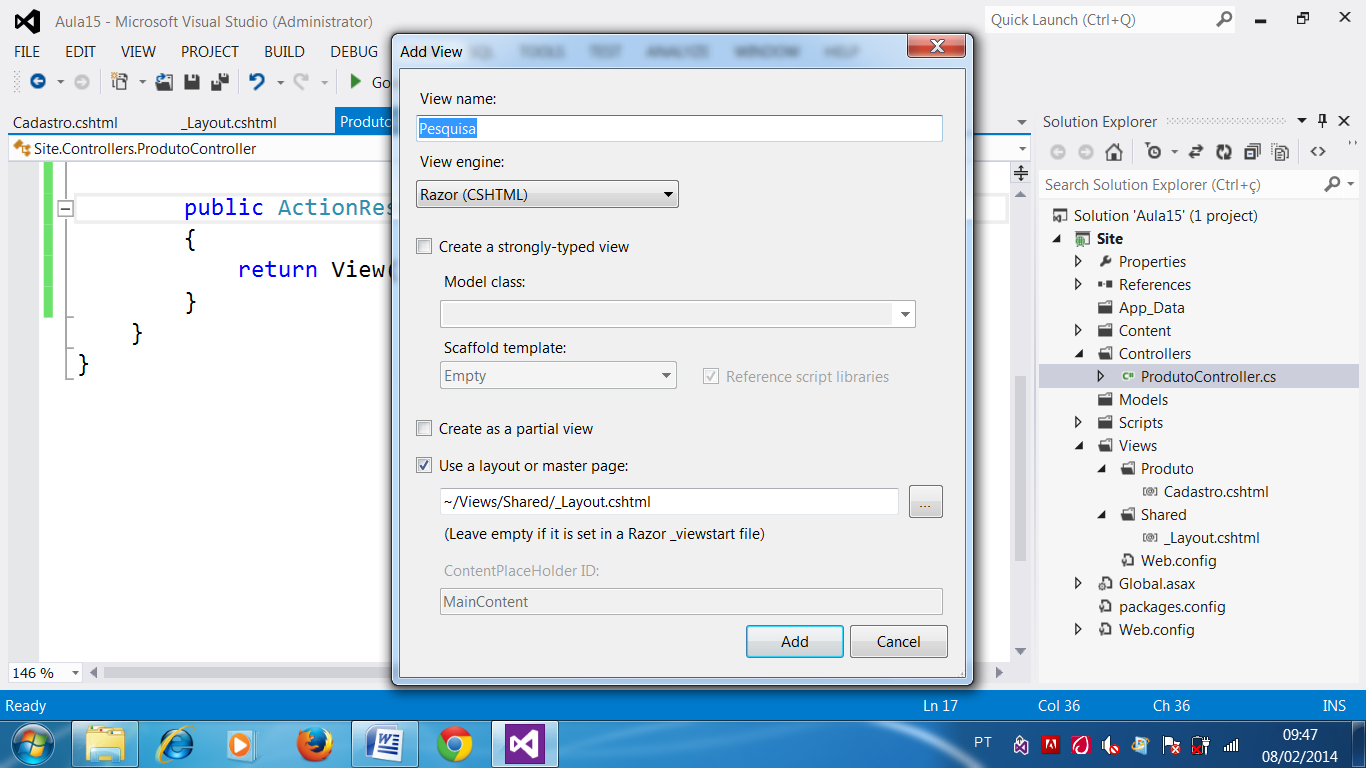
@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Cadastro de Produtos</h2>





@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Pesquisa de Produtos</h2>

Configurar a página inicial do projeto MVC

Global.asax

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Routing;

namespace Site

{

// Note: For instructions on enabling IIS6 or IIS7 classic mode,

// visit http://go.microsoft.com/?LinkId=9394801

public class MvcApplication : System.Web.HttpApplication

{

public static void RegisterGlobalFilters(GlobalFilterCollection filters)

{

filters.Add(new HandleErrorAttribute());

}

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

routes.MapRoute(

"Default", // Route name

"{controller}/{action}/{id}", // URL with parameters

new { **controller = "Produto", action = "Cadastro"**,

id = UrlParameter.Optional } // Parameter defaults

);

}

protected void Application\_Start()

{

AreaRegistration.RegisterAllAreas();

// Use LocalDB for Entity Framework by default

Database.DefaultConnectionFactory = new SqlConnectionFactory(@"Data Source=(localdb)\v11.0; Integrated Security=True; MultipleActiveResultSets=True");

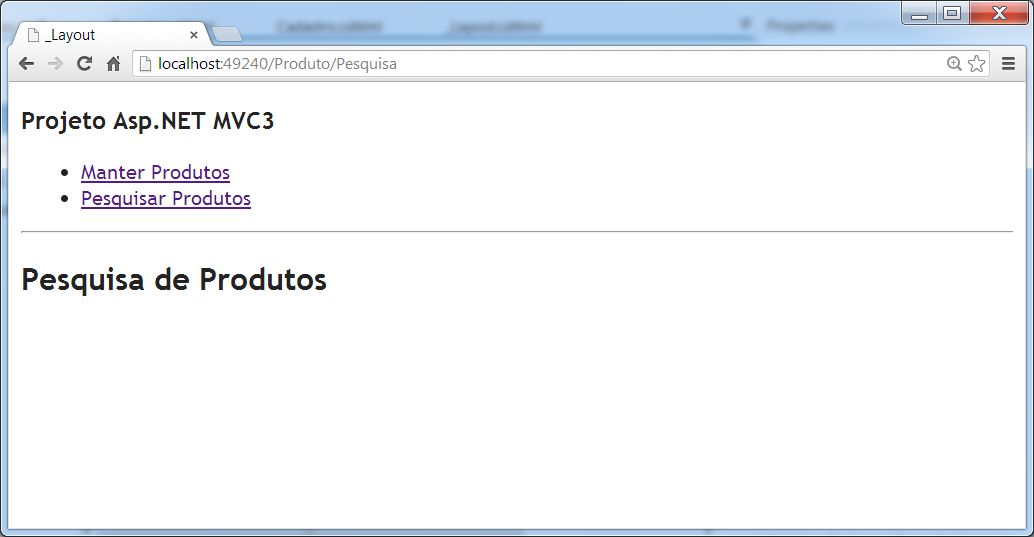
RegisterGlobalFilters(GlobalFilters.Filters);

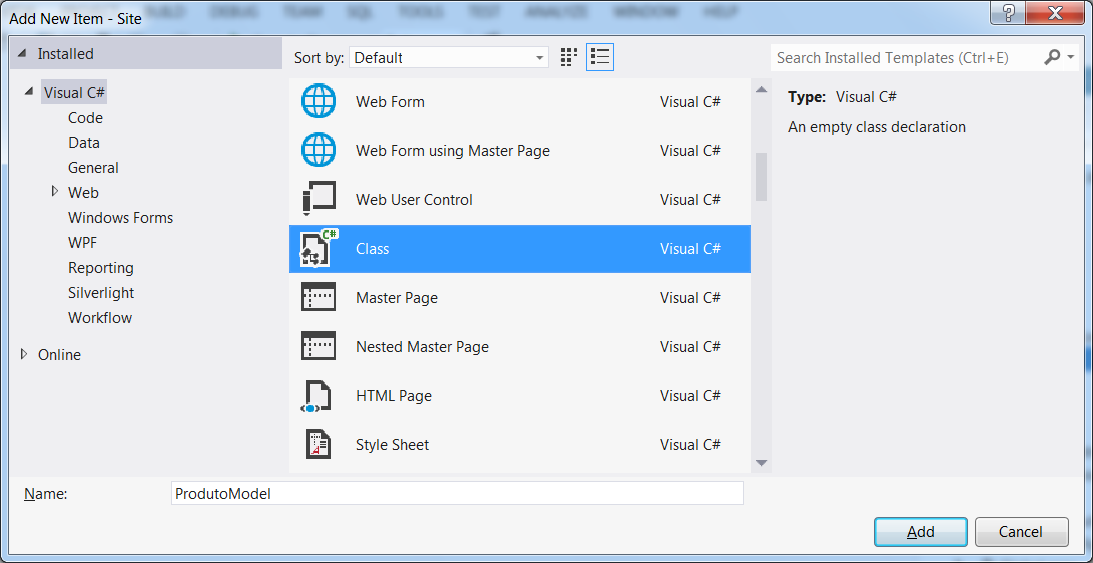
RegisterRoutes(RouteTable.Routes);

}

}

}





using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

namespace Site.Models

{

public class ProdutoCadastroModel

{

[Display(Name = "Nome do Produto:")]

public string Nome { get; set; }

[Display(Name = "Preço:")]

public double Preco { get; set; }

[Display(Name = "Quantidade:")]

public int Quantidade { get; set; }

[Display(Name = "Data de Compra:")]

public DateTime DataCompra { get; set; }

}

public class ProdutoConsultaModel

{

}

}

@model Site.Models.ProdutoCadastroModel

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Cadastro de Produtos</h2>

@using (Html.BeginForm())

{

@Html.LabelFor(p => p.Nome) <br />

@Html.TextBoxFor(p => p.Nome)

<br /><br />

@Html.LabelFor(p => p.Preco) <br />

@Html.TextBoxFor(p => p.Preco)

<br /><br />

@Html.LabelFor(p => p.Quantidade) <br />

@Html.TextBoxFor(p => p.Quantidade)

<br /><br />

@Html.LabelFor(p => p.DataCompra) <br />

@Html.TextBoxFor(p => p.DataCompra)

<br /><br />

<input type="submit" value="Cadastrar Produto" />

}



@model Site.Models.ProdutoCadastroModel

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Cadastro de Produtos</h2>

@using (Html.BeginForm())

{

@Html.LabelFor(p => p.Nome) <br />

@Html.TextBoxFor(p => p.Nome, new { @required = "required" } )

<br /><br />

@Html.LabelFor(p => p.Preco) <br />

@Html.TextBoxFor(p => p.Preco, new { @required = "required" })

<br /><br />

@Html.LabelFor(p => p.Quantidade) <br />

@Html.TextBoxFor(p => p.Quantidade, new { @required = "required",

@type = "number", @min = "1" })

<br /><br />

@Html.LabelFor(p => p.DataCompra) <br />

@Html.TextBoxFor(p => p.DataCompra, new { @required = "required",

@type = "date" })

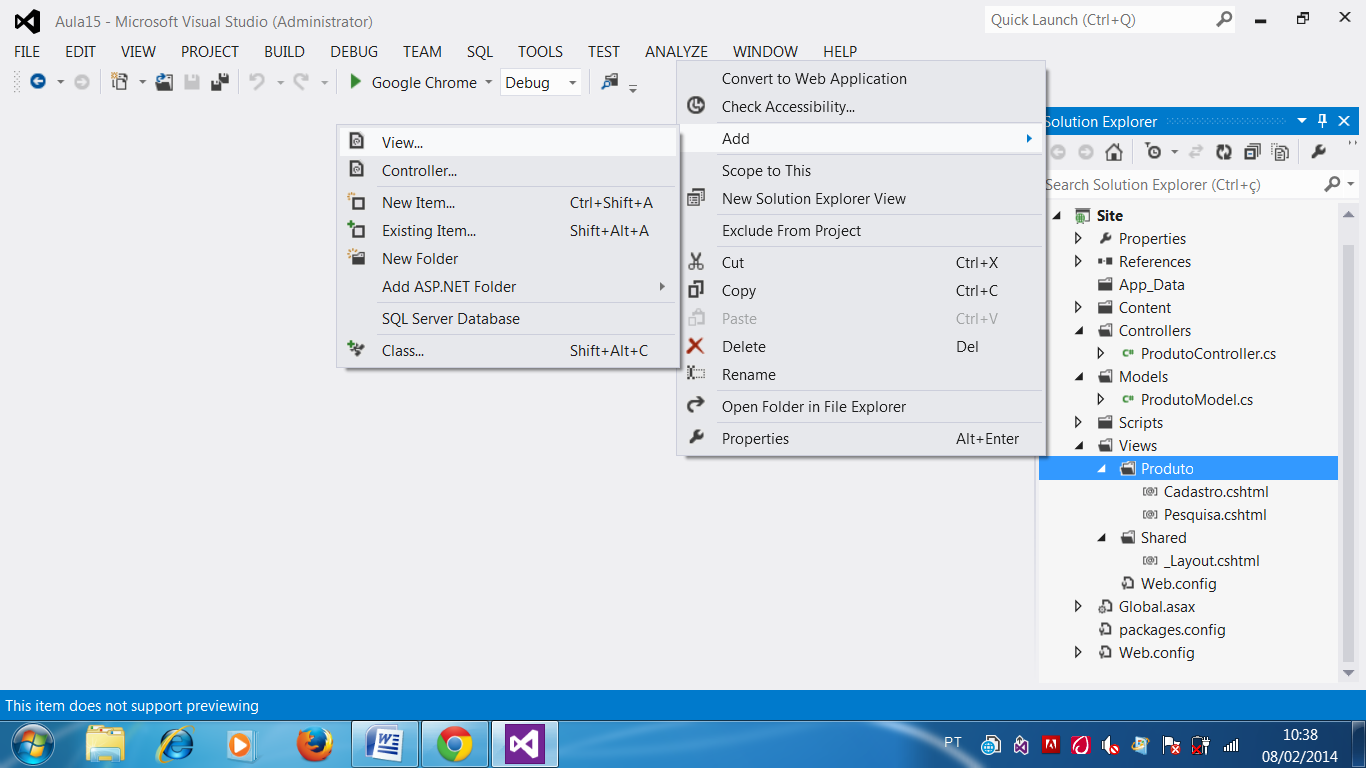
<br /><br />

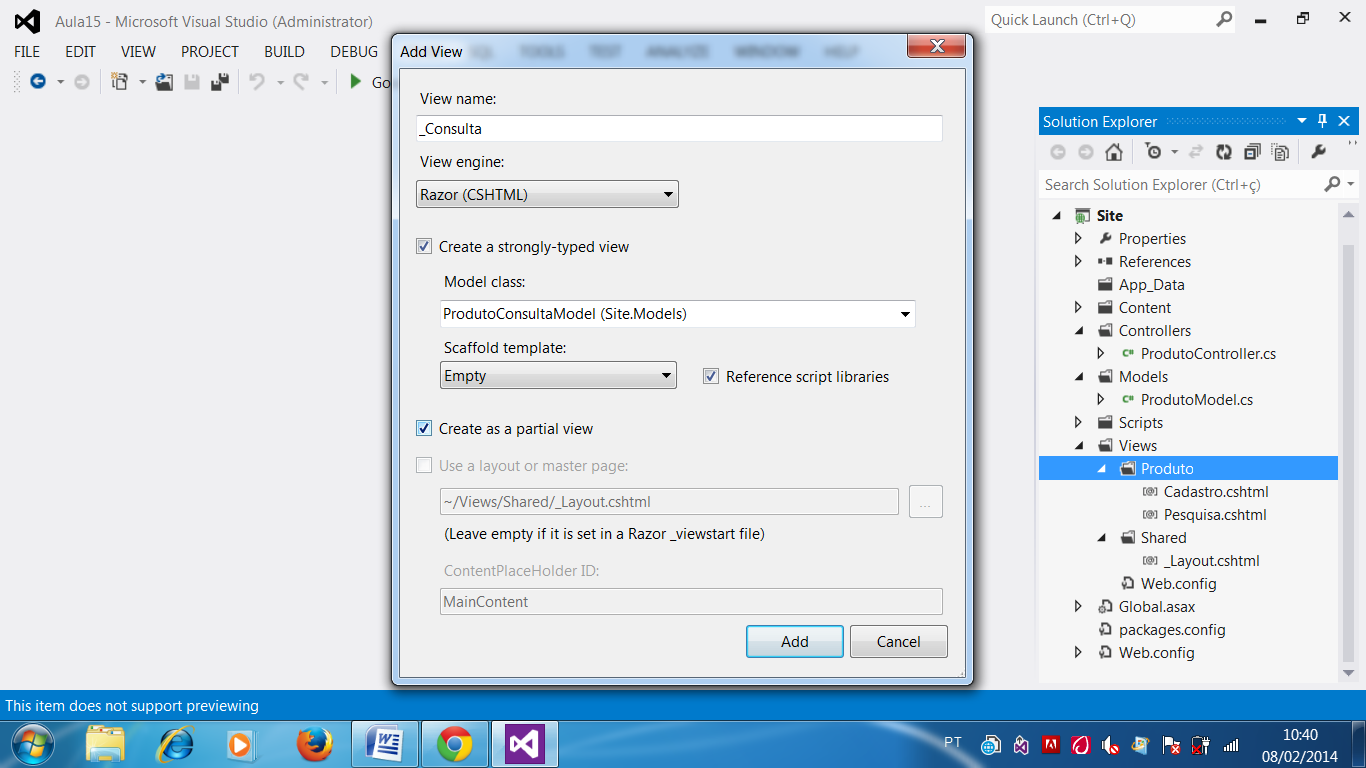
<input type="submit" value="Cadastrar Produto" />

}

PartialView

Página .CSHTML que será exibida dentro de outra página...





@model Site.Models.ProdutoConsultaModel

<p>

Relação de Produtos cadastrados

</p>

<table border="1" style="width: 100px">

<thead>

<tr>

<th>Código</th>

<th>Nome do Produto</th>

<th>Preço</th>

<th>Quantidade</th>

<th>Total</th>

<th>Data de Compra</th>

</tr>

</thead>

<tbody>

</tbody>

<tfoot>

<tr>

<td colspan="6">Quantidade de Produtos:</td>

</tr>

</tfoot>

</table>

@model Site.Models.ProdutoCadastroModel

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Cadastro de Produtos</h2>

@using (Html.BeginForm())

{

@Html.LabelFor(p => p.Nome) <br />

@Html.TextBoxFor(p => p.Nome, new { @required = "required" } )

<br /><br />

@Html.LabelFor(p => p.Preco) <br />

@Html.TextBoxFor(p => p.Preco, new { @required = "required" })

<br /><br />

@Html.LabelFor(p => p.Quantidade) <br />

@Html.TextBoxFor(p => p.Quantidade, new { @required = "required", @type = "number", @min = "1" })

<br /><br />

@Html.LabelFor(p => p.DataCompra) <br />

@Html.TextBoxFor(p => p.DataCompra, new { @required = "required", @type = "date" })

<br /><br />

<input type="submit" value="Cadastrar Produto" />

<p>

@ViewBag.Mensagem

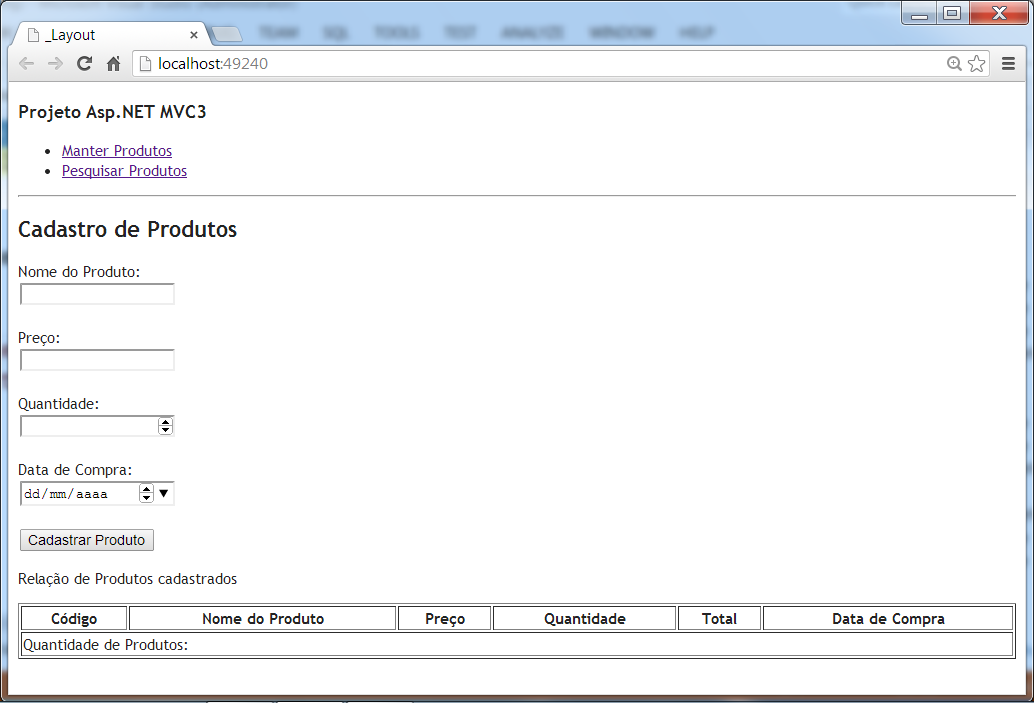
</p>

<!-- Incluir o conteúdo da página de Consulta -->

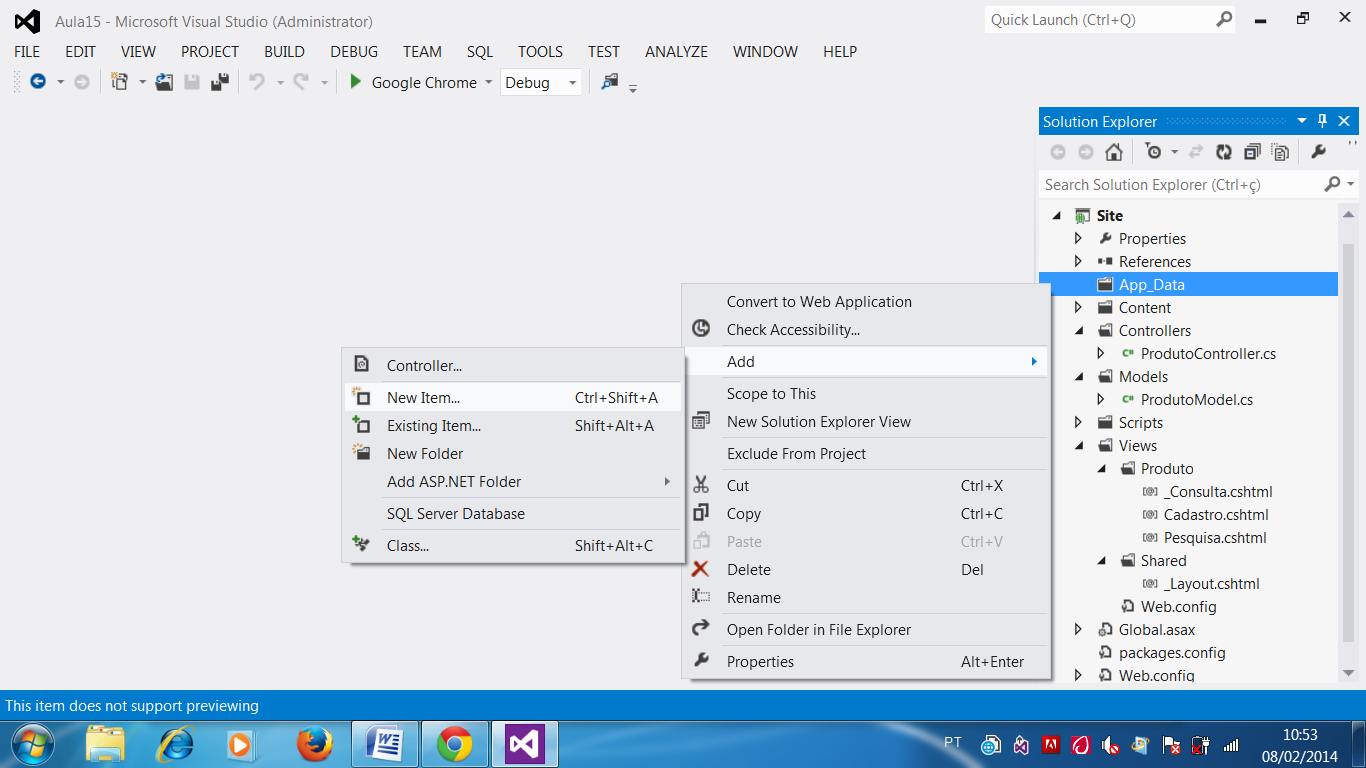
**@Html.Partial("\_Consulta")**

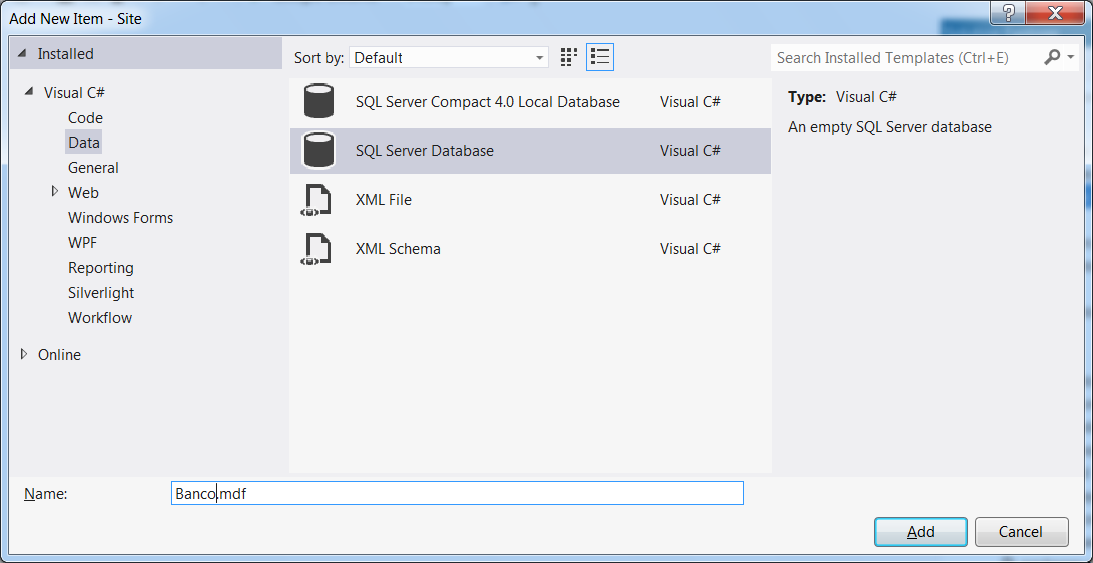
}

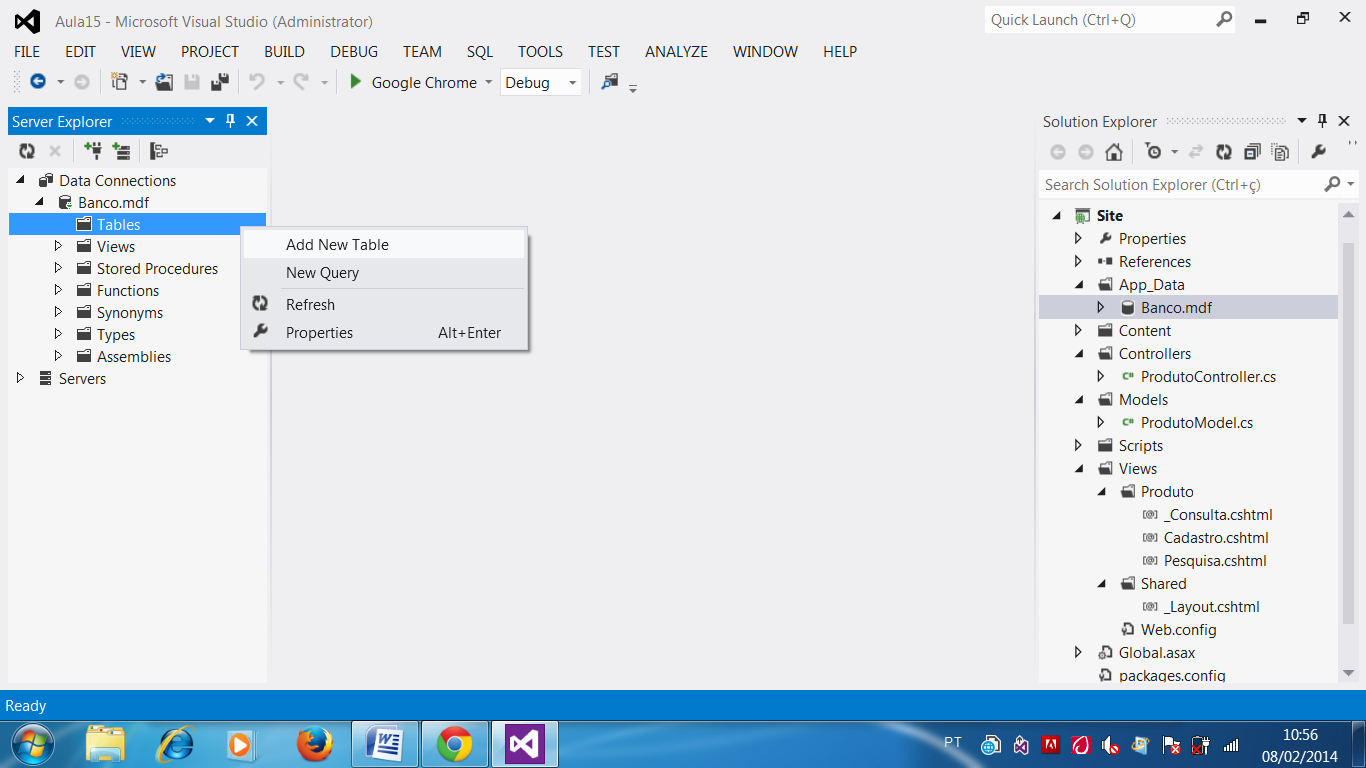
Resultado...



Criando a base de dados...







go

create table Produto(

IdProduto integer identity,

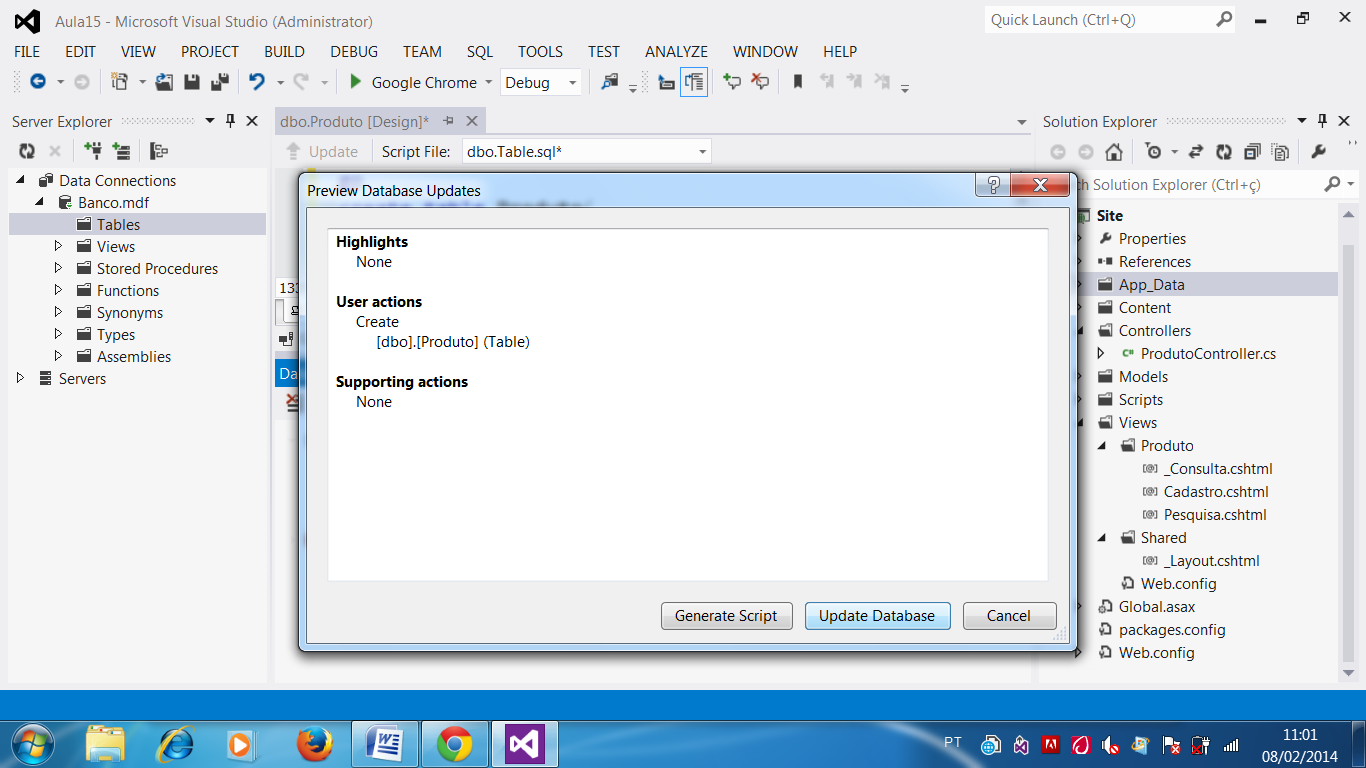
Nome nvarchar(50) not null,

Preco numeric(10,2) not null,

Quantidade integer not null,

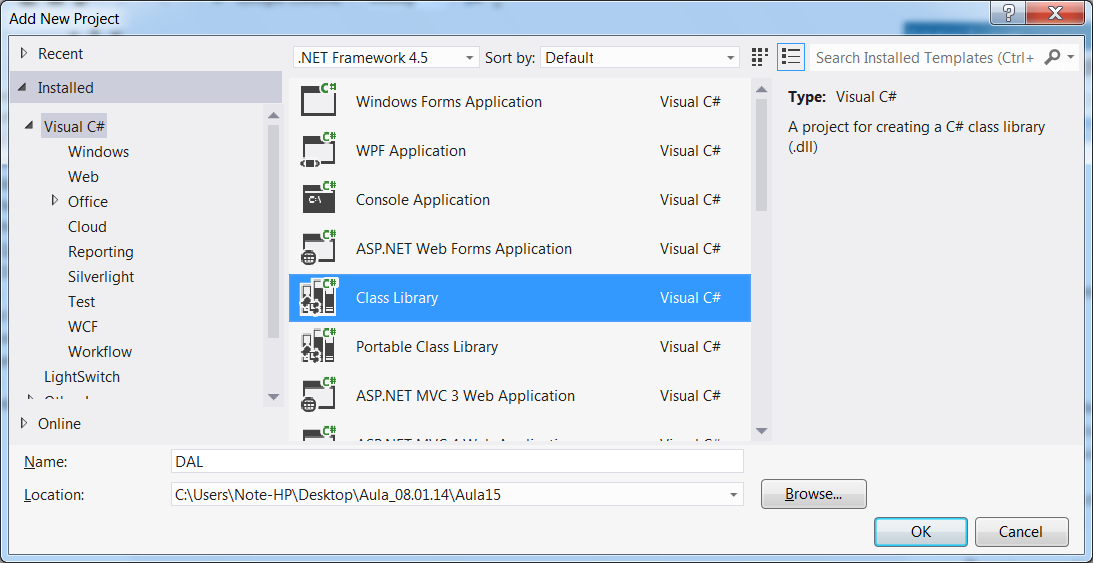
DataCompra date not null,

primary key(IdProduto))

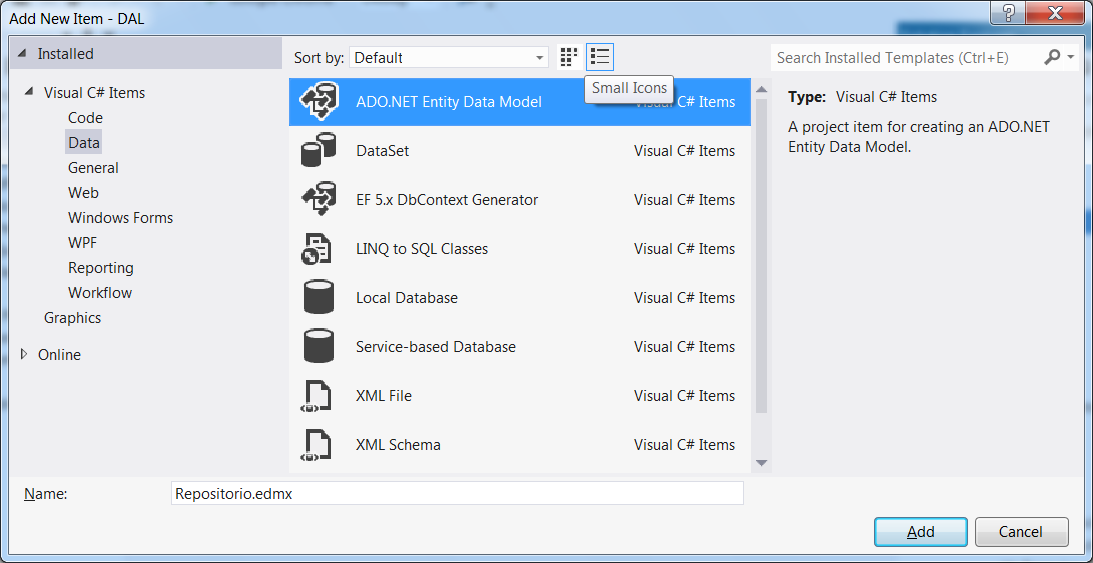


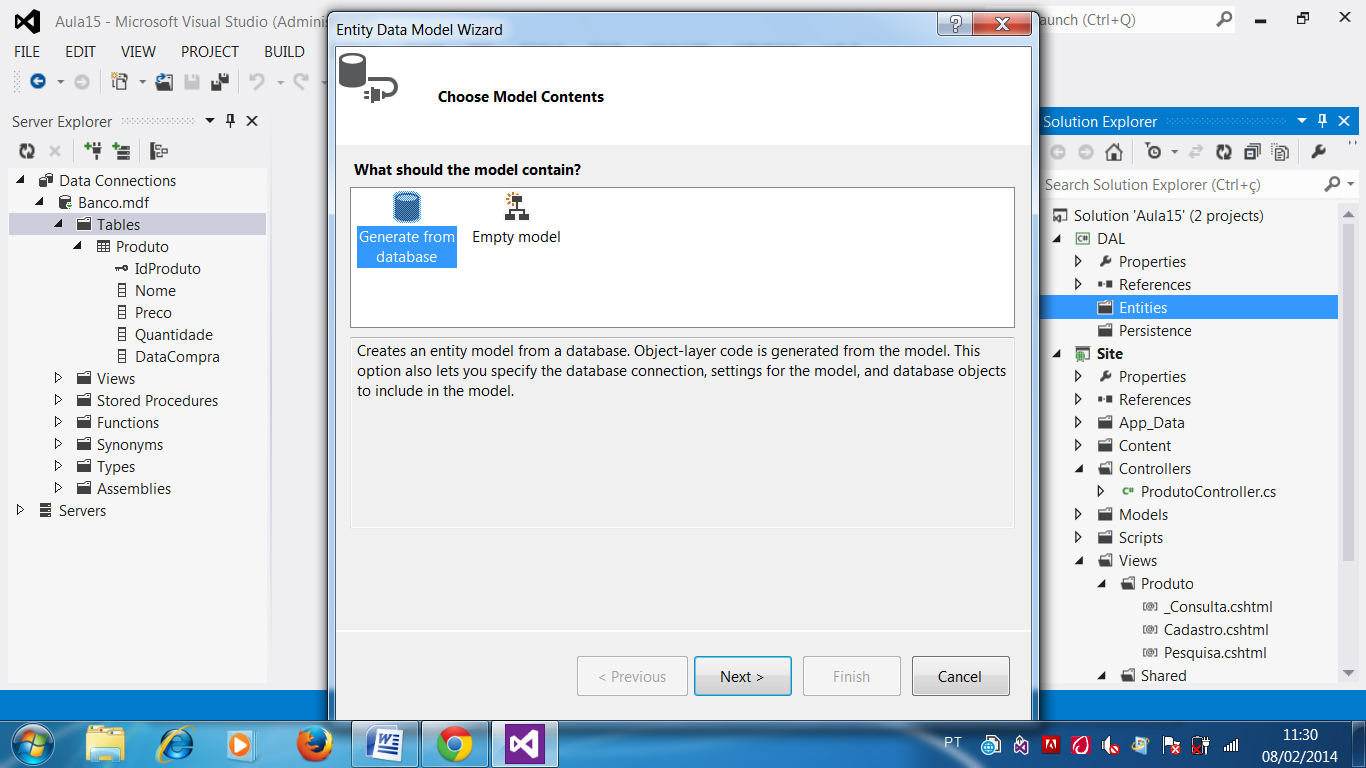
Data Access Layer (DAL)

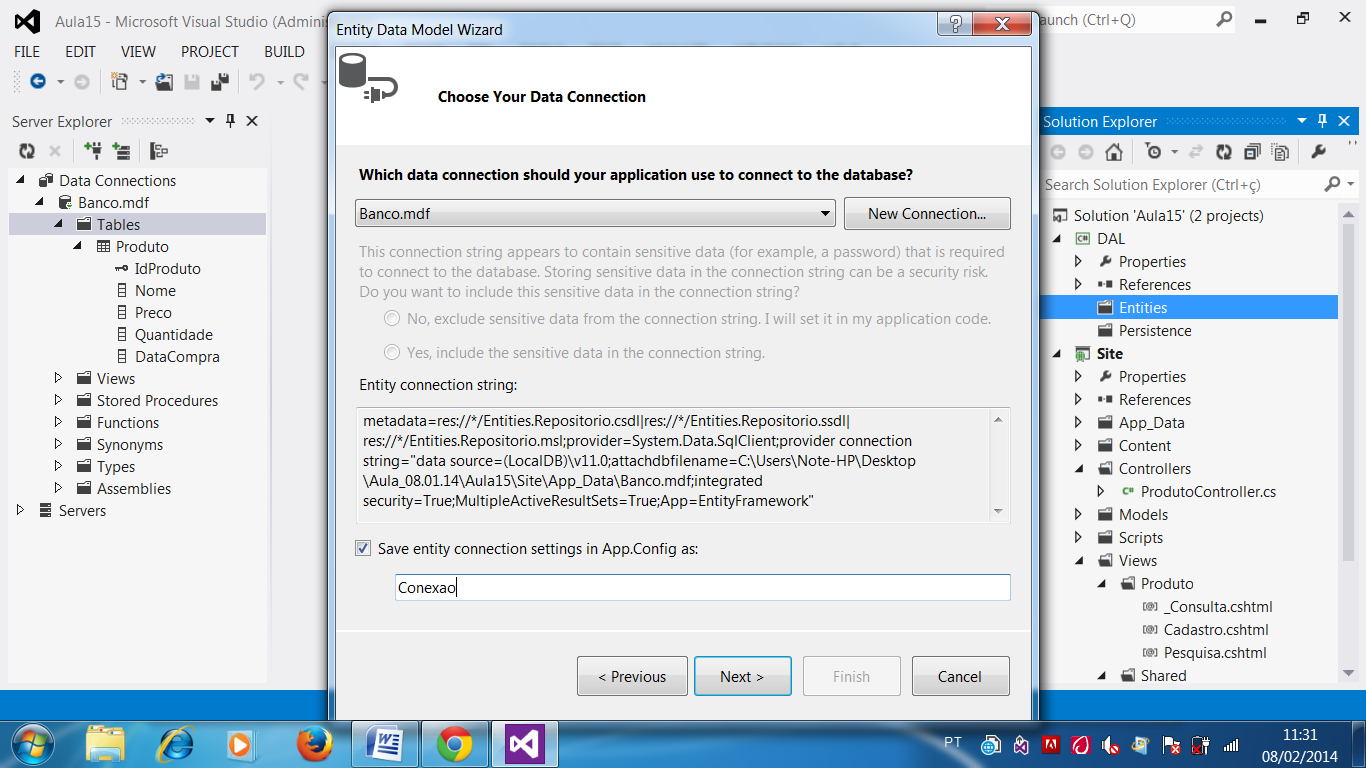
Camada de acesso a dados

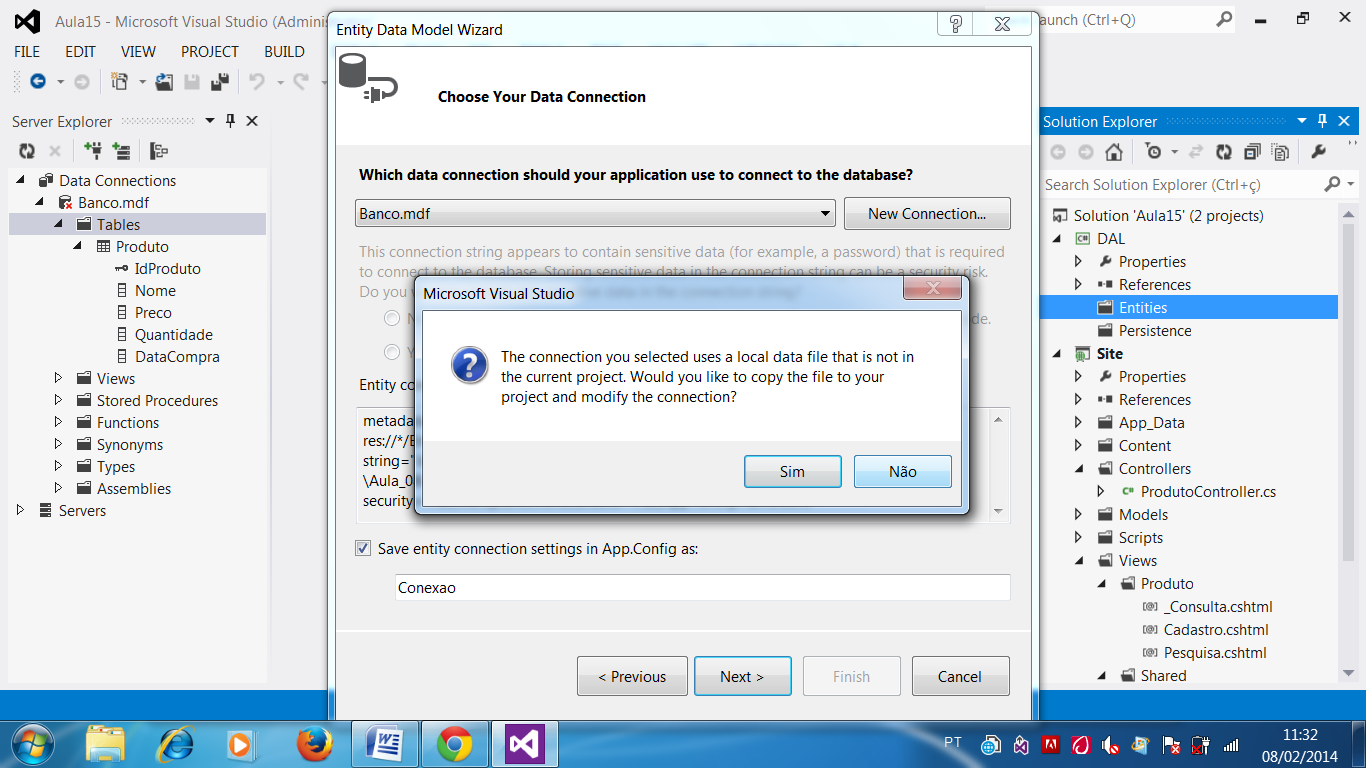


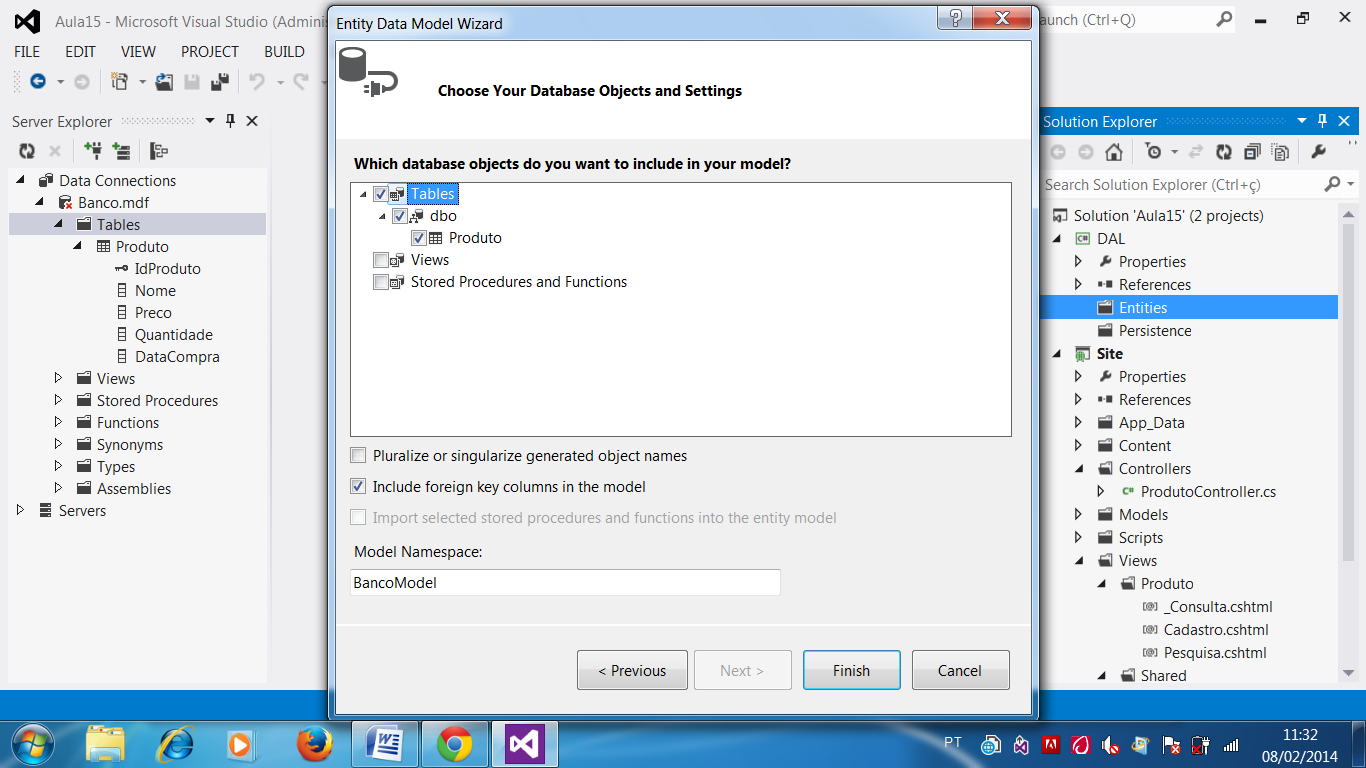
Entity Data Model (.edmx)

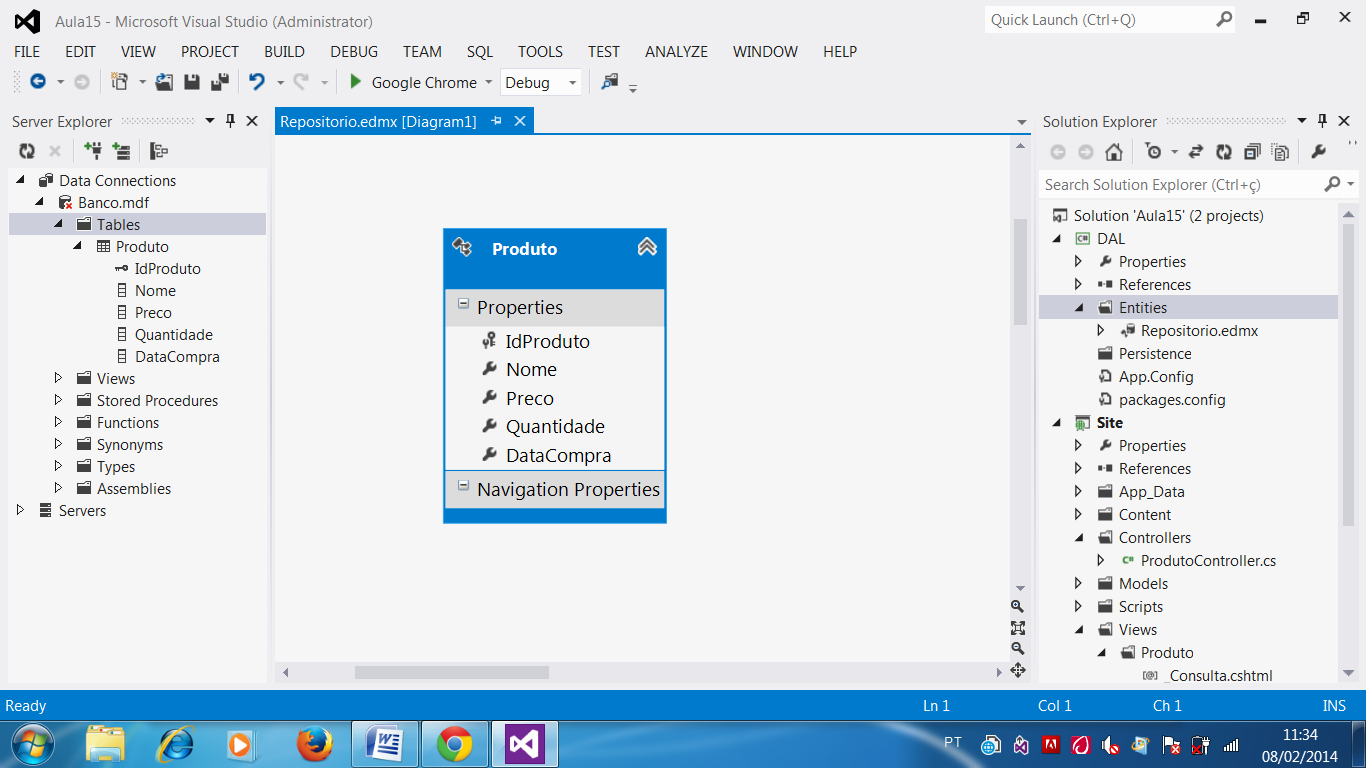


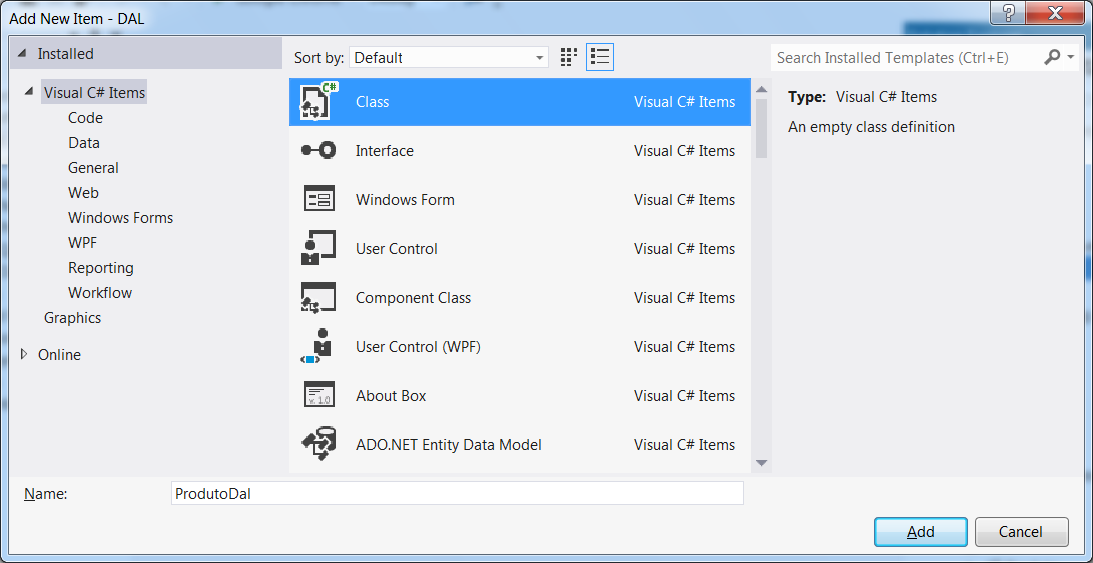












using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DAL.Entities;

namespace DAL.Persistence

{

public class ProdutoDal : IDisposable

{

private Conexao Con; //atributo null

//Construtor [ctor] + 2x[tab]

public ProdutoDal()

{

Con = new Conexao(); //inicializando

}

public void Salvar(Produto p)

{

try

{

Con.Produto.Add(p); //insert

Con.SaveChanges();

}

catch(Exception e)

{

throw new Exception("Erro ao salvar produto: " + e.Message);

}

}

public List<Produto> ListarTodos()

{

try

{

//SQL => select \* from Produto order by DataCompra desc

return Con.Produto.OrderByDescending(p => p.DataCompra).ToList();

}

catch(Exception e)

{

throw new Exception("Erro ao listar produtos: " + e.Message);

}

}

public void Dispose()

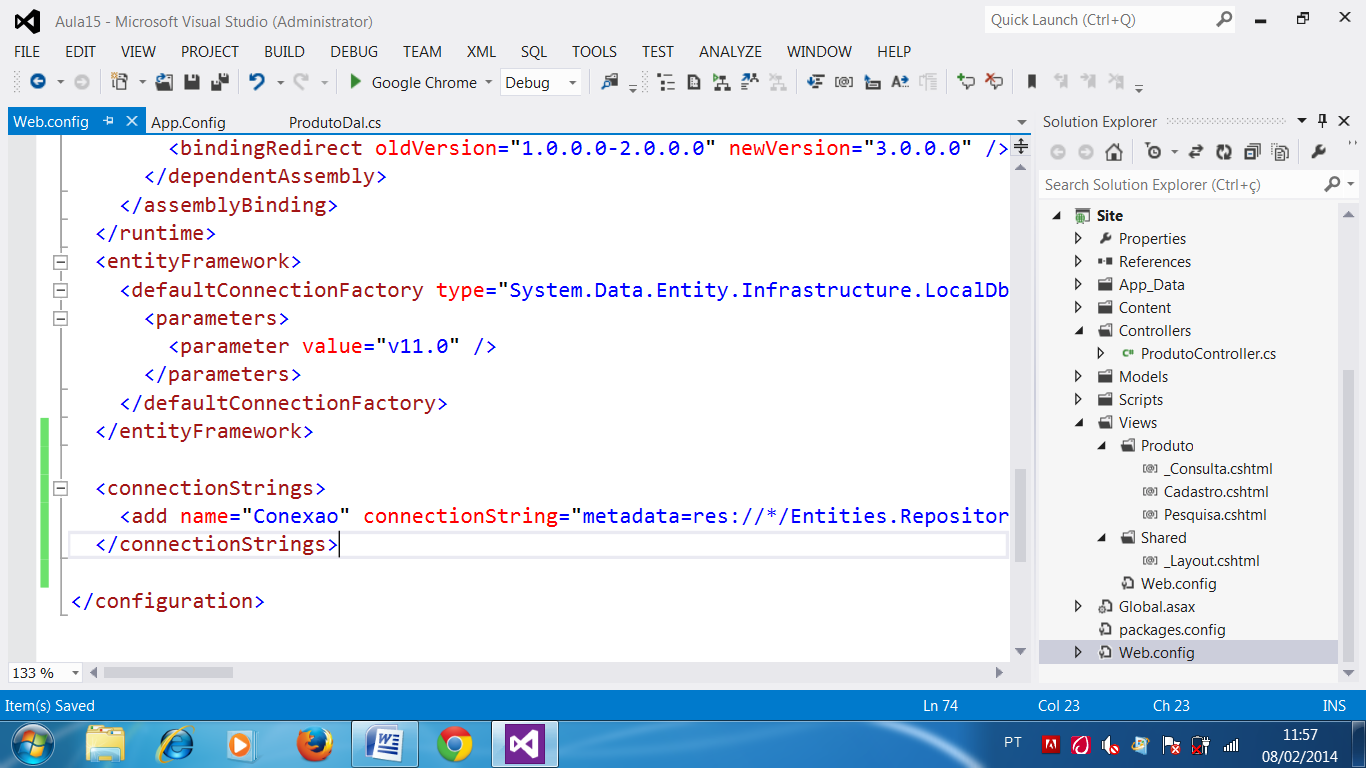
{

Con.Dispose(); //destruir a conexão

}

}

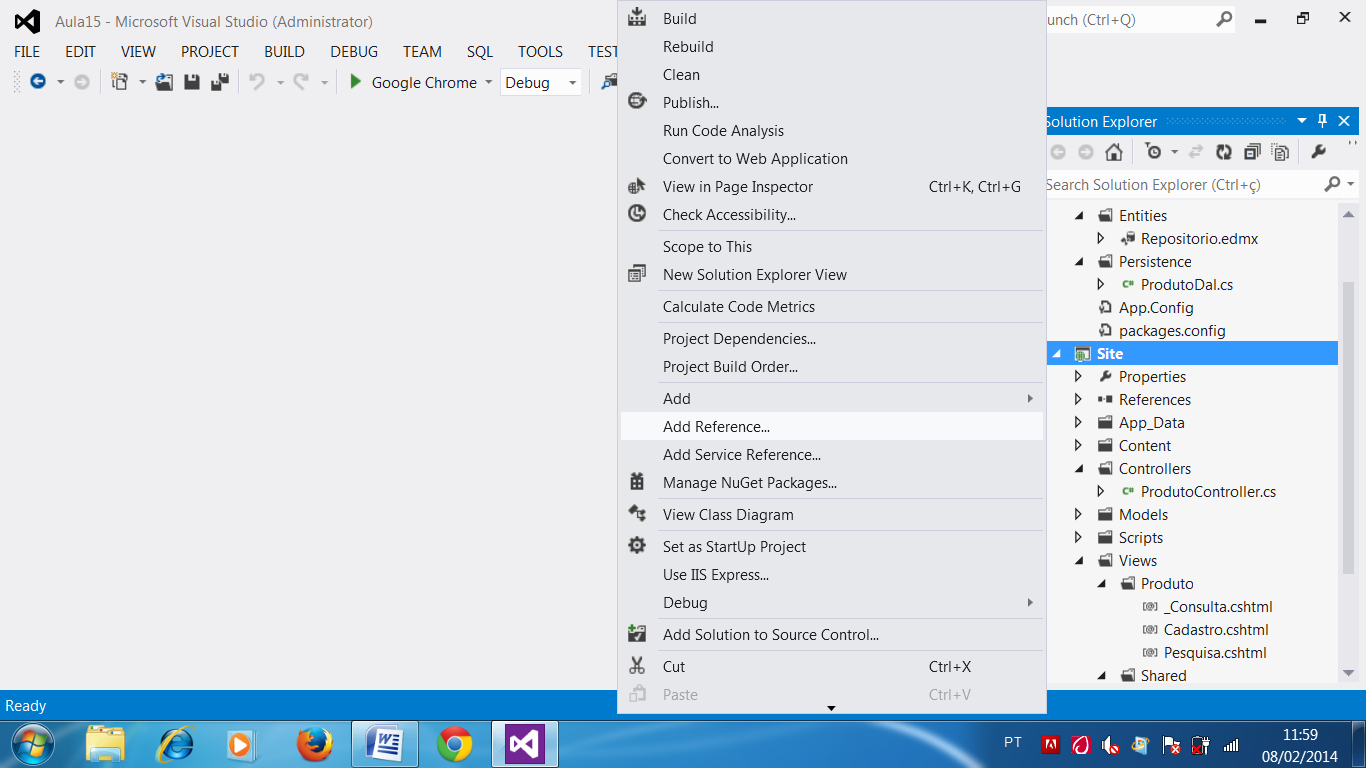
}

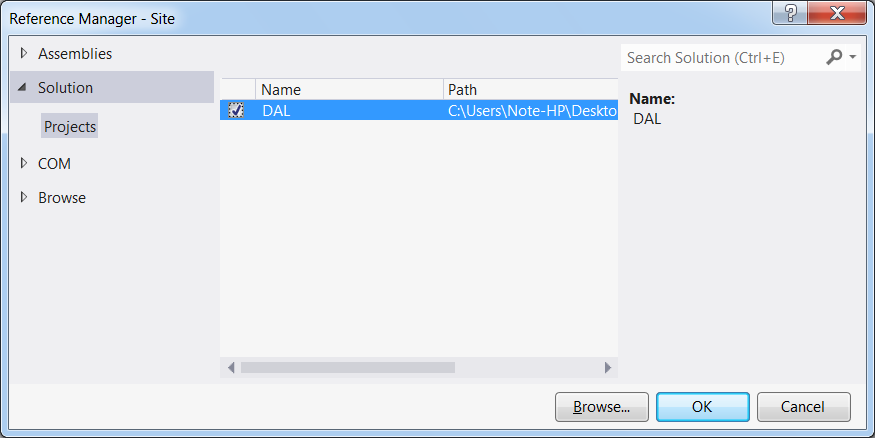


<connectionStrings>

**<add name="Conexao"** connectionString="metadata=res://\*/Entities.Repositorio.csdl|res://\*/Entities.Repositorio.ssdl|res://\*/Entities.Repositorio.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=(LocalDB)\v11.0;attachdbfilename=C:\Users\Note-HP\Desktop\Aula\_08.01.14\Aula15\Site\App\_Data\Banco.mdf;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

</connectionStrings>





**@model Site.Models.ProdutoCadastroModel**

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Cadastro de Produtos</h2>

**@using (Html.BeginForm("CadastrarProduto", "Produto", FormMethod.Post))**

{

@Html.LabelFor(p => p.Nome) <br />

@Html.TextBoxFor(p => p.Nome, new { @required = "required" } )

<br /><br />

@Html.LabelFor(p => p.Preco) <br />

@Html.TextBoxFor(p => p.Preco, new { @required = "required" })

<br /><br />

@Html.LabelFor(p => p.Quantidade) <br />

@Html.TextBoxFor(p => p.Quantidade, new { @required = "required", @type = "number", @min = "1" })

<br /><br />

@Html.LabelFor(p => p.DataCompra) <br />

@Html.TextBoxFor(p => p.DataCompra, new { @required = "required", @type = "date" })

<br /><br />

<input type="submit" value="Cadastrar Produto" />

<p>

@ViewBag.Mensagem

</p>

<!-- Incluir o conteúdo da página de Consulta -->

@Html.Partial("\_Consulta")

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using DAL.Entities;

using DAL.Persistence;

using Site.Models;

namespace Site.Controllers

{

public class ProdutoController : Controller

{

//Página ser aberta! => /Produto/Cadastro

public ActionResult Cadastro() //Page\_Load

{

return View();

}

public ActionResult Pesquisa() //Page\_Load

{

return View();

}

**//Método para cadastrar produto obtido pelo formulário**

**[HttpPost] //método chamado a partir de um formulário FormMethod.Post**

**public ActionResult CadastrarProduto(ProdutoCadastroModel Model)**

**{**

**try**

**{**

**Produto p = new Produto();**

**p.Nome = Model.Nome;**

**p.Preco = Model.Preco;**

**p.Quantidade = Model.Quantidade;**

**p.DataCompra = Model.DataCompra;**

**using(ProdutoDal d = new ProdutoDal())**

**{**

**d.Salvar(p);**

**ViewBag.Mensagem = "Produto cadastrado com sucesso.";**

**ModelState.Clear(); //limpar o formulário**

**}**

**}**

**catch(Exception e)**

**{**

**ViewBag.Mensagem = e.Message;**

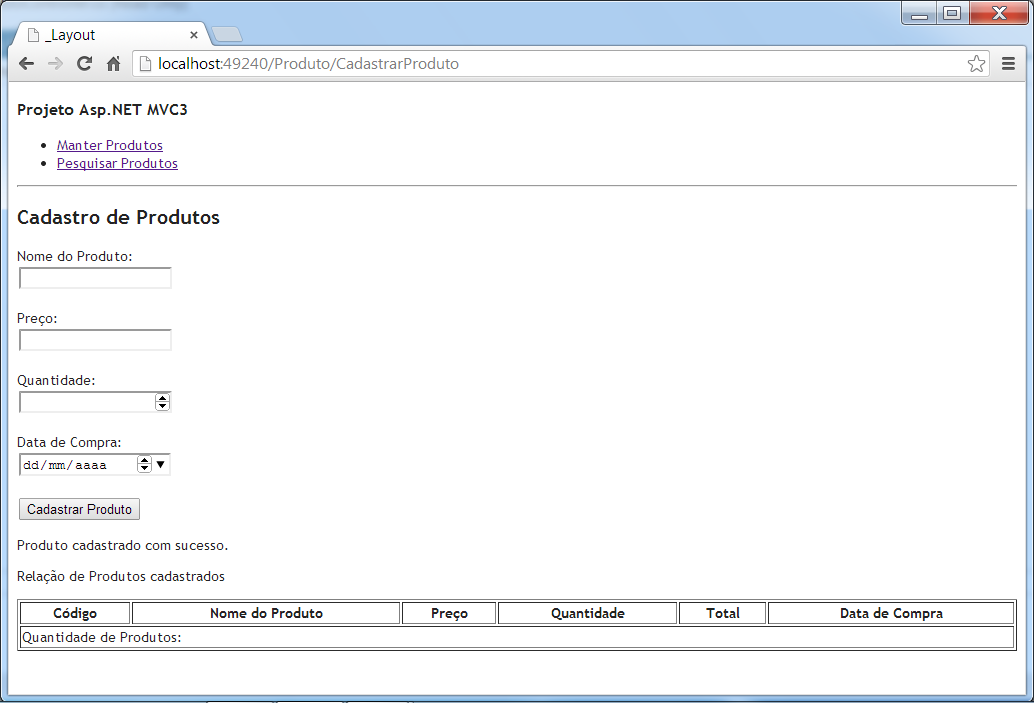
**}**

**return View("Cadastro"); //redirecionamento**

**}**

}

}



using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

using DAL.Entities;

namespace Site.Models

{

public class ProdutoCadastroModel

{

[Display(Name = "Nome do Produto:")]

public string Nome { get; set; }

[Display(Name = "Preço:")]

public decimal Preco { get; set; }

[Display(Name = "Quantidade:")]

public int Quantidade { get; set; }

[Display(Name = "Data de Compra:")]

public DateTime DataCompra { get; set; }

}

**public class ProdutoConsultaModel**

**{**

**//Listagem dos Produtos que serão obtidos do banco**

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

**public int QtdProdutos { get; set; }**

**}**

}

ProdutoDal

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DAL.Entities;

namespace DAL.Persistence

{

public class ProdutoDal : IDisposable

{

private Conexao Con; //atributo null

//Construtor [ctor] + 2x[tab]

public ProdutoDal()

{

Con = new Conexao(); //inicializando

}

public void Salvar(Produto p)

{

try

{

Con.Produto.Add(p); //insert

Con.SaveChanges();

}

catch(Exception e)

{

throw new Exception("Erro ao salvar produto: " + e.Message);

}

}

public List<Produto> ListarTodos()

{

try

{

//SQL => select \* from Produto order by DataCompra desc

return Con.Produto.OrderByDescending(p => p.DataCompra).ToList();

}

catch(Exception e)

{

throw new Exception("Erro ao listar produtos: " + e.Message);

}

}

**public int ObterQuantidadeRegistros()**

**{**

**try**

**{**

**//SQL => select count(\*) from Produto**

**return Con.Produto.Count();**

**}**

**catch(Exception e)**

**{**

**throw new Exception("Erro ao obter quantidade de registros: "**

**+ e.Message);**

**}**

**}**

public void Dispose()

{

Con.Dispose(); //destruir a conexão

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using DAL.Entities;

using DAL.Persistence;

using Site.Models;

namespace Site.Controllers

{

public class ProdutoController : Controller

{

//Página ser aberta! => /Produto/Cadastro

public ActionResult **Cadastro**() //Page\_Load

{

**ViewData["dados"] = ObterDadosConsulta();**

return View();

}

public ActionResult Pesquisa() //Page\_Load

{

return View();

}

//Método para cadastrar produto obtido pelo formulário

[HttpPost] //método chamado a partir de um formulário FormMethod.Post

public ActionResult CadastrarProduto(ProdutoCadastroModel Model)

{

try

{

Produto p = new Produto();

p.Nome = Model.Nome;

p.Preco = Model.Preco;

p.Quantidade = Model.Quantidade;

p.DataCompra = Model.DataCompra;

using(ProdutoDal d = new ProdutoDal())

{

d.Salvar(p);

ViewBag.Mensagem = "Produto cadastrado com sucesso.";

ModelState.Clear(); //limpar o formulário

}

}

catch(Exception e)

{

ViewBag.Mensagem = e.Message;

}

**ViewData["dados"] = ObterDadosConsulta();**

return View("Cadastro"); //redirecionamento

}

//Método para produzir conteudo na Classe ProdutoConsultaModel

private ProdutoConsultaModel ObterDadosConsulta()

{

try

{

using(ProdutoDal d = new ProdutoDal())

{

ProdutoConsultaModel Model = new ProdutoConsultaModel();

Model.Produtos = d.ListarTodos();

Model.QtdProdutos = d.ObterQuantidadeRegistros();

return Model;

}

}

catch(Exception e)

{

ViewBag.Mensagem = e.Message;

return null; //retorna vazio

}

}

}

}

@model Site.Models.ProdutoCadastroModel

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Cadastro de Produtos</h2>

@using (Html.BeginForm("CadastrarProduto", "Produto", FormMethod.Post))

{

@Html.LabelFor(p => p.Nome) <br />

@Html.TextBoxFor(p => p.Nome, new { @required = "required" } )

<br /><br />

@Html.LabelFor(p => p.Preco) <br />

@Html.TextBoxFor(p => p.Preco, new { @required = "required" })

<br /><br />

@Html.LabelFor(p => p.Quantidade) <br />

@Html.TextBoxFor(p => p.Quantidade, new { @required = "required", @type = "number", @min = "1" })

<br /><br />

@Html.LabelFor(p => p.DataCompra) <br />

@Html.TextBoxFor(p => p.DataCompra, new { @required = "required", @type = "date" })

<br /><br />

<input type="submit" value="Cadastrar Produto" />

<p>

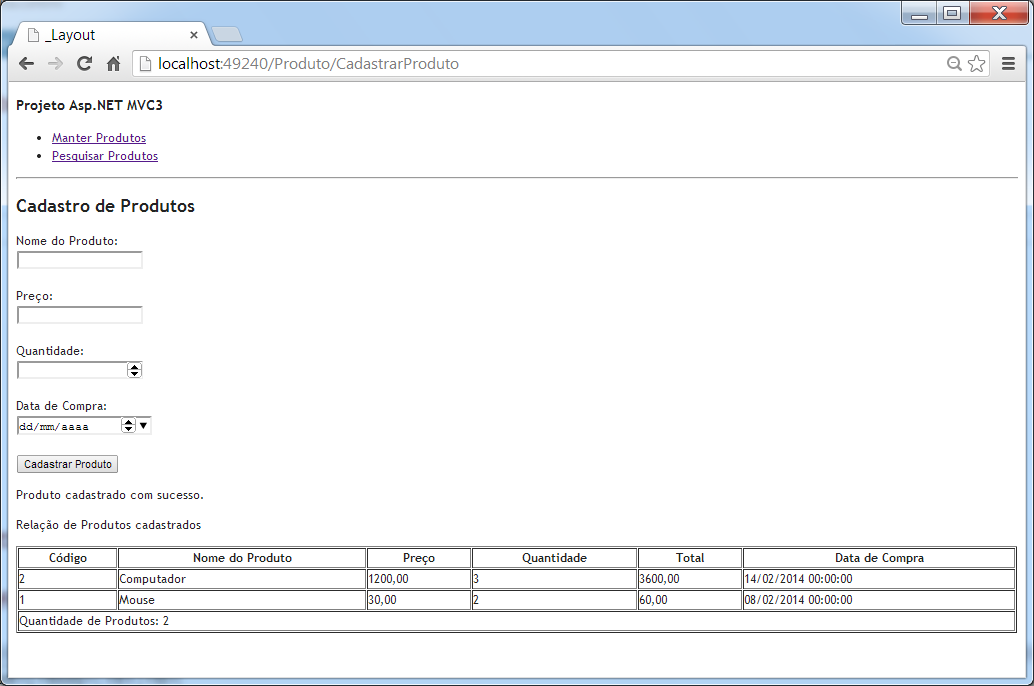
@ViewBag.Mensagem

</p>

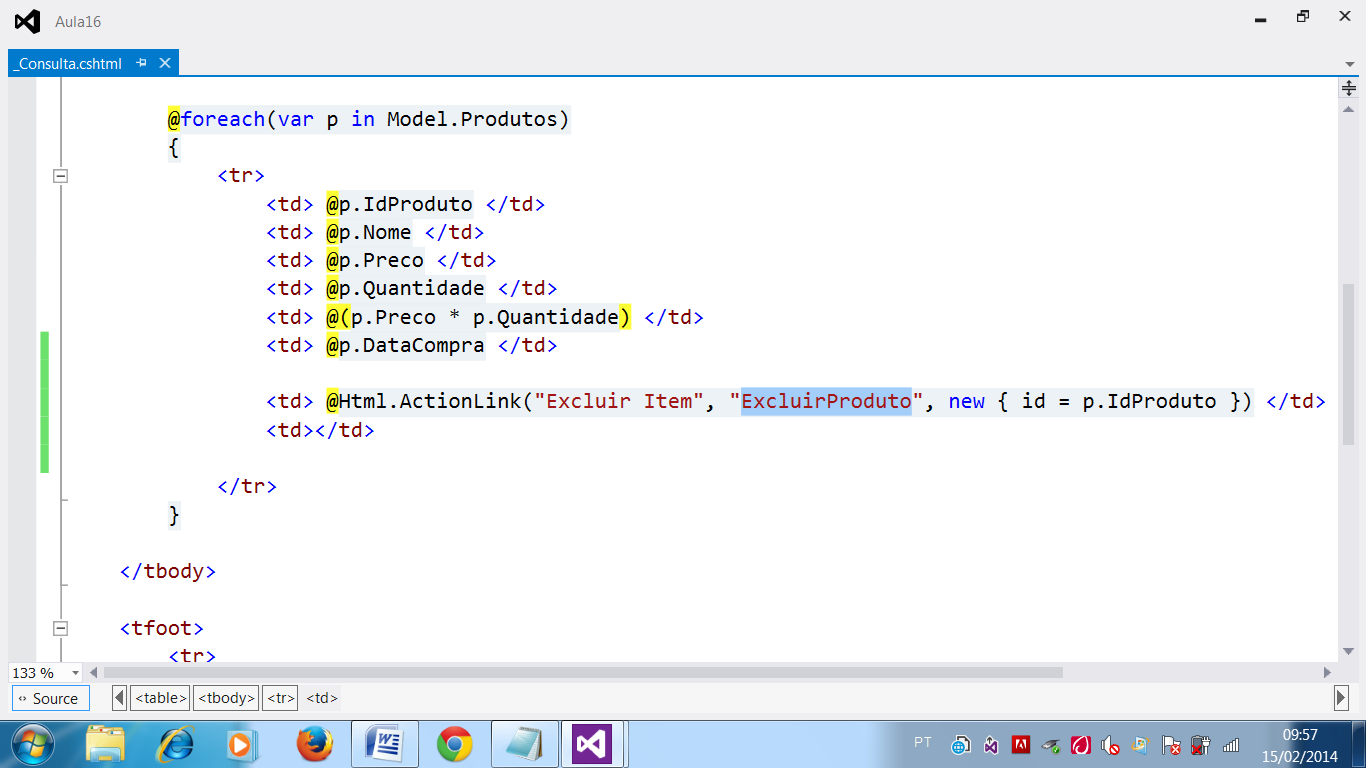
<!-- Incluir o conteúdo da página de Consulta -->

**@Html.Partial("\_Consulta", ViewData["dados"])**

}



Implementando a Exclusão



@model Site.Models.ProdutoConsultaModel

<p>

Relação de Produtos cadastrados

</p>

<table border="1" style="width: 100%;">

<thead>

<tr>

<th>Código</th>

<th>Nome do Produto</th>

<th>Preço</th>

<th>Quantidade</th>

<th>Total</th>

<th>Data de Compra</th>

<th>Excluir</th>

<th>Editar</th>

</tr>

</thead>

<tbody>

@foreach(var p in Model.Produtos)

{

<tr>

<td> @p.IdProduto </td>

<td> @p.Nome </td>

<td> @p.Preco </td>

<td> @p.Quantidade </td>

<td> @(p.Preco \* p.Quantidade) </td>

<td> @p.DataCompra </td>

**<td> @Html.ActionLink("Excluir Item", "ExcluirProduto",**

**new { id = p.IdProduto }) </td>**

<td></td>

</tr>

}

</tbody>

<tfoot>

<tr>

<td colspan="8">Quantidade de Produtos: @Model.QtdProdutos </td>

</tr>

</tfoot>

</table>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DAL.Entities;

namespace DAL.Persistence

{

public class ProdutoDal : IDisposable

{

private Conexao Con; //atributo null

//Construtor [ctor] + 2x[tab]

public ProdutoDal()

{

Con = new Conexao(); //inicializando

}

public void Salvar(Produto p)

{

try

{

Con.Produto.Add(p); //insert

Con.SaveChanges();

}

catch(Exception e)

{

throw new Exception("Erro ao salvar produto: " + e.Message);

}

}

public List<Produto> ListarTodos()

{

try

{

//SQL => select \* from Produto order by DataCompra desc

return Con.Produto.OrderByDescending(p => p.DataCompra).ToList();

}

catch(Exception e)

{

throw new Exception("Erro ao listar produtos: " + e.Message);

}

}

public int ObterQuantidadeRegistros()

{

try

{

//SQL => select count(\*) from Produto

return Con.Produto.Count();

}

catch(Exception e)

{

throw new Exception("Erro ao obter quantidade de registros: "

+ e.Message);

}

}

public Produto ObterPorId(int IdProduto)

{

try

{

//SQL -> select \* from Produto where IdProduto = @v1

return Con.Produto.Find(IdProduto);

//Find -> obter registro pelo id

}

catch(Exception e)

{

throw new Exception("Erro ao obter produto: " + e.Message);

}

}

public void Excluir(Produto p)

{

try

{

//SQL -> delete from Produto where IdProduto = @v1

Con.Produto.Remove(p); //excluir

Con.SaveChanges(); //executar

}

catch(Exception e)

{

throw new Exception("Erro ao excluir produto: " + e.Message);

}

}

public void Dispose()

{

Con.Dispose(); //destruir a conexão

}

}

}

Implementando o Controller

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using DAL.Entities;

using DAL.Persistence;

using Site.Models;

namespace Site.Controllers

{

public class ProdutoController : Controller

{

//Página ser aberta! => /Produto/Cadastro

public ActionResult Cadastro() //Page\_Load

{

ViewData["dados"] = ObterDadosConsulta();

return View();

}

public ActionResult Pesquisa() //Page\_Load

{

return View();

}

//Método para cadastrar produto obtido pelo formulário

[HttpPost] //método chamado a partir de um formulário FormMethod.Post

public ActionResult CadastrarProduto(ProdutoCadastroModel Model)

{

try

{

Produto p = new Produto();

p.Nome = Model.Nome;

p.Preco = Model.Preco;

p.Quantidade = Model.Quantidade;

p.DataCompra = Model.DataCompra;

using(ProdutoDal d = new ProdutoDal())

{

d.Salvar(p);

ViewBag.Mensagem = "Produto cadastrado com sucesso.";

ModelState.Clear(); //limpar o formulário

}

}

catch(Exception e)

{

ViewBag.Mensagem = e.Message;

}

ViewData["dados"] = ObterDadosConsulta();

return View("Cadastro"); //redirecionamento

}

//Método para produzir conteudo na Classe ProdutoConsultaModel

private ProdutoConsultaModel ObterDadosConsulta()

{

try

{

using(ProdutoDal d = new ProdutoDal())

{

ProdutoConsultaModel Model = new ProdutoConsultaModel();

Model.Produtos = d.ListarTodos();

Model.QtdProdutos = d.ObterQuantidadeRegistros();

return Model;

}

}

catch(Exception e)

{

ViewBag.Mensagem = e.Message;

return null; //retorna vazio

}

}

[HttpGet] //POST (Submit de formulário) GET (URL)

public ActionResult ExcluirProduto(int id)

{

try

{

using(ProdutoDal d = new ProdutoDal())

{

//Obter o produto pelo id

Produto p = d.ObterPorId(id);

if(p != null) //Produto encontrado!

{

d.Excluir(p); //excluir o produto

ViewBag.Resultado = "Produto " + p.Nome

+ " excluído com sucesso.";

}

else

{

throw new Exception("Produto não encontrado.");

}

}

}

catch(Exception e)

{

ViewBag.Resultado = e.Message;

}

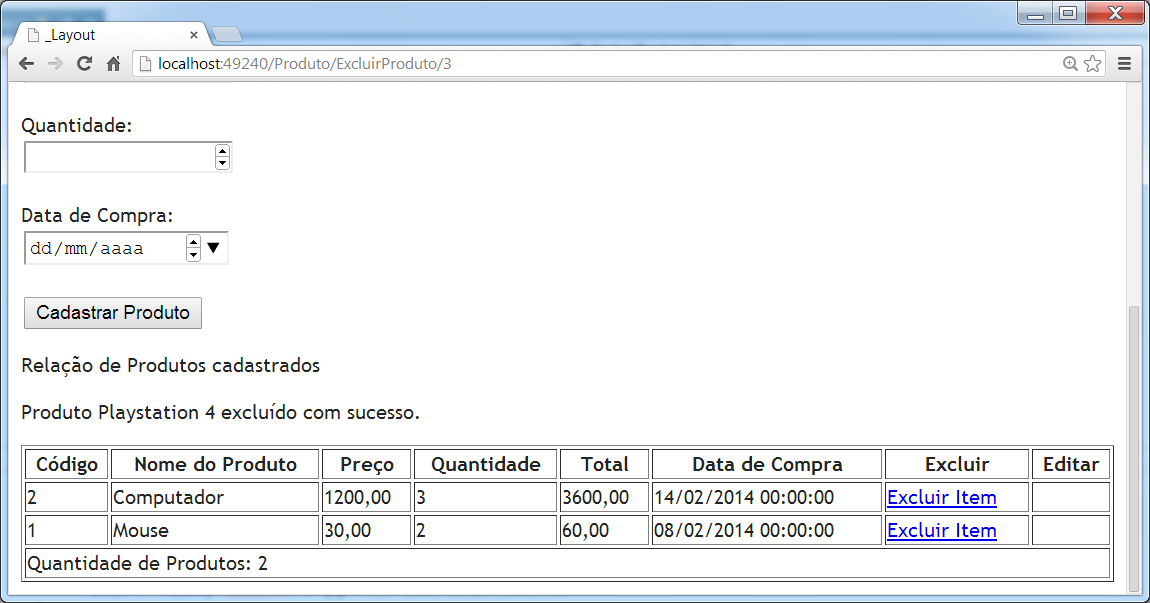
ViewData["dados"] = ObterDadosConsulta();

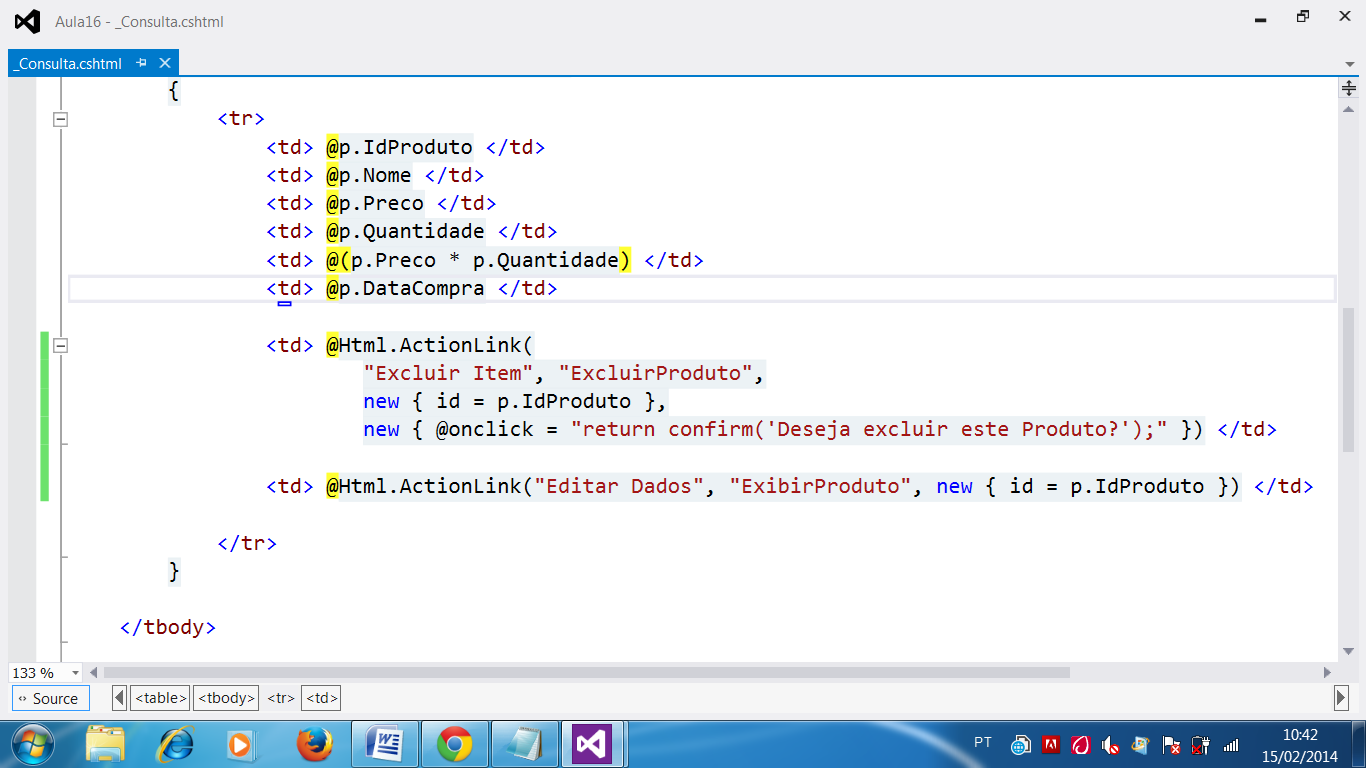
return View("Cadastro"); //redirecionamento

}

}

}





@model Site.Models.ProdutoConsultaModel

<p>

Relação de Produtos cadastrados

</p>

<p>

@ViewBag.Resultado

</p>

<table border="1" style="width: 100%;">

<thead>

<tr>

<th>Código</th>

<th>Nome do Produto</th>

<th>Preço</th>

<th>Quantidade</th>

<th>Total</th>

<th>Data de Compra</th>

<th>Excluir</th>

<th>Editar</th>

</tr>

</thead>

<tbody>

@foreach(var p in Model.Produtos)

{

<tr>

<td> @p.IdProduto </td>

<td> @p.Nome </td>

<td> @p.Preco </td>

<td> @p.Quantidade </td>

<td> @(p.Preco \* p.Quantidade) </td>

<td> @p.DataCompra </td>

<td> @Html.ActionLink(

"Excluir Item", "ExcluirProduto",

new { id = p.IdProduto },

new { @onclick = "return confirm('Deseja excluir

este Produto?');" }) </td>

<td> @Html.ActionLink("Editar Dados", "ExibirProduto",

new { id = p.IdProduto }) </td>

</tr>

}

</tbody>

<tfoot>

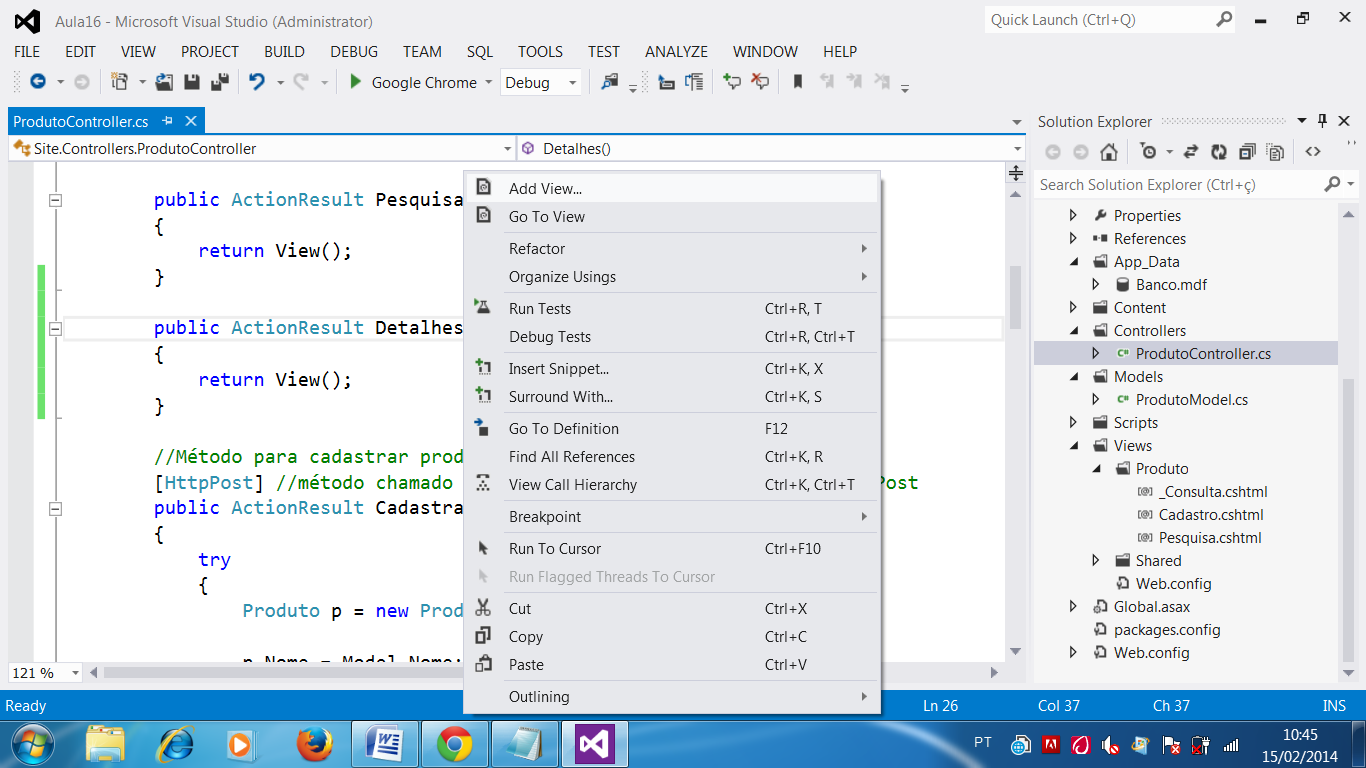
<tr>

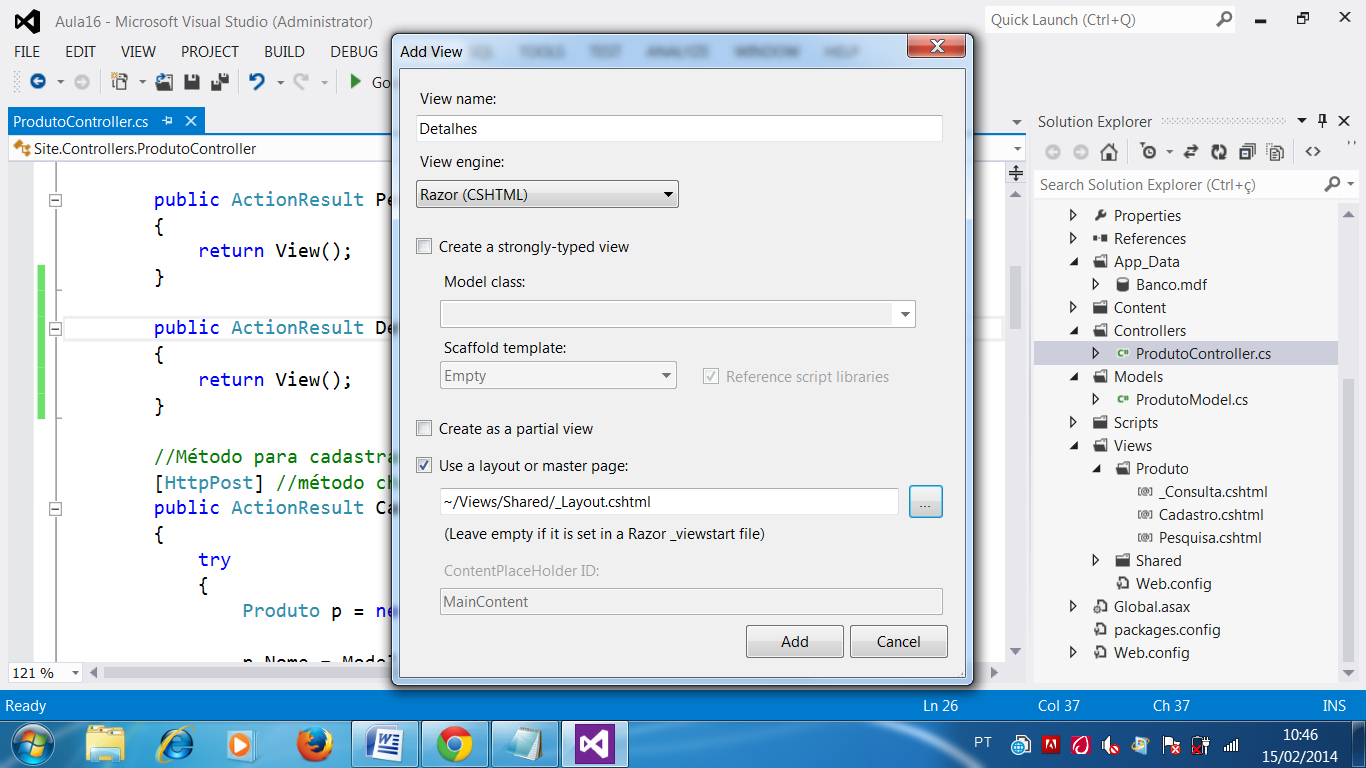
<td colspan="8">Quantidade de Produtos: @Model.QtdProdutos </td>

</tr>

</tfoot>

</table>





Criando a Classe de Modelo para a página de Edição

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

using DAL.Entities;

namespace Site.Models

{

public class ProdutoCadastroModel

{

[Display(Name = "Nome do Produto:")]

public string Nome { get; set; }

[Display(Name = "Preço:")]

public decimal Preco { get; set; }

[Display(Name = "Quantidade:")]

public int Quantidade { get; set; }

[Display(Name = "Data de Compra:")]

public DateTime DataCompra { get; set; }

}

public class ProdutoConsultaModel

{

//Listagem dos Produtos que serão obtidos do banco

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

public int QtdProdutos { get; set; }

}

public class ProdutoDetalhesModel

{

[Display(Name = "Código")]

public int IdProduto { get; set; }

[Display(Name = "Nome do Produto:")]

public string Nome { get; set; }

[Display(Name = "Preço:")]

public decimal Preco { get; set; }

[Display(Name = "Quantidade:")]

public int Quantidade { get; set; }

}

}

@model Site.Models.ProdutoDetalhesModel

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Detalhes do Produto</h2>

<!-- Criando o formulário -->

@using(Html.BeginForm("AtualizarProduto", "Produto", FormMethod.Post))

{

@Html.LabelFor(p => p.IdProduto) <br />

@Html.TextBoxFor(p => p.IdProduto)

<br /><br />

@Html.LabelFor(p => p.Nome) <br />

@Html.TextBoxFor(p => p.Nome)

<br /><br />

@Html.LabelFor(p => p.Preco) <br />

@Html.TextBoxFor(p => p.Preco)

<br /><br />

@Html.LabelFor(p => p.Quantidade) <br />

@Html.TextBoxFor(p => p.Quantidade)

<br /><br />

<input type="submit" value="Atualizar Produto" />

<p>

@ViewBag.Mensagem

</p>

}

[HttpGet] //recebe dados por URL

public ActionResult ExibirProduto(int id)

{

ProdutoDetalhesModel Model = new ProdutoDetalhesModel();

//classe de modelo

try

{

using(ProdutoDal d = new ProdutoDal())

{

Produto p = d.ObterPorId(id);

if(p != null) //Produto foi encontrado

{

Model.IdProduto = p.IdProduto;

Model.Nome = p.Nome;

Model.Preco = p.Preco;

Model.Quantidade = p.Quantidade;

}

else

{

throw new Exception("Produto nao encontrado.");

}

}

}

catch(Exception e)

{

ViewBag.Mensagem = e.Message;

}

//redirecionamento

return View("Detalhes", Model);

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DAL.Entities;

namespace DAL.Persistence

{

public class ProdutoDal : IDisposable

{

private Conexao Con; //atributo null

//Construtor [ctor] + 2x[tab]

public ProdutoDal()

{

Con = new Conexao(); //inicializando

}

public void Salvar(Produto p)

{

try

{

Con.Produto.Add(p); //insert

Con.SaveChanges();

}

catch(Exception e)

{

throw new Exception("Erro ao salvar produto: " + e.Message);

}

}

public List<Produto> ListarTodos()

{

try

{

//SQL => select \* from Produto order by DataCompra desc

return Con.Produto.OrderByDescending(p => p.DataCompra).ToList();

}

catch(Exception e)

{

throw new Exception("Erro ao listar produtos: " + e.Message);

}

}

public int ObterQuantidadeRegistros()

{

try

{

//SQL => select count(\*) from Produto

return Con.Produto.Count();

}

catch(Exception e)

{

throw new Exception("Erro ao obter quantidade de

registros: " + e.Message);

}

}

public Produto ObterPorId(int IdProduto)

{

try

{

//SQL -> select \* from Produto where IdProduto = @v1

return Con.Produto.Find(IdProduto);

//Find -> obter registro pelo id

}

catch(Exception e)

{

throw new Exception("Erro ao obter produto: " + e.Message);

}

}

public void Excluir(Produto p)

{

try

{

//SQL -> delete from Produto where IdProduto = @v1

Con.Produto.Remove(p); //excluir

Con.SaveChanges(); //executar

}

catch(Exception e)

{

throw new Exception("Erro ao excluir produto: " + e.Message);

}

}

public void Atualizar(Produto p)

{

try

{

//Buscar o registro do Produto original da base de dados, pelo id

Produto registro = ObterPorId(p.IdProduto);

registro.Nome = p.Nome; //alterar o nome

registro.Preco = p.Preco; //alterar o preço

registro.Quantidade = p.Quantidade; //alterar a quantidade

Con.SaveChanges(); //executa a alteração

}

catch(Exception e)

{

throw new Exception("Erro ao atualizar produto: " + e.Message);

}

}

public void Dispose()

{

Con.Dispose(); //destruir a conexão

}

}

}

Implementando no Controle...

[HttpPost]

public ActionResult AtualizarProduto(ProdutoDetalhesModel Model)

{

try

{

Produto p = new Produto();

p.IdProduto = Model.IdProduto;

p.Nome = Model.Nome;

p.Preco = Model.Preco;

p.Quantidade = Model.Quantidade;

using(ProdutoDal d = new ProdutoDal())

{

d.Atualizar(p);

ViewBag.Mensagem = "Produto " + p.Nome

+ ", atualizado com sucesso.";

//ModelState.Clear(); //limpa os dados do formulário

}

}

catch(Exception e)

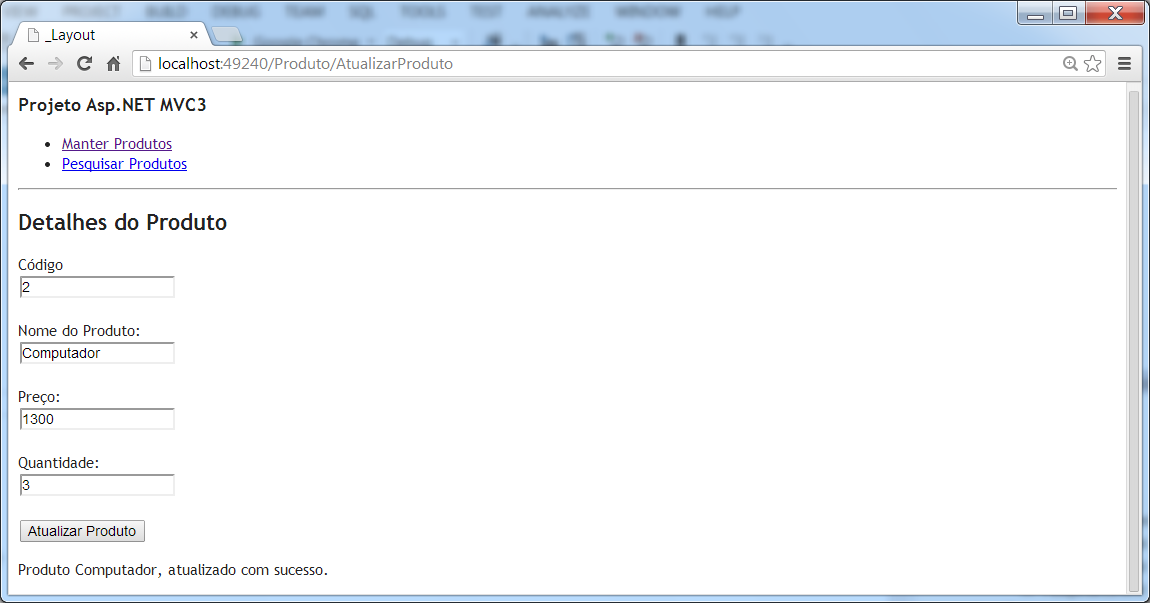
{

ViewBag.Mensagem = e.Message;

}

return View("Detalhes");

}



Implementando a Pesquisa de Produtos...

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

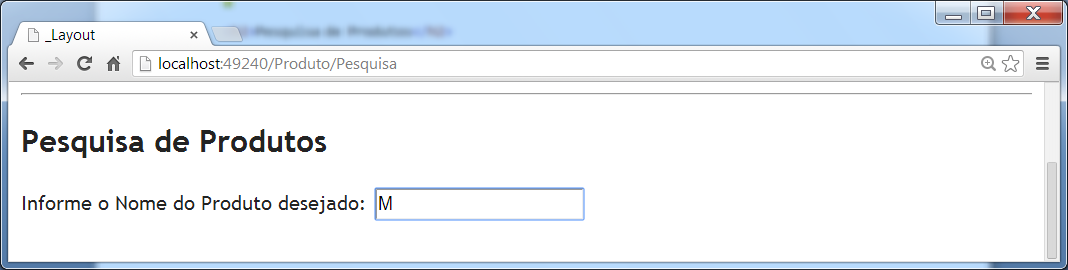
<h2>Pesquisa de Produtos</h2>

Informe o Nome do Produto desejado:

<input type="text" id="nome" />

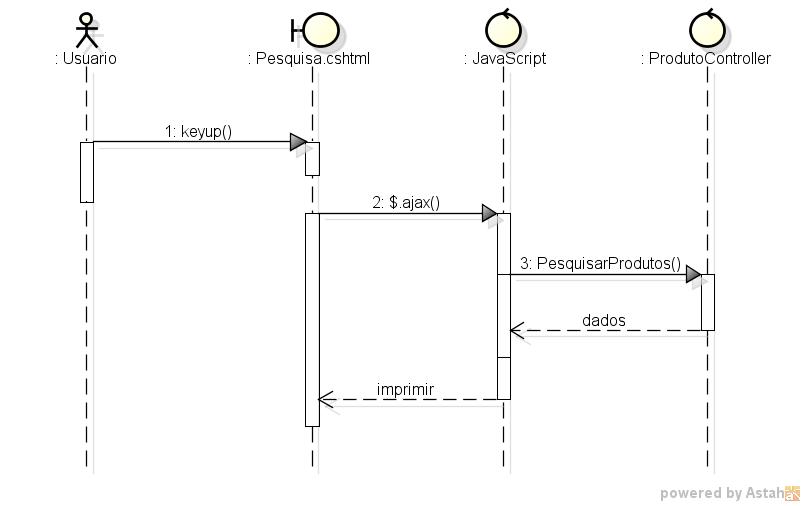
<br /><br />

<span id="resultado"></span>



Ajax

Assyncronous JavaScript and XML



<script src="../../Scripts/jquery-1.7.1.js"></script>

<script type="text/javascript">

$(document).ready( //quando a página for carregada

function () { //faça...

//criar um evento keypress no campo nome

$("#nome").keyup(

function () {

//capturar o valor digitado no campo

var nome = $(this).val();

$.ajax(

{

type : "POST",

url: "/Produto/PesquisarProdutos",

data: "nome=" + nome,

success: function (dados) {

//dados -> enviado pelo servidor

$("#resultado").html(dados);

},

error: function (e) { //e -> exception

$("#resultado").html(e.responseText);

}

}

);

}

);

}

);

</script>

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

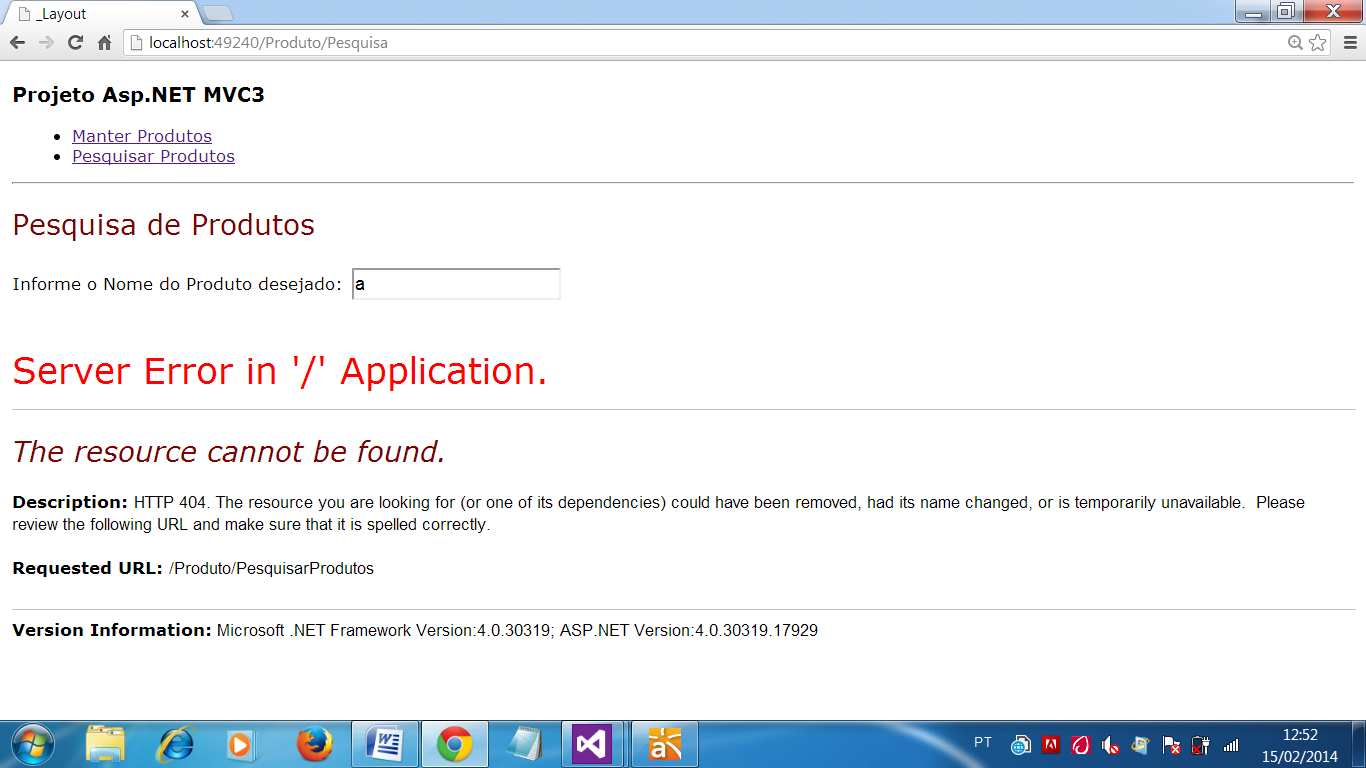
<h2>Pesquisa de Produtos</h2>

Informe o Nome do Produto desejado:

<input type="text" id="nome" />

<br /><br />

<span id="resultado"></span>





//JsonResult (Json -> JavaScript Object Notation)

public JsonResult PesquisarProdutos(string nome)

{

try

{

using(ProdutoDal d = new ProdutoDal())

{

string dados = string.Empty;

foreach(Produto p in d.ListarPorNome(nome))

{

dados += string.Format("{0}, Preço: {1:c} <br/>",

p.Nome, p.Preco);

}

return Json(dados); //enviar o resultado para a página

}

}

catch(Exception e)

{

return Json(e.Message);

}

}

