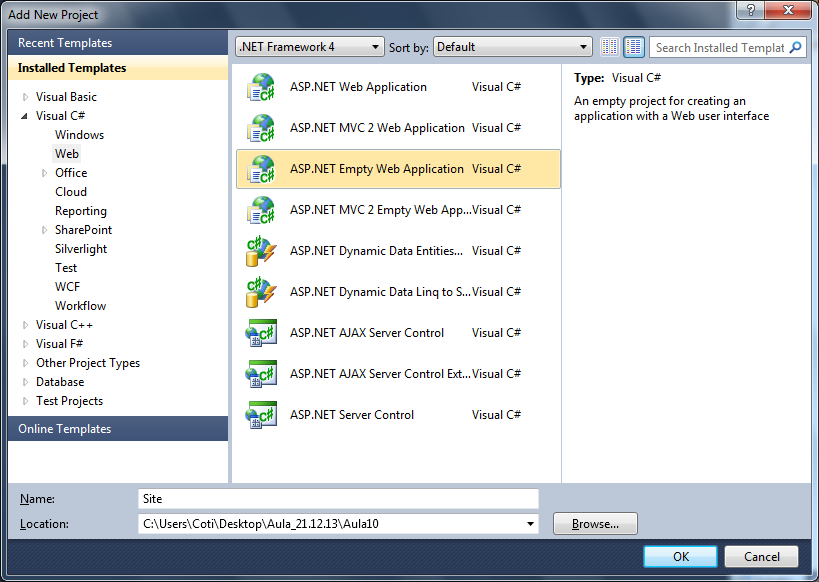
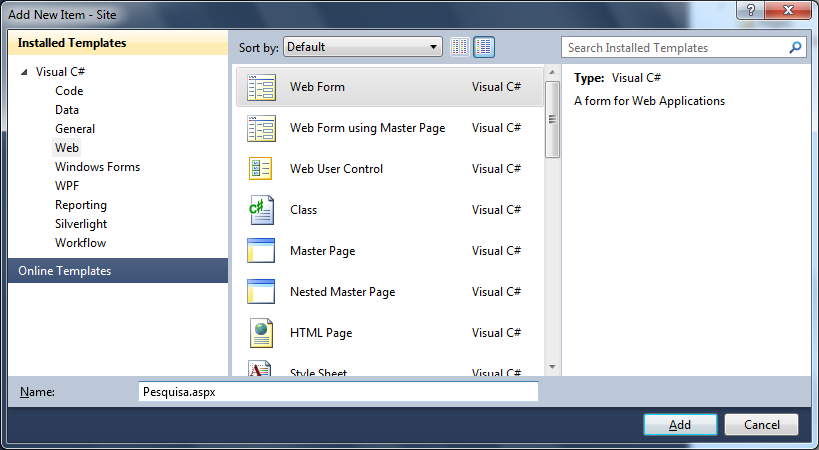


Criando o Projeto Asp.NET



/Pages/Pesquisa.aspx



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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<head runat="server">

<title></title>

<style type="text/css">

body { font-family: Verdana; font-size: 9pt; padding: 30px; }

</style>

</head>

<body>

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

<div>

<h3>Pesquisa de Vendas</h3>

Data de Início: <asp:TextBox ID="txtDataInicio" runat="server" />

Término: <asp:TextBox ID="txtDataTermino" runat="server" />

<asp:Button ID="btnPesquisa" runat="server"

Text="Gerar Relatório" />

<p>

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

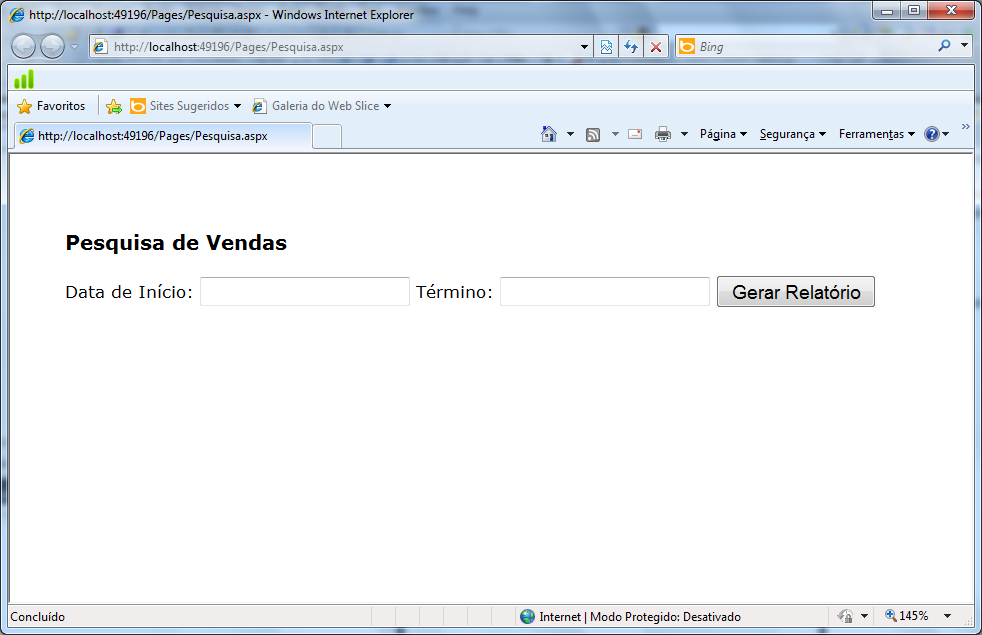
</p>

</div>

</form>

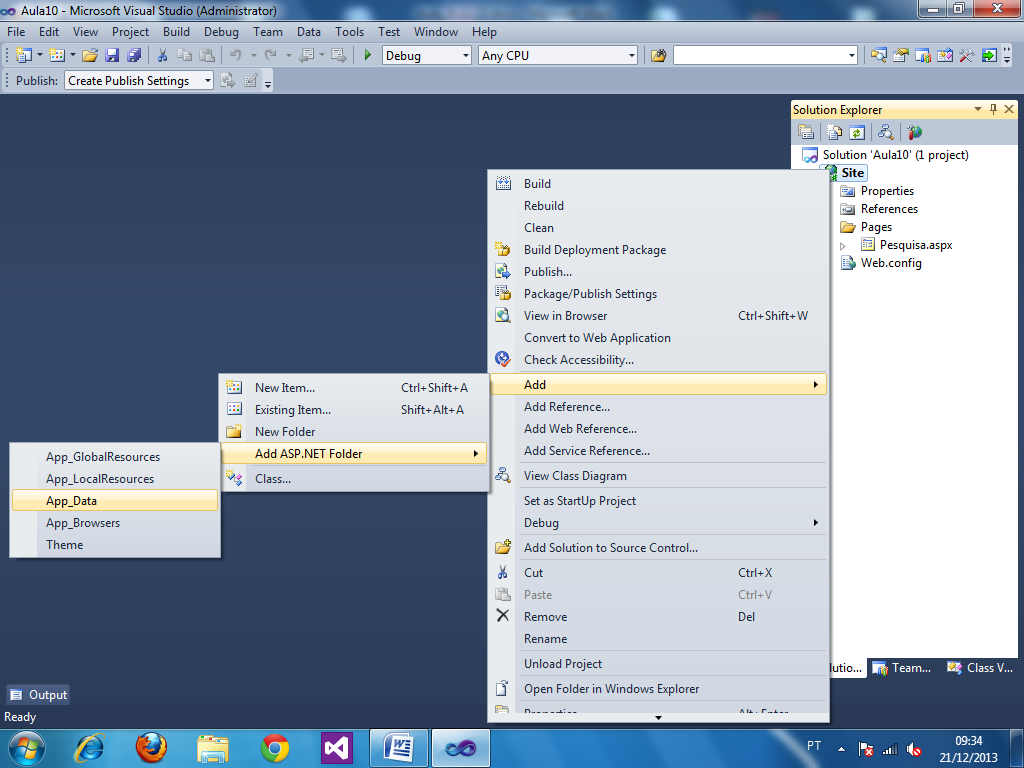
</body>

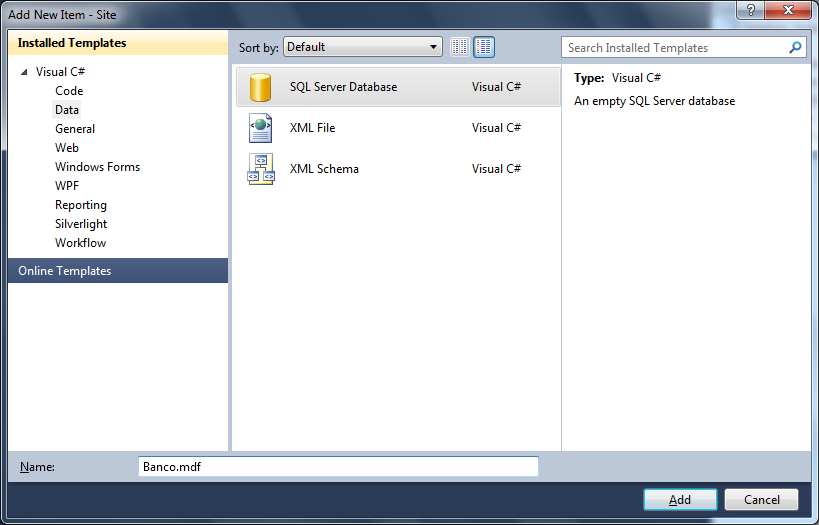
</html>



Criando a base de dados…

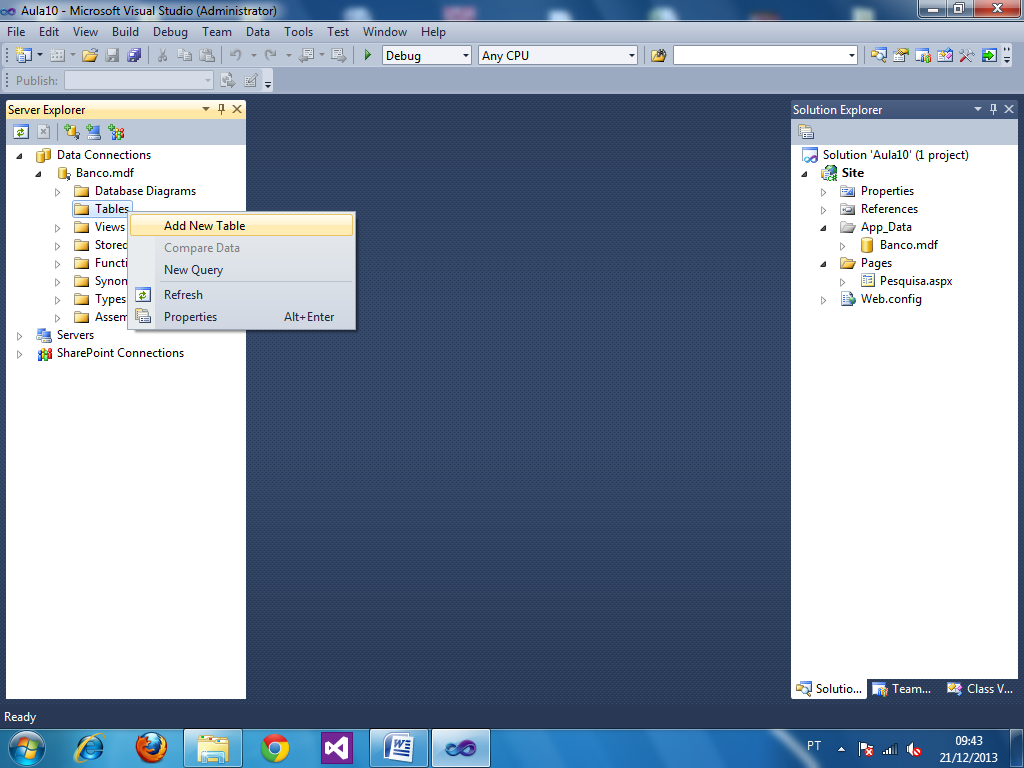
Formato **.mdf**

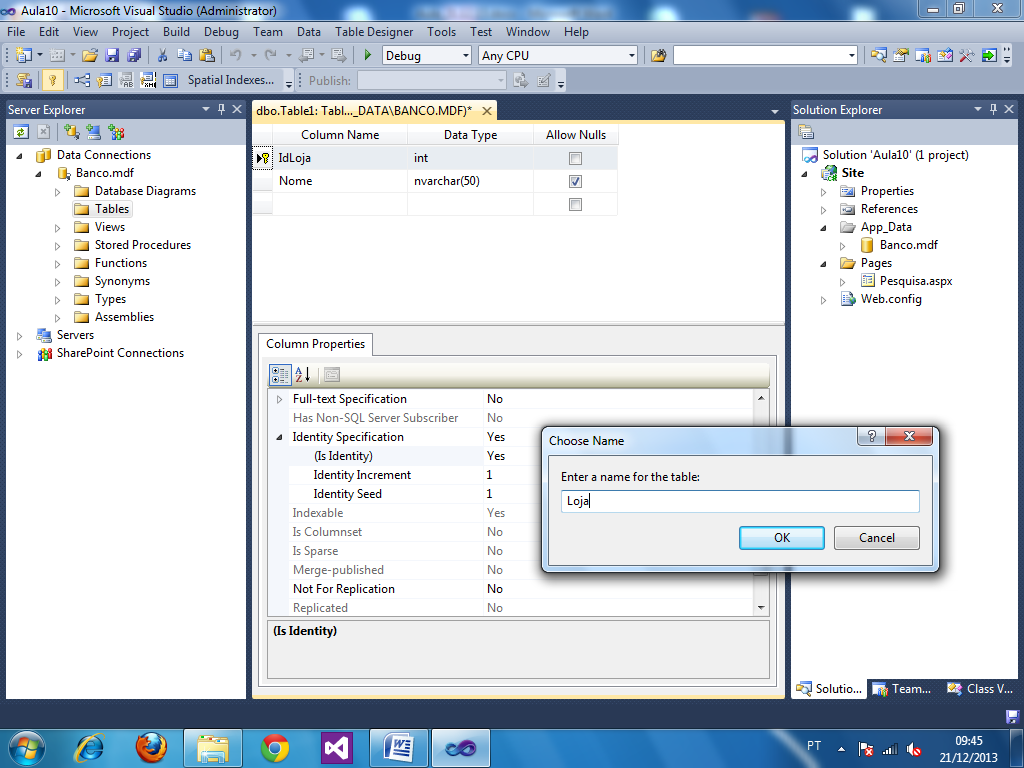


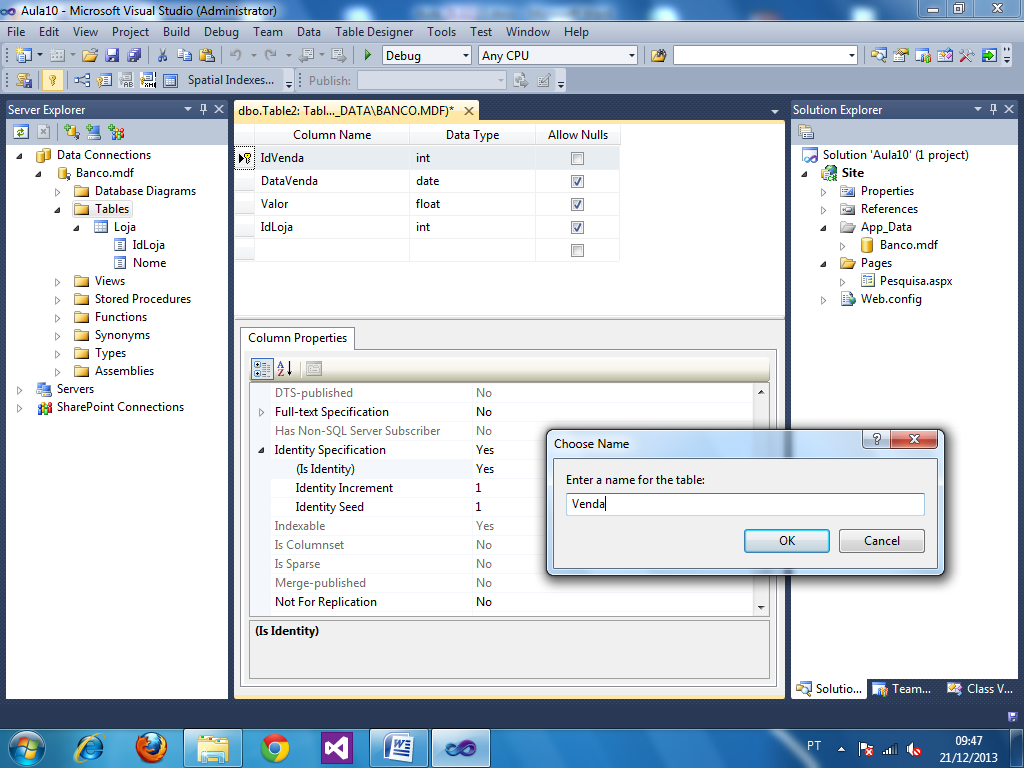


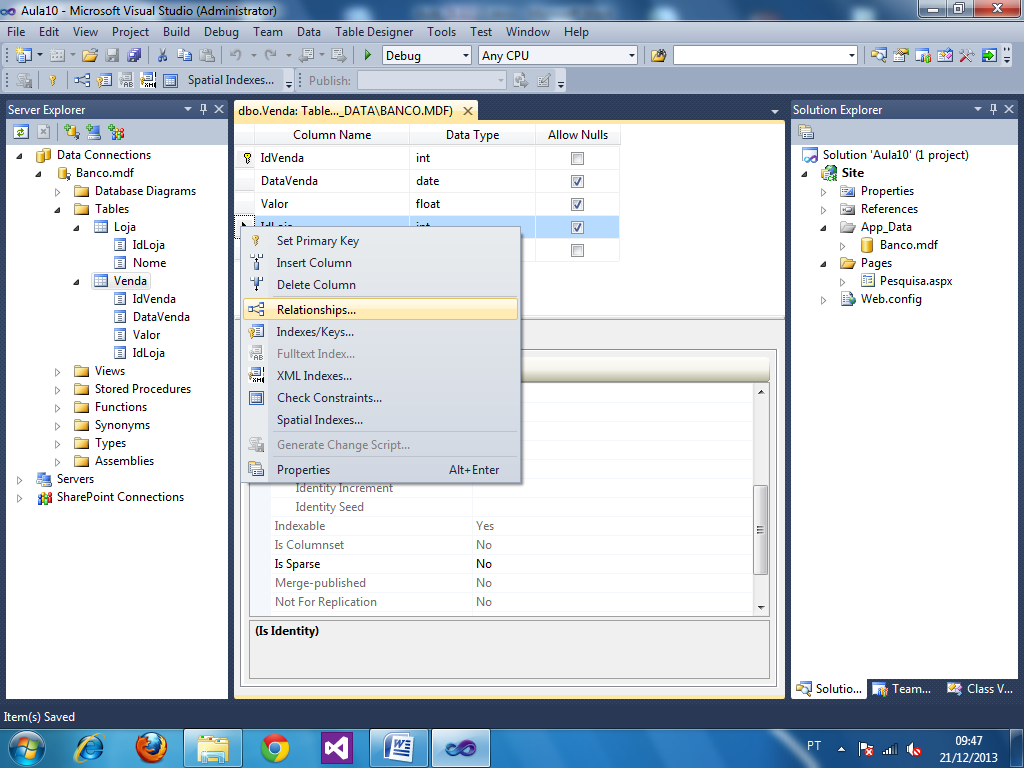
Modelagem do banco de dados

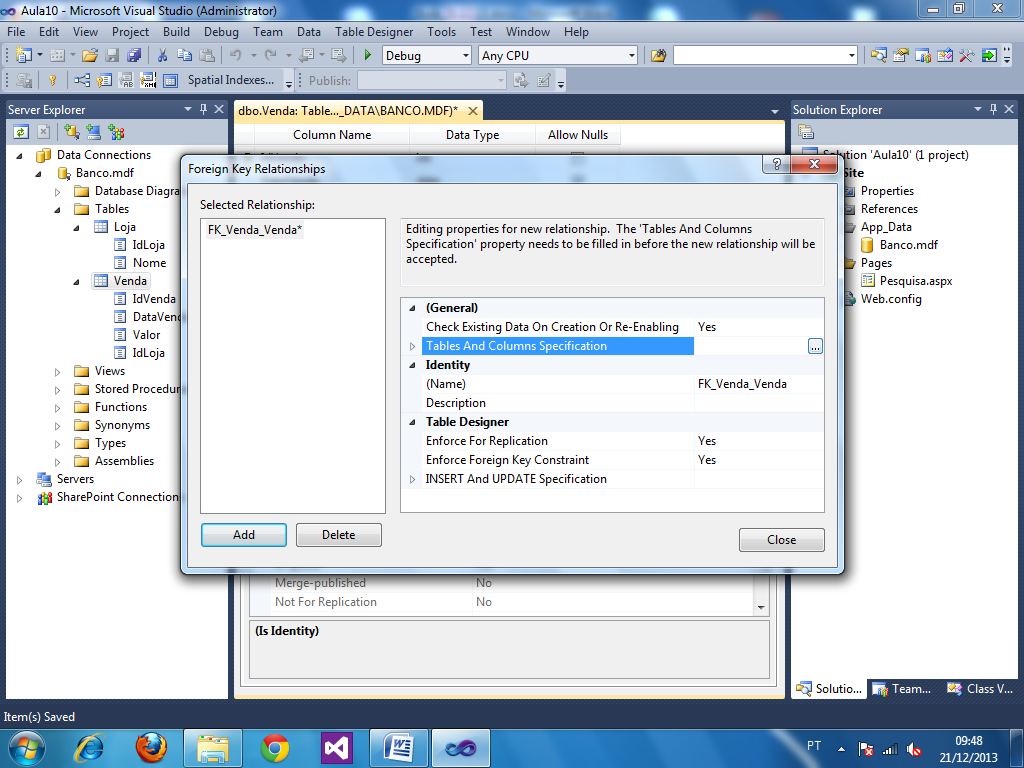
* Loja
* Venda

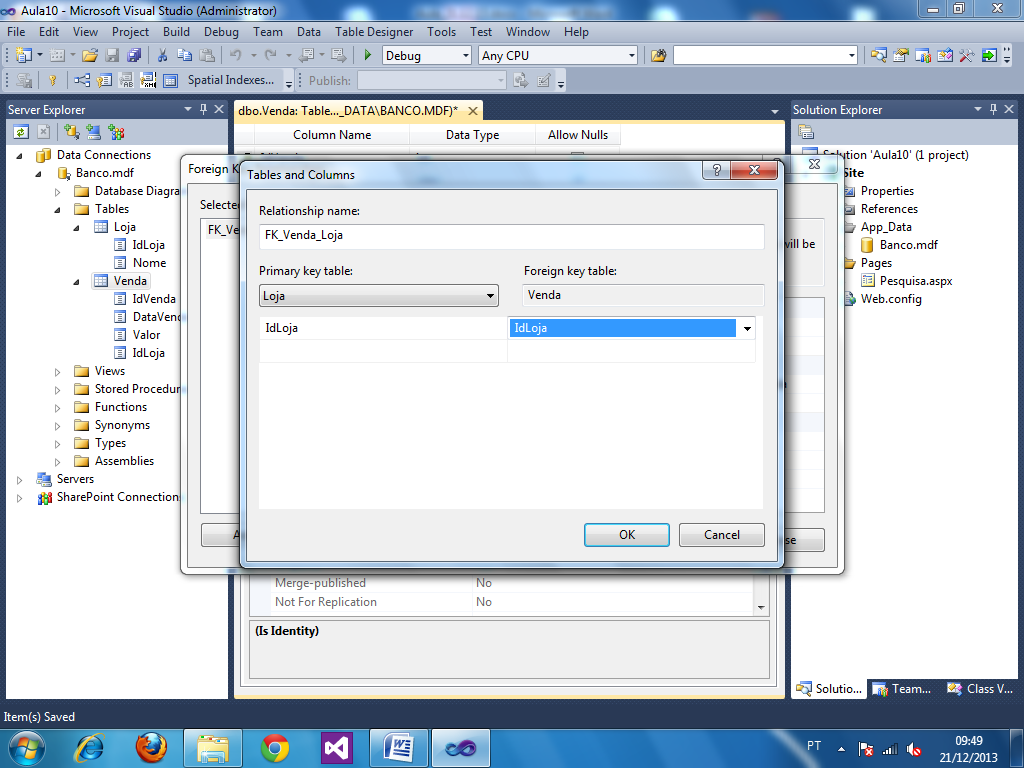












Mapeamento da ConnectionString

<?xml version="1.0"?>

<configuration>

**<!-- Mapeamento da ConnectionString -->**

**<connectionStrings>**

**<add**

**name="AULA"**

**connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\Coti\Desktop\Aula\_21.12.13\Aula10\Site\App\_Data\Banco.mdf;Integrated Security=True;User Instance=True"**

**/>**

**</connectionStrings>**

<system.web>

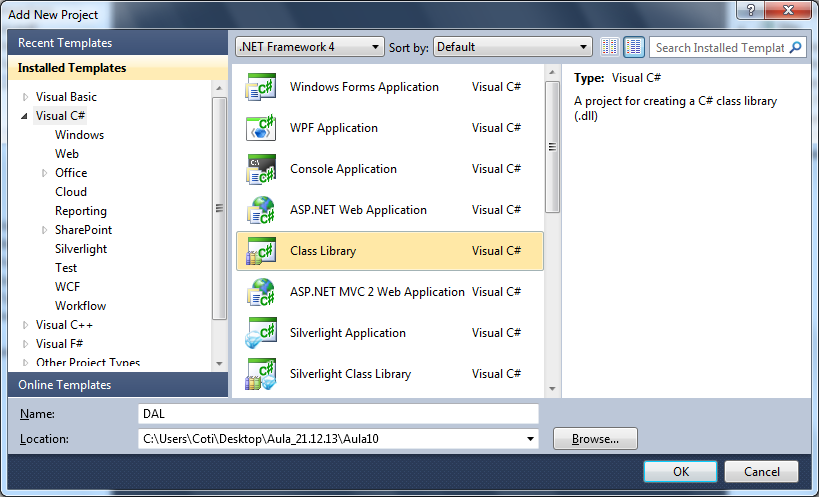
<compilation debug="true" targetFramework="4.0" />

</system.web>

</configuration>

Data Access Layer

Camada de acesso a dados



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace DAL.Model

{

public class Loja

{

public int IdLoja { get; set; }

public string Nome { get; set; }

//Associação -> N Vendas

public List<Venda> Vendas { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace DAL.Model

{

public class Venda

{

public int IdVenda { get; set; }

public DateTime DataVenda { get; set; }

public double Valor { get; set; }

//Associação -> 1 Loja

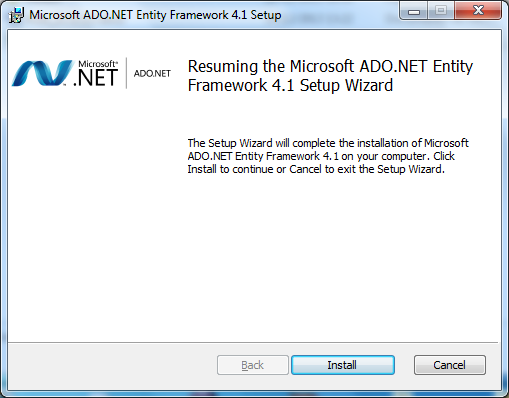
Public virtual Loja Loja { get; set; }

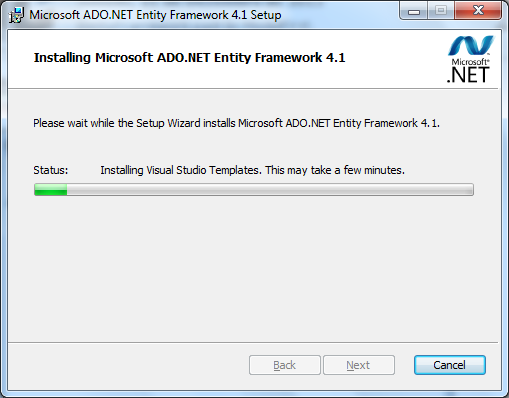
}

}

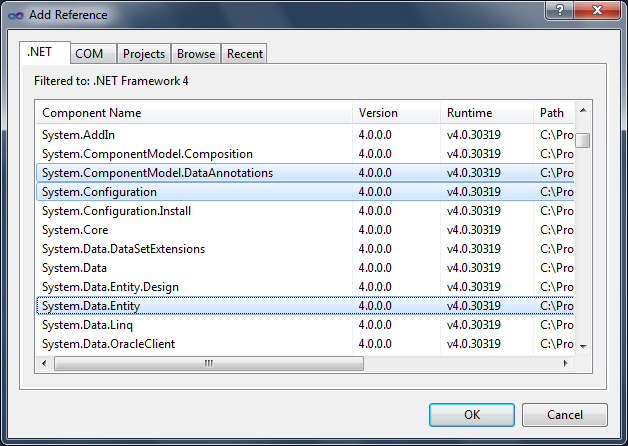
EntityFramework

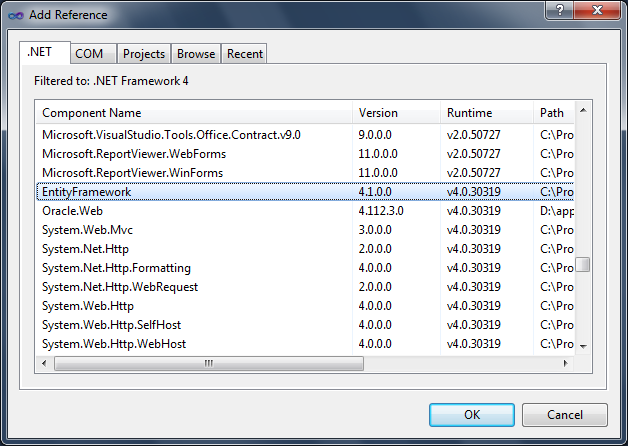
Instalando…





Adicionando referências no projeto…





Mapeamento Objeto relacional (ORM)

Anotações que definem as Classes de modelo como representações as entidades existentes na base de dados.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.ComponentModel.DataAnnotations; //ORM

namespace DAL.Model

{

[Table("Loja")]

public class Loja

{

[Key] //Chave primária

[DatabaseGenerated(DatabaseGeneratedOption.Identity)] //identity

[Column("IdLoja")]

public int IdLoja { get; set; }

[Column("Nome")]

public string Nome { get; set; }

//Associação -> N Vendas

public List<Venda> Vendas { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.ComponentModel.DataAnnotations; //ORM

namespace DAL.Model

{

[Table("Venda")]

public class Venda

{

[Key] //chave primária

[DatabaseGenerated(DatabaseGeneratedOption.Identity)] //identity

[Column("IdVenda")]

public int IdVenda { get; set; }

[Column("DataVenda")]

public DateTime DataVenda { get; set; }

[Column("Valor")]

public double Valor { get; set; }

//Propriedade para representar a chave estrangeira

//Só é necessária pois estamos trabalhando com EntityFramework

[Column("IdLoja")] //nome da coluna na tabela

public int IdLojaFK { get; set; }

//Associação -> 1 Loja

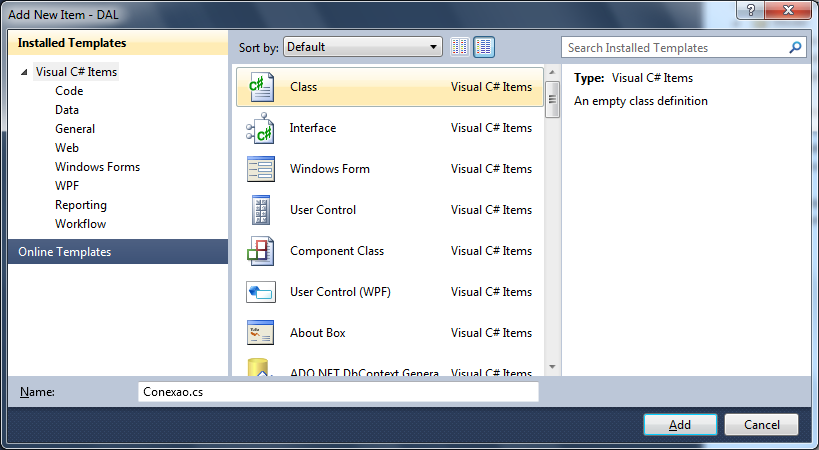
[ForeignKey("IdLojaFK")] //nome do atributo da Classe

public Loja Loja { get; set; }

}

}

Classe de conexão...



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Data.Entity; //DbContext e DbSet

using System.Configuration; //referência da ConnectionString

using DAL.Model; //Classes mapeadas

namespace DAL.Persistence

{

//Para que esta classe possa conectar-se a uma base de dados

//é necessário que façamos uma herança de System.Data.Entity.DbContext

public class Conexao : DbContext

{

//[ctor] + 2x[tab]

public Conexao() //Construtor da subclasse chamar o da SuperClasse

: base(ConfigurationManager.ConnectionStrings

["AULA"].ConnectionString)

{

//O construtor da Classe Conexão inicializa o Construtor da Classe

//DbContext enviado o caminho da connectionstring do \Web.config.xml

}

//declarar atributos para cada Classe mapeada

//DbSet -> dá suporte às operações de CRUD (insert,

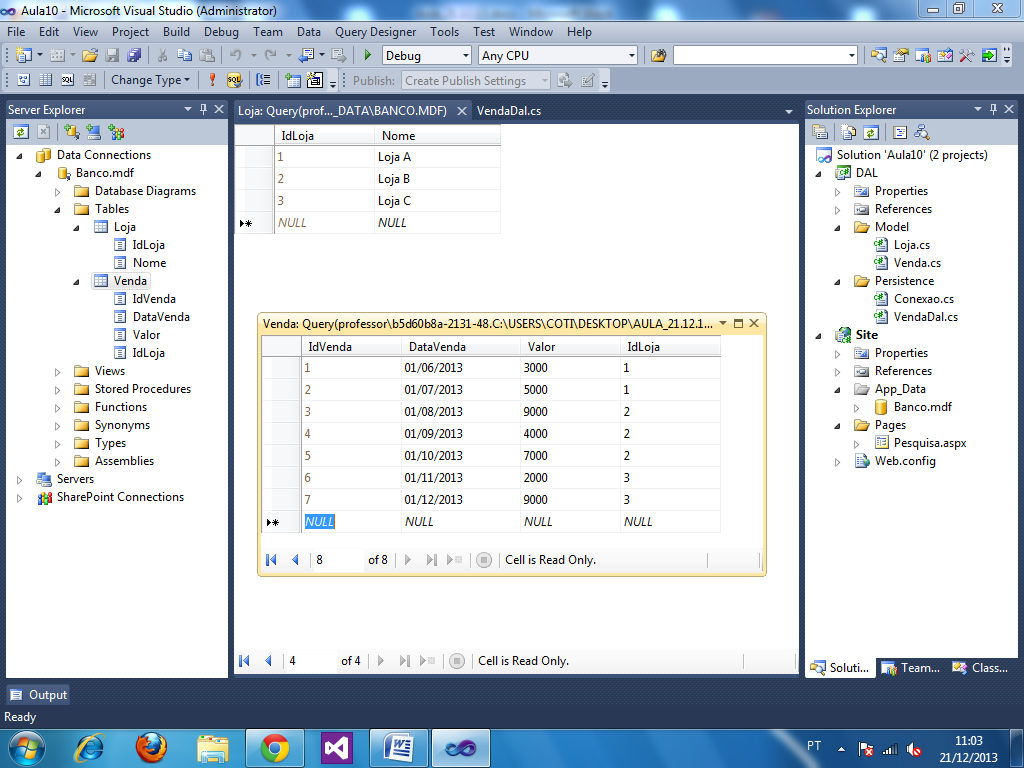
delete, update, etc...)

public DbSet<Loja> Loja { get; set; }

public DbSet<Venda> Venda { get; set; }

}

}

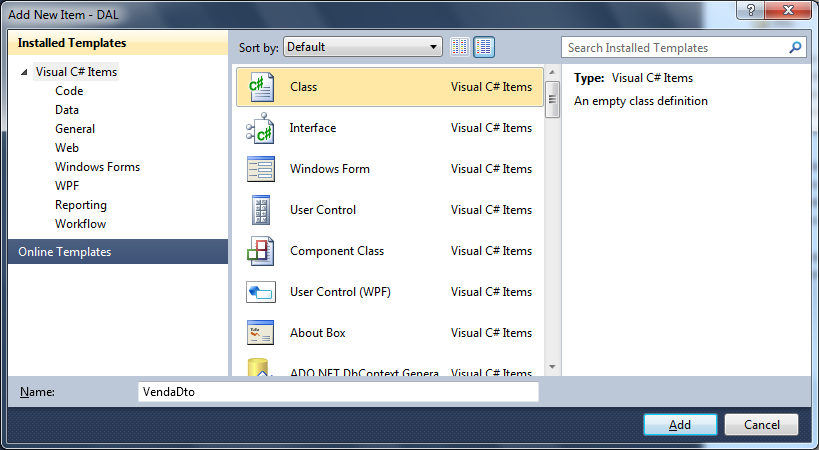


Modelagem do Relatório

* Codigo id da Venda inteiro
* Data data da venda date
* Valor valor double
* Loja nome da loja string

DTO – Data Transfer Object

Classe utilizada somente para transferência de dados entre aplicações, não comprometida com modelagem de banco de dados.



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace DAL.Dto

{

public class VendaDto

{

public int Codigo { get; set; }

public DateTime Data { get; set; }

public double Valor { get; set; }

public string Loja { get; set; }

}

}

Classe de persistência...

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using DAL.Model;

using DAL.Dto; //Data Transfer Object

namespace DAL.Persistence

{

public class VendaDal : IDisposable

{

//Atributo para a Classe de conexão

private Conexao Con; //null

//[ctor] + 2x[tab]

public VendaDal() //construtor default

{

//inicialização do atributo Conexao

Con = new Conexao(); //espaço de memória

}

//Método para listar todos as vendas por um intervalo de datas

public List<VendaDto> ListarVendas(DateTime DataInicio,

DateTime DataTermino)

{

try

{

//LINQ -> Language Integrated Query

//linguagem de consultas nativa do C# aplicada a SQL

var query = from v in Con.Venda

join l in Con.Loja

on v.IdLojaFK equals l.IdLoja

where v.DataVenda >= DataInicio

&& v.DataVenda <= DataTermino

select v;

List<VendaDto> lista = new List<VendaDto>(); //lista vazia

foreach(var registro in query.ToList()) //varrendo a query

{

VendaDto dto = new VendaDto()

{

Codigo = registro.IdVenda,

Data = registro.DataVenda,

Valor = registro.Valor,

Loja = registro.Loja.Nome

};

lista.Add(dto); //adiciono na lista

}

return lista; //retornar a lista

}

catch(Exception e)

{

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

}

}

//Método destrutor

public void Dispose()

{

Con.Dispose(); //liberar a conexão de memória

}

}

}

Em Sql...

select

IdVenda,

DataVenda,

Nome

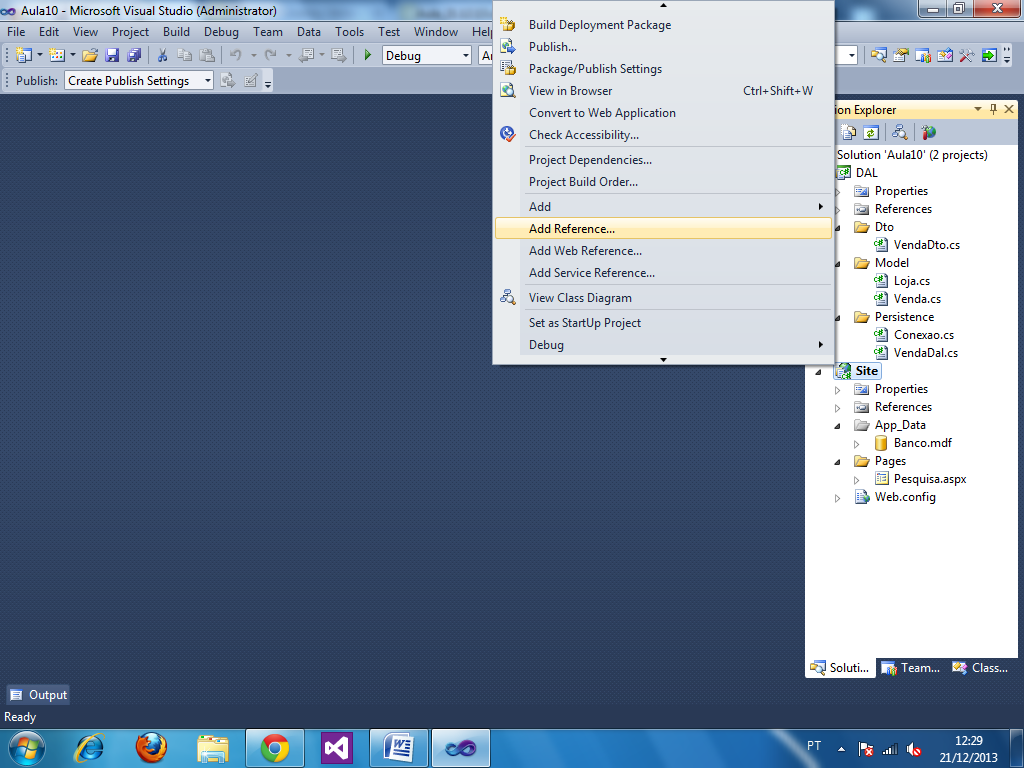
from Venda as v

inner join Loja as l

on v.IdLoja = l.IdLoja

where

DataVenda between @v1 and @v2





Criando o evento...

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<head runat="server">

<title></title>

<style type="text/css">

body { font-family: Verdana; font-size: 9pt; padding: 30px; }

</style>

</head>

<body>

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

<div>

<h3>Pesquisa de Vendas</h3>

Data de Início: <asp:TextBox ID="txtDataInicio" runat="server" />

Término: <asp:TextBox ID="txtDataTermino" runat="server" />

<asp:Button ID="btnPesquisa" runat="server"

Text="Gerar Relatório"

**OnClick="GerarRelatorio"** />

<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;

namespace Site.Pages

{

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

{

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

{

}

**protected void GerarRelatorio(object sender, EventArgs e)**

{

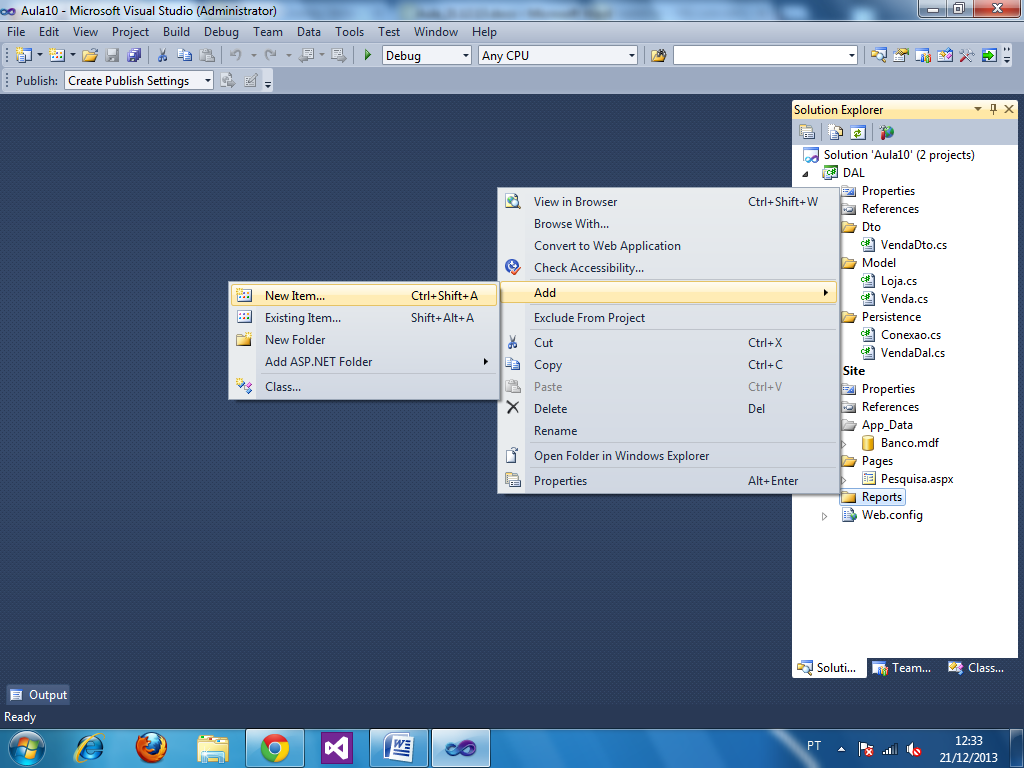
}

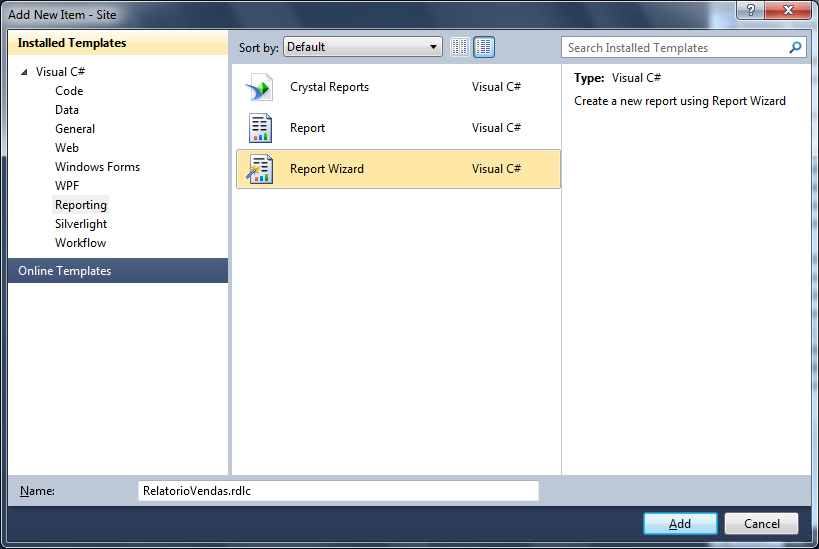
}

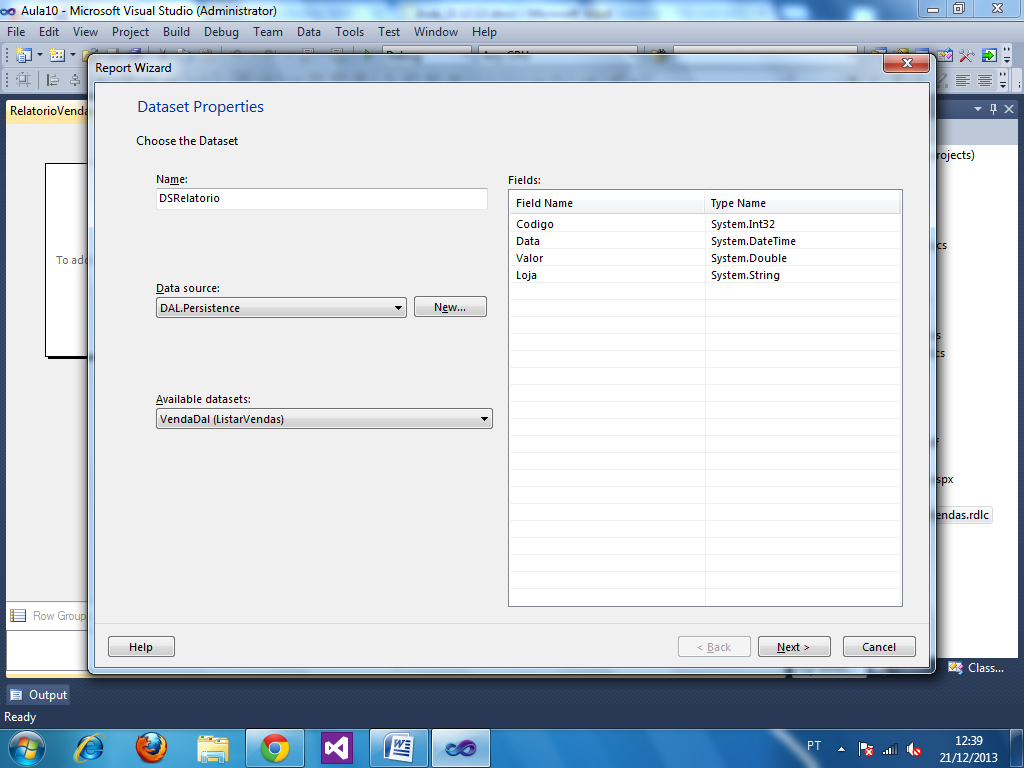
}

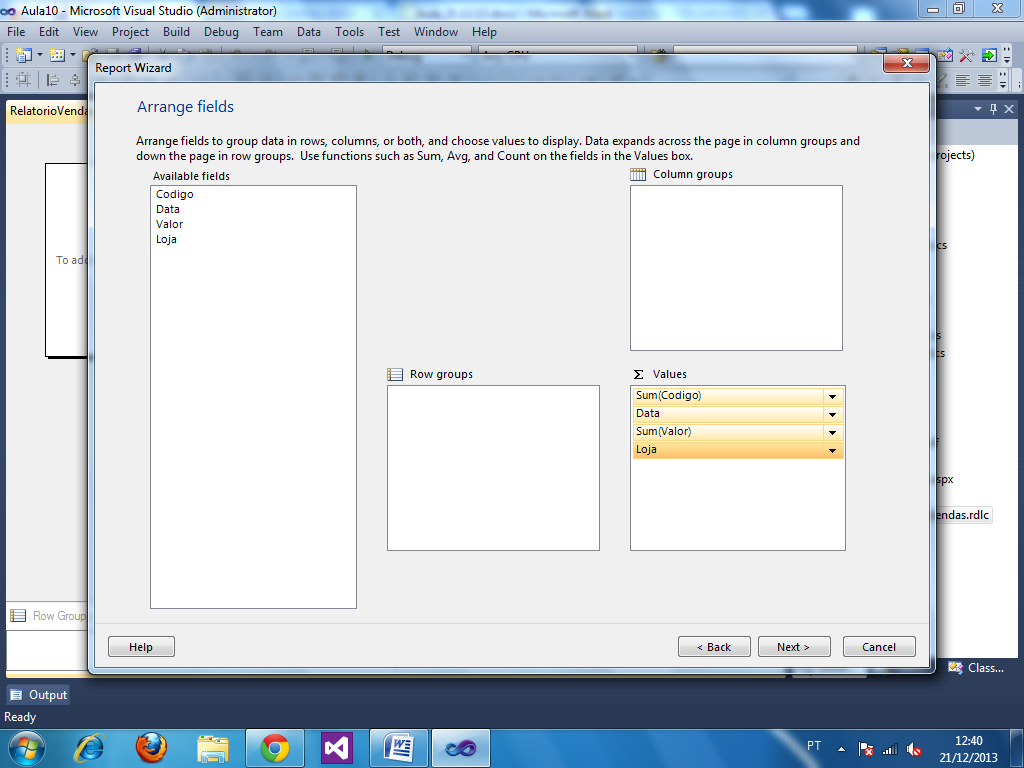
Criando o relatório

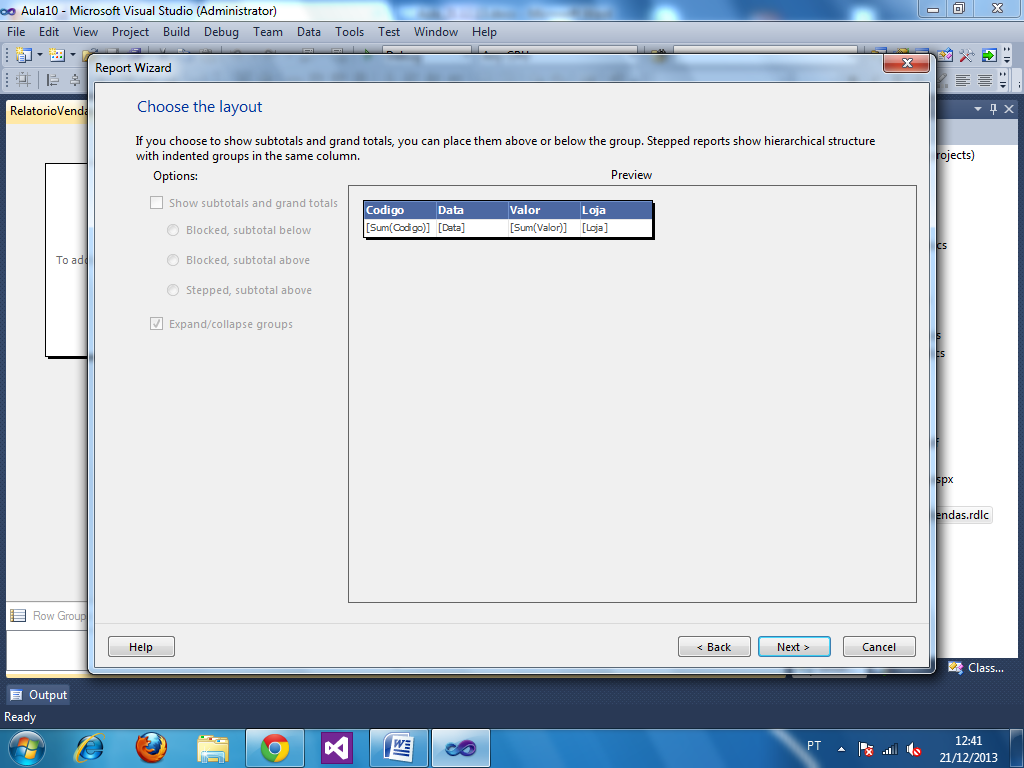
ReportWizard

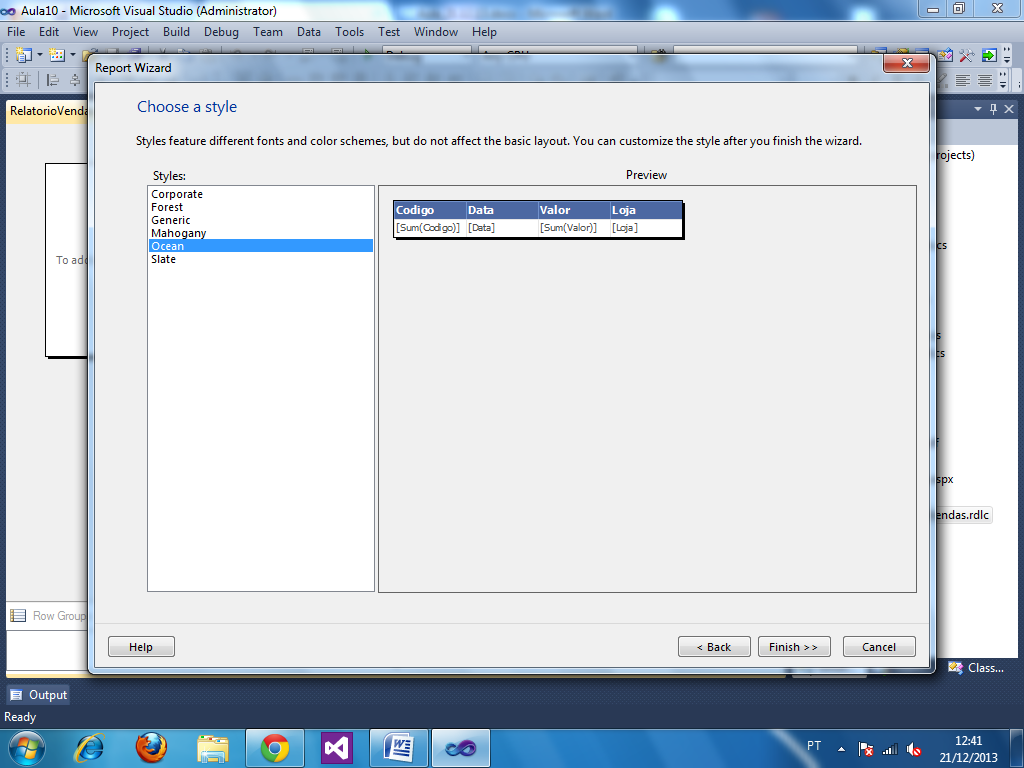


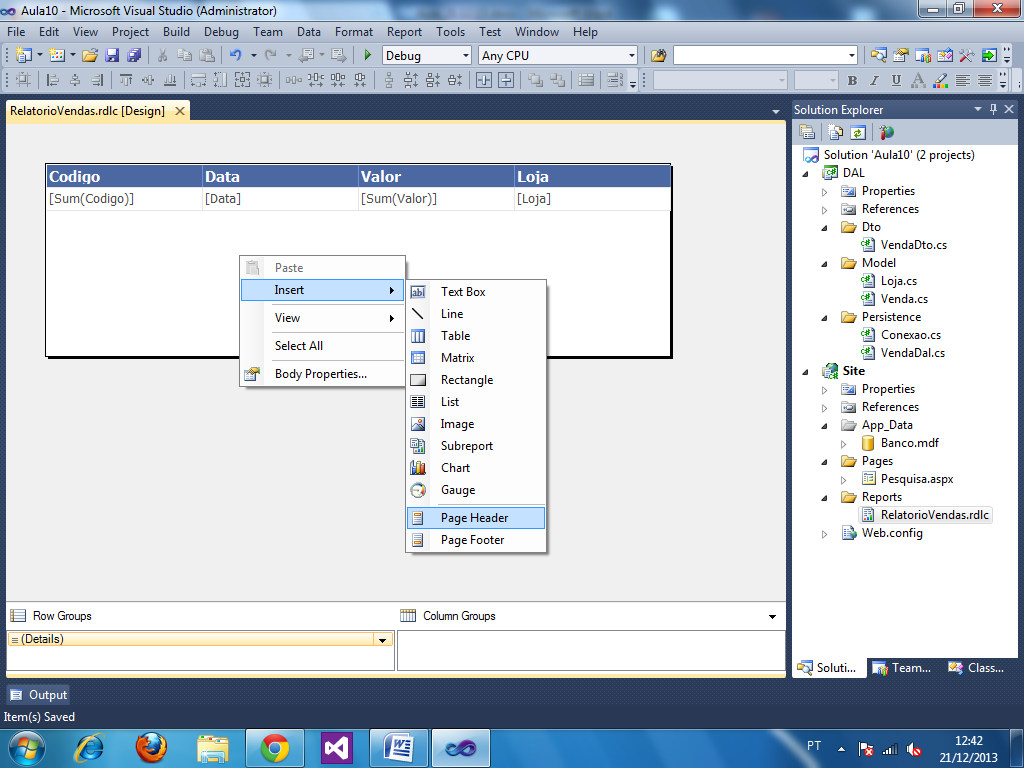


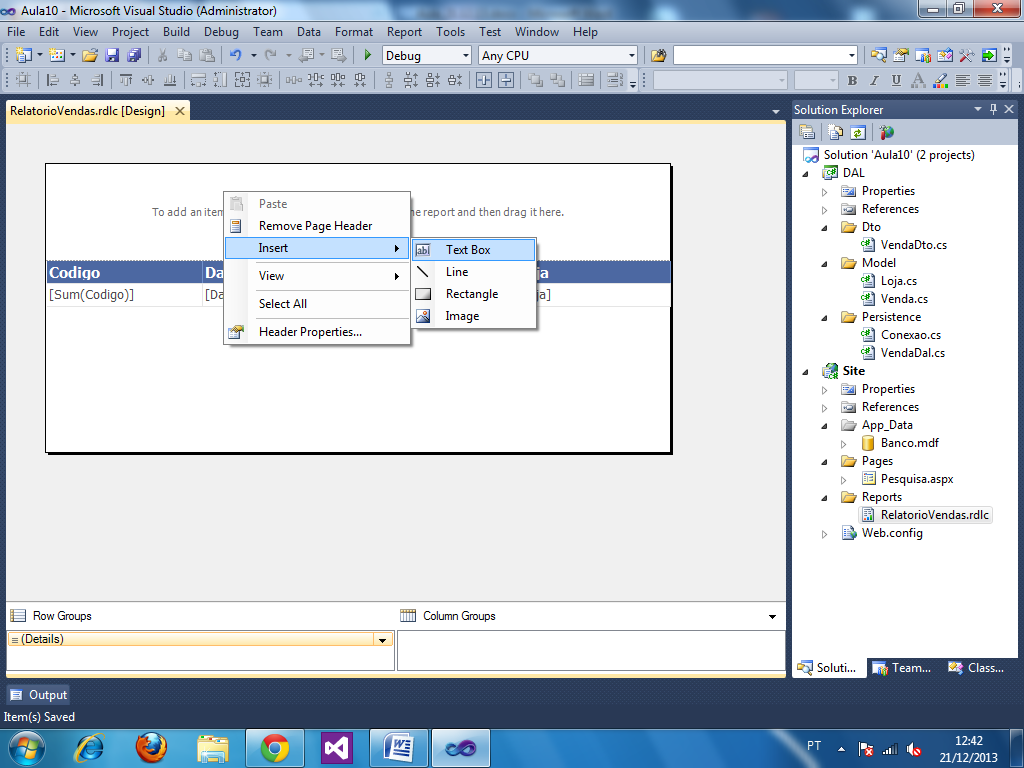


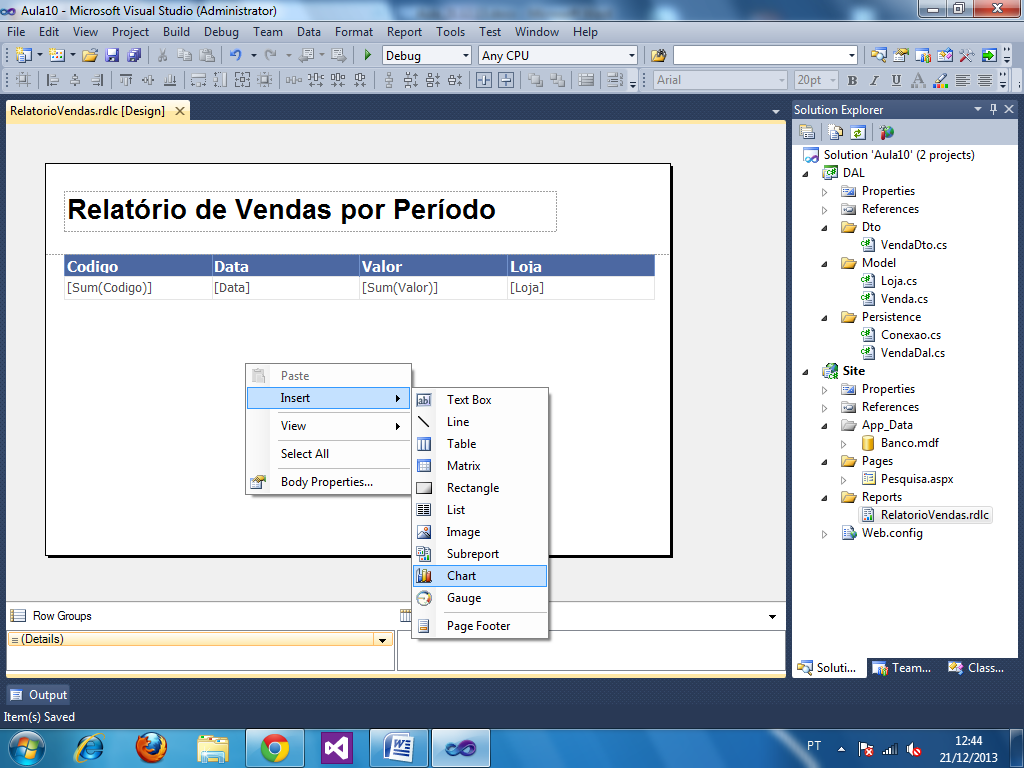


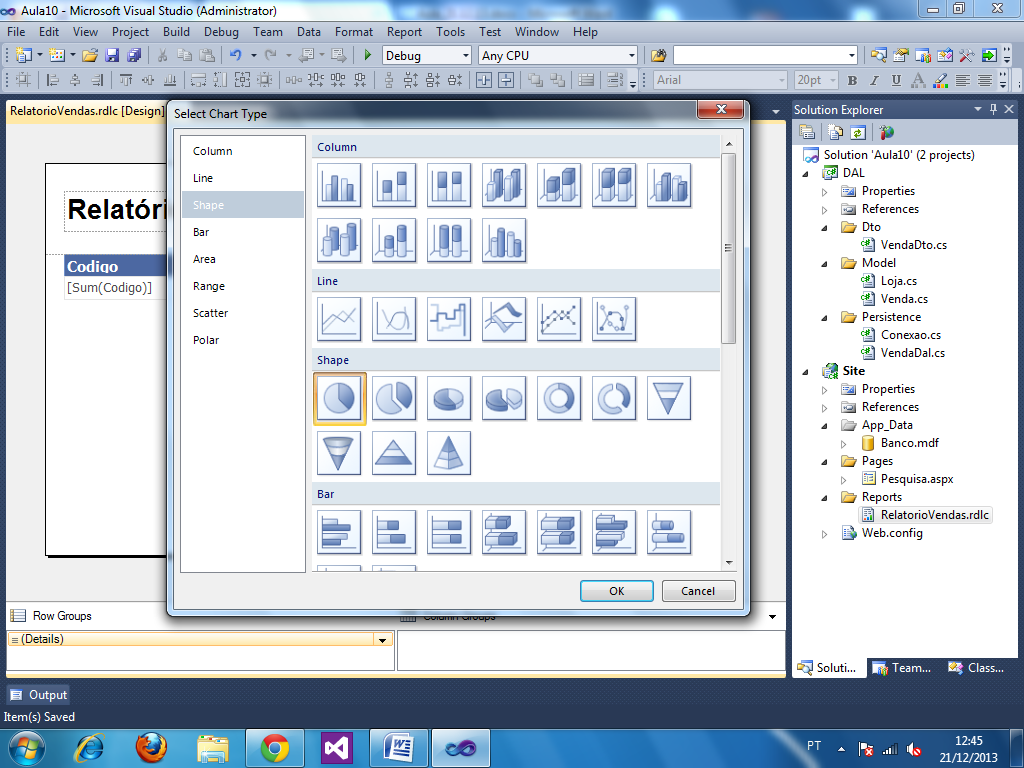


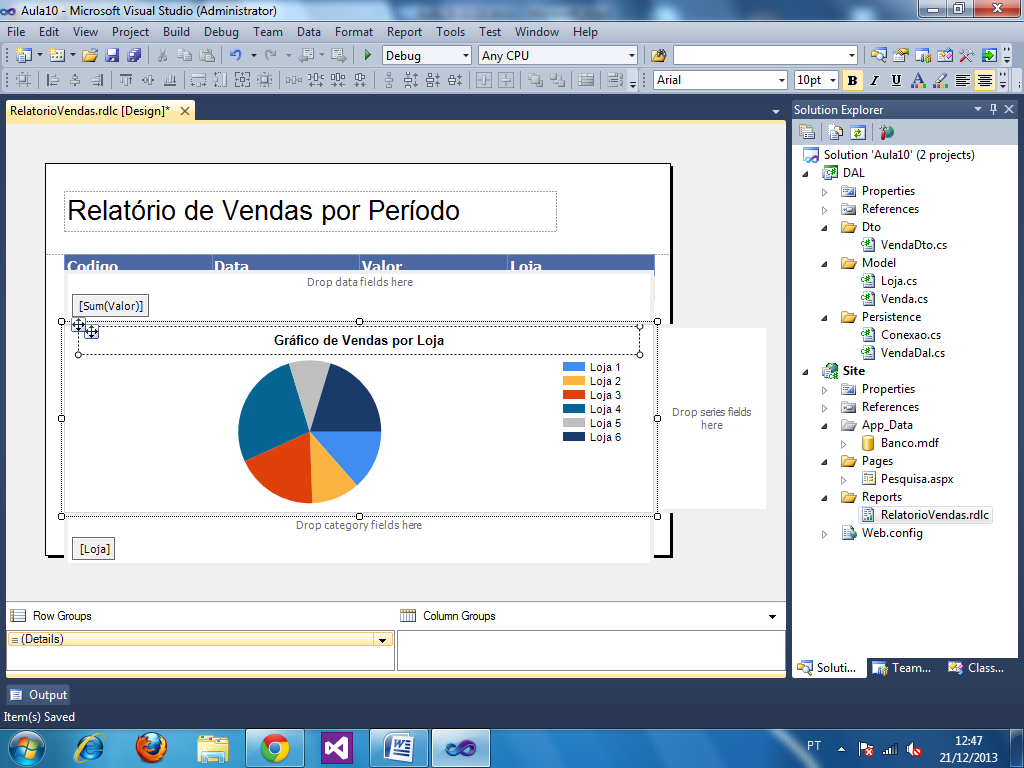


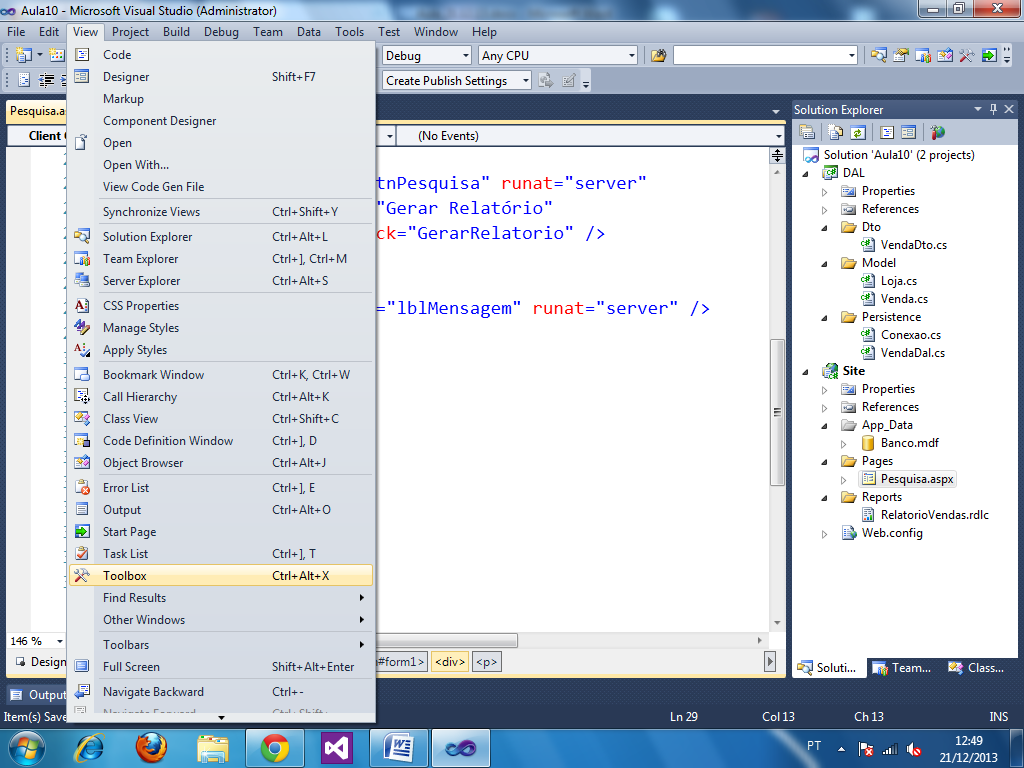


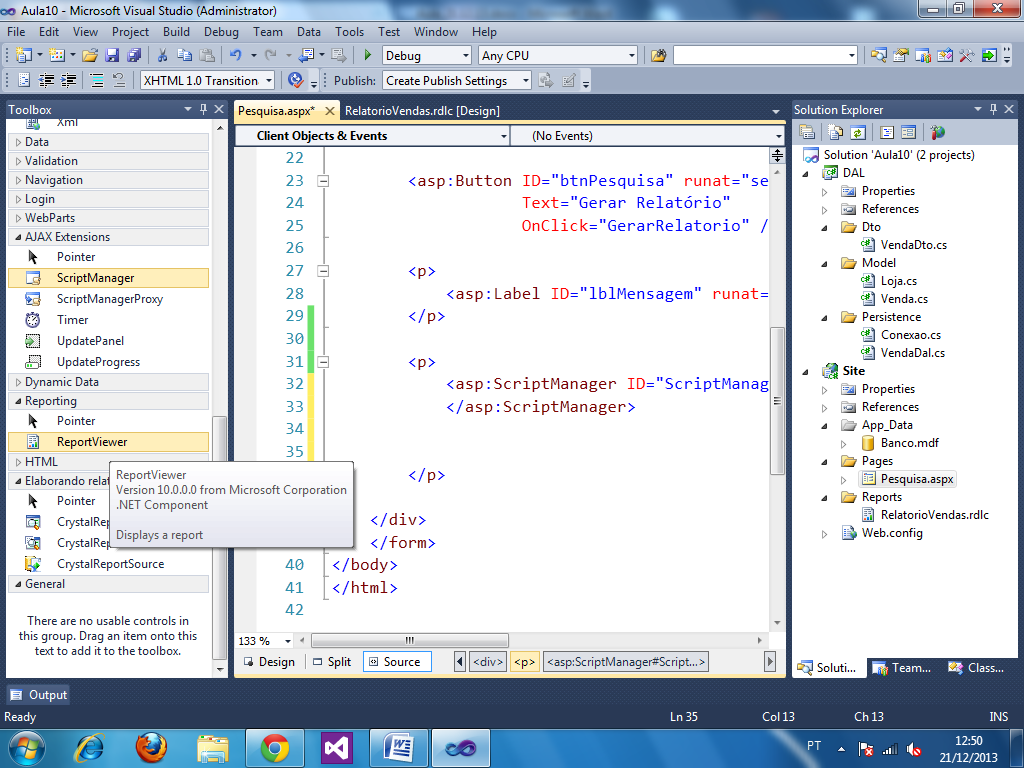












<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Pesquisa.aspx.cs" Inherits="Site.Pages.Pesquisa" %>

<%@ Register Assembly="Microsoft.ReportViewer.WebForms, Version=10.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"

Namespace="Microsoft.Reporting.WebForms" TagPrefix="rsweb" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<head runat="server">

<title></title>

<style type="text/css">

body { font-family: Verdana; font-size: 9pt; padding: 30px; }

</style>

</head>

<body>

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

<div>

<h3>Pesquisa de Vendas</h3>

Data de Início: <asp:TextBox ID="txtDataInicio" runat="server" />

Término: <asp:TextBox ID="txtDataTermino" runat="server" />

<asp:Button ID="btnPesquisa" runat="server"

Text="Gerar Relatório"

OnClick="GerarRelatorio" />

<p>

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

</p>

<p>

**<asp:ScriptManager ID="ScriptManager1" runat="server">**

**</asp:ScriptManager>**

**<rsweb:ReportViewer ID="ReportViewer" runat="server"**

**Width="800" Height="600">**

**</rsweb:ReportViewer>**

</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.Persistence;

using Microsoft.Reporting.WebForms;

using System.Data;

namespace Site.Pages

{

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

{

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

{

}

protected void GerarRelatorio(object sender, EventArgs e)

{

try

{

//Obter o relatorio (caminho)

string Path = HttpContext.Current.Server.

MapPath("/Reports/RelatorioVendas.rdlc");

//Resgatar as datas

DateTime dtIni = Convert.ToDateTime(txtDataInicio.Text);

DateTime dtFim = Convert.ToDateTime(txtDataTermino.Text);

using(VendaDal d = new VendaDal()) //inicializar

{

ReportDataSource dados = new ReportDataSource

("DSRelatorio", d.ListarVendas(dtIni, dtFim));

//Mostrar o relatorio

ReportViewer.LocalReport.ReportPath = Path;

//caminho do relatório

ReportViewer.LocalReport.DataSources.Add(dados);

//passando os dados

ReportViewer.DataBind(); //Exibir

} //dispose (destruir)

}

catch(Exception ex)

{

lblMensagem.Text = ex.Message;

}

}

}

}

