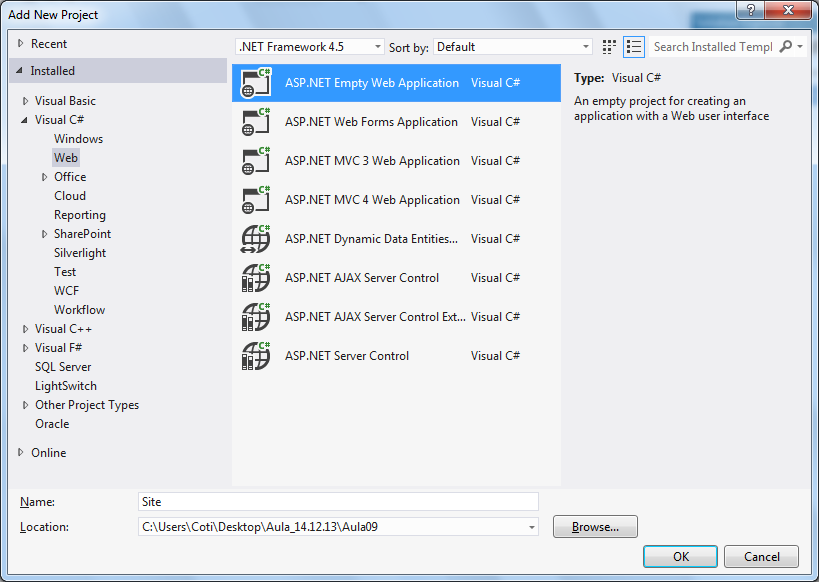
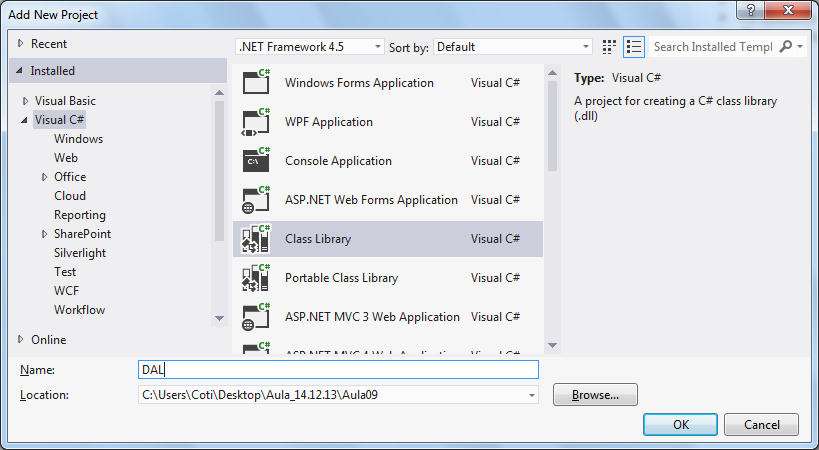


Criando o Projeto Web

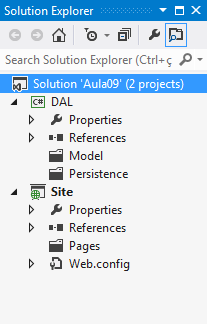


Data Access Layer

Camada de acesso a dados

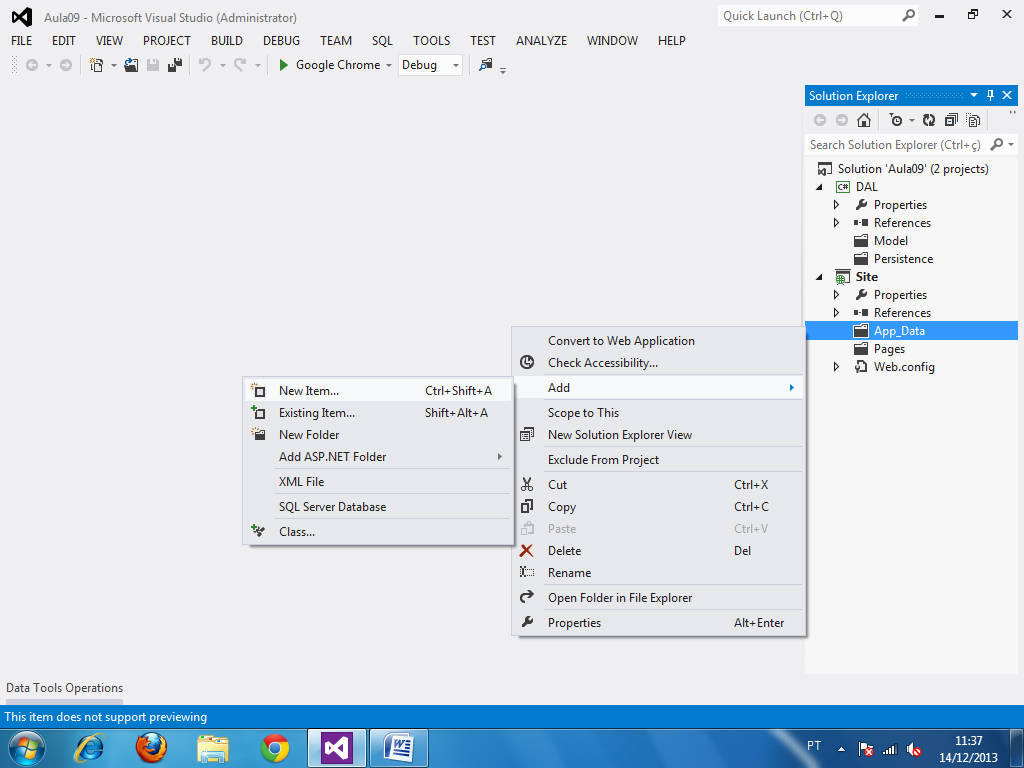


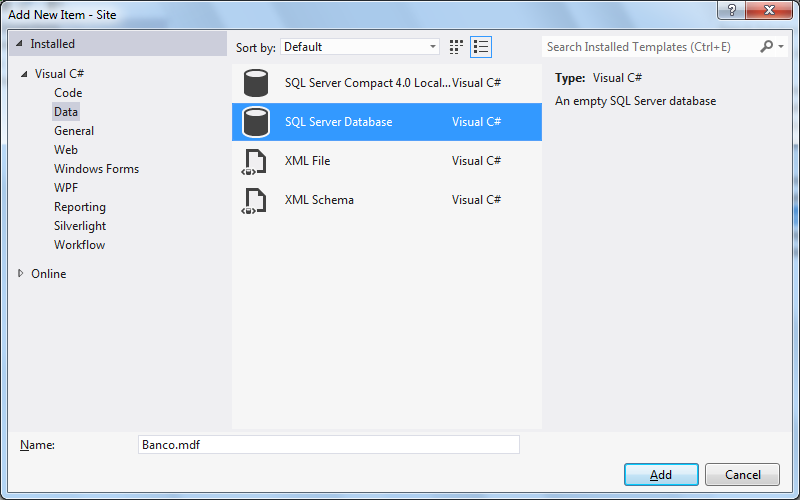
Estrutura do Projeto...

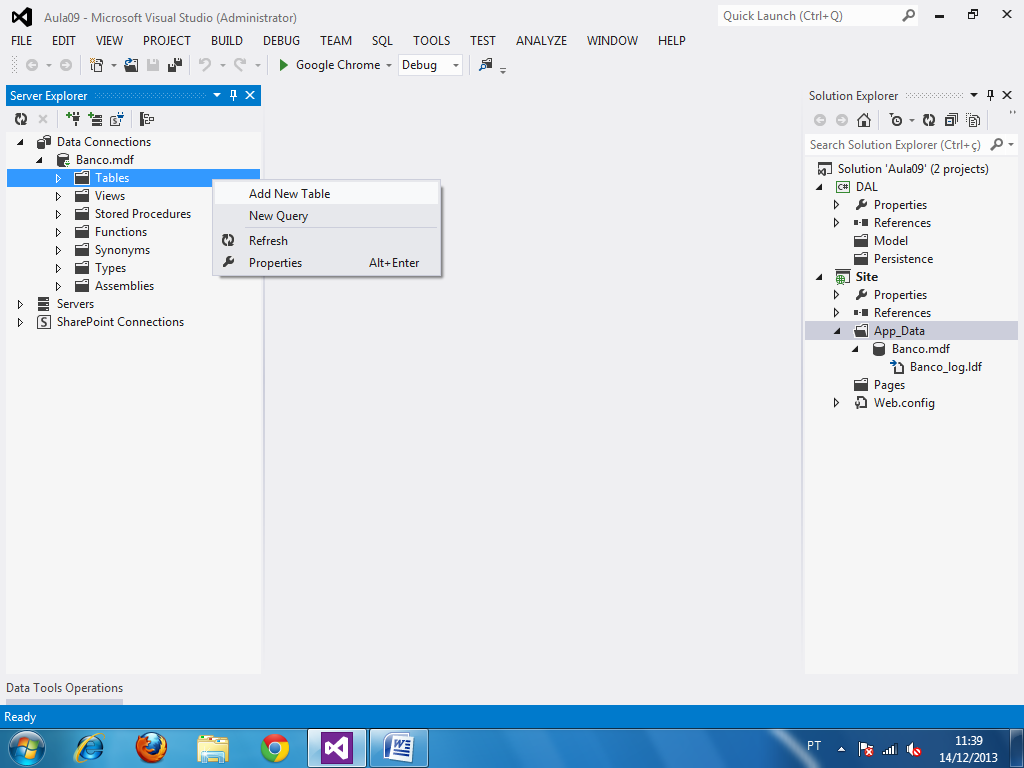


Criando a base de dados

Arquivo **.mdf**







go

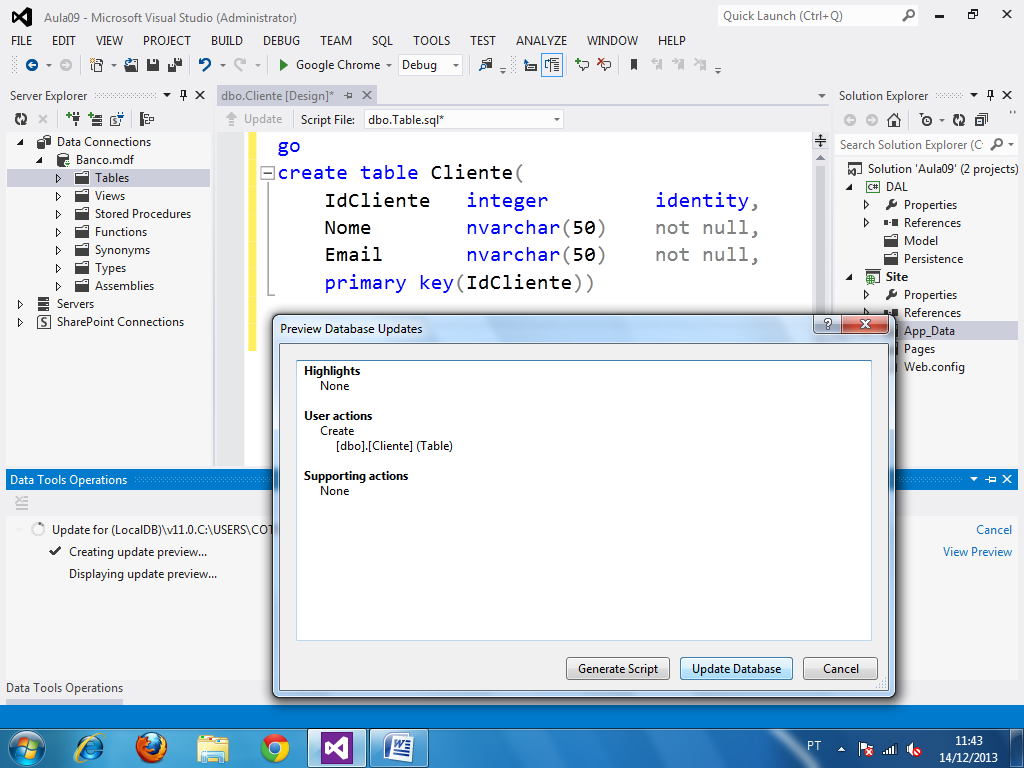
create table Cliente(

IdCliente integer identity,

Nome nvarchar(50) not null,

Email nvarchar(50) not null,

primary key(IdCliente))



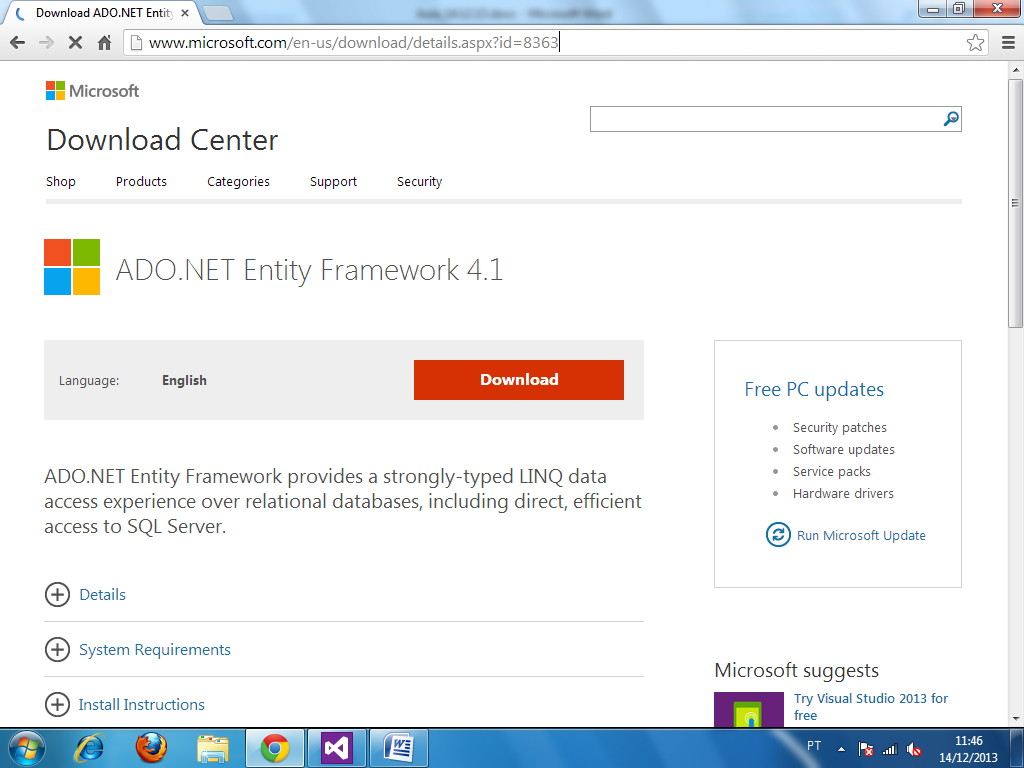
Entity Framework

Tecnologia .NET para acesso a bases de dados em aplicações.

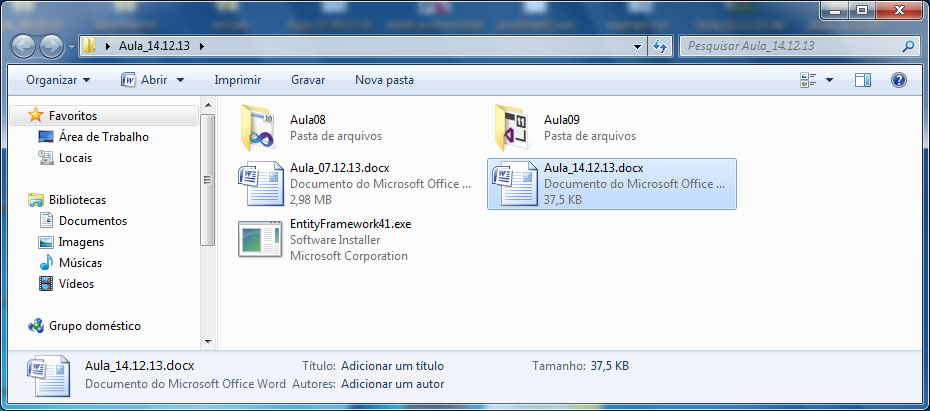
É considerado o framework padrão para aplicações .NET que acessar e persistem dados em BD.

**\*\* Instalar o EntityFramework**

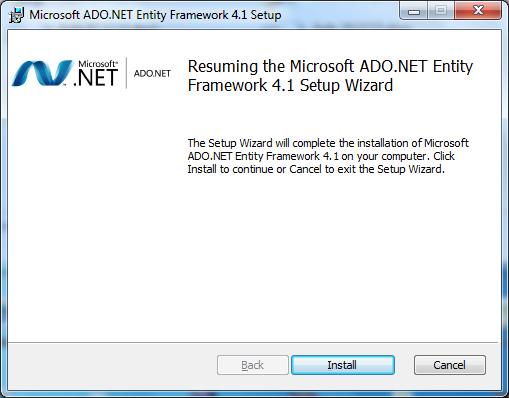
<http://www.microsoft.com/en-us/download/details.aspx?id=8363>

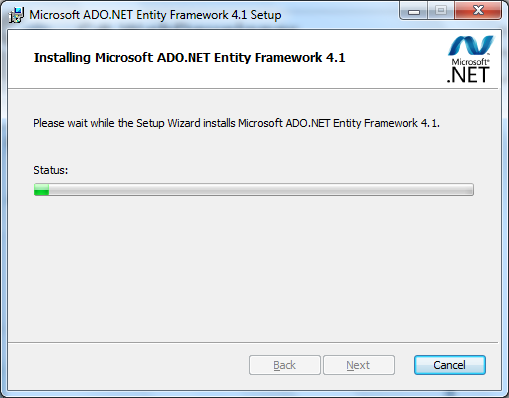


Instalando...

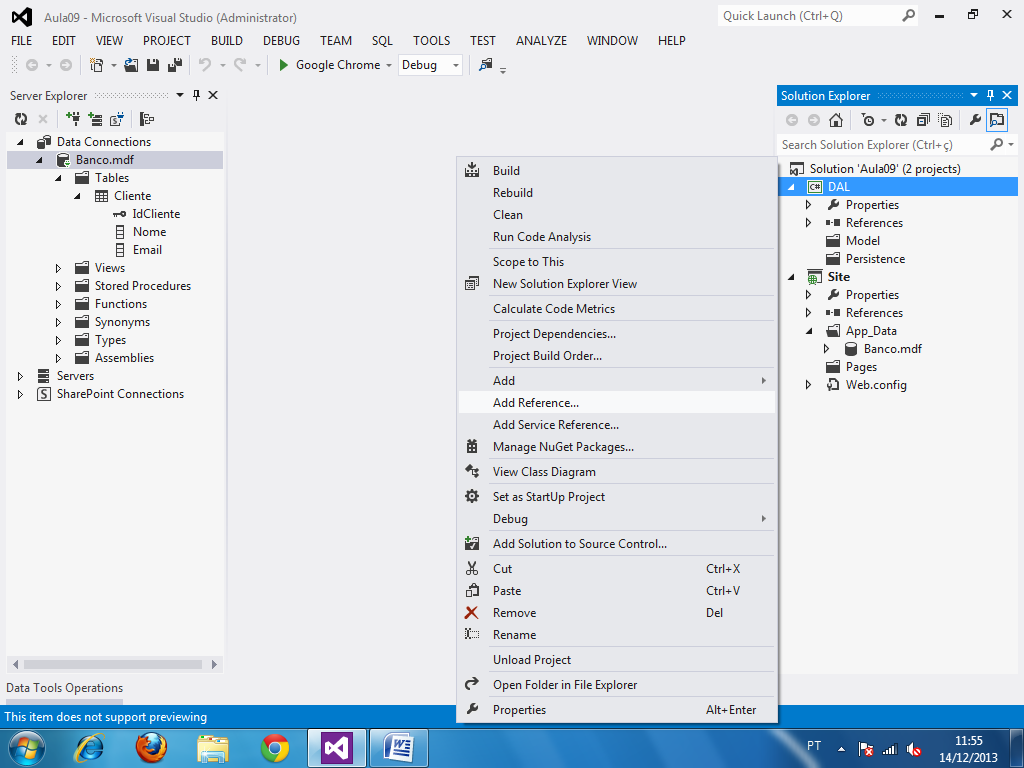


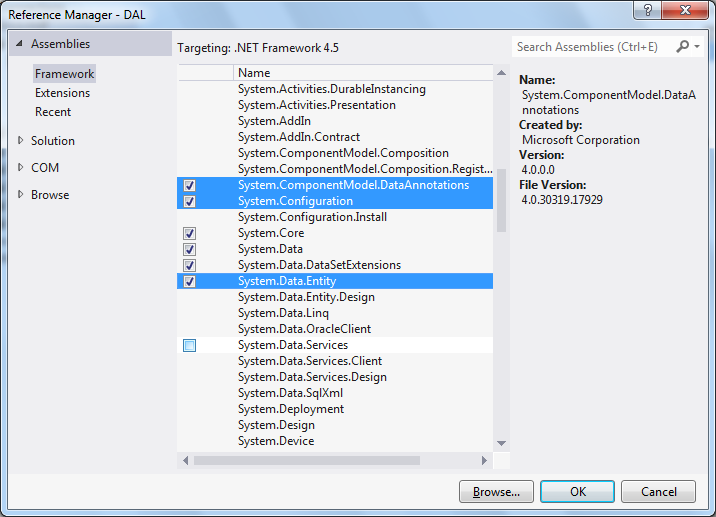
Executando...

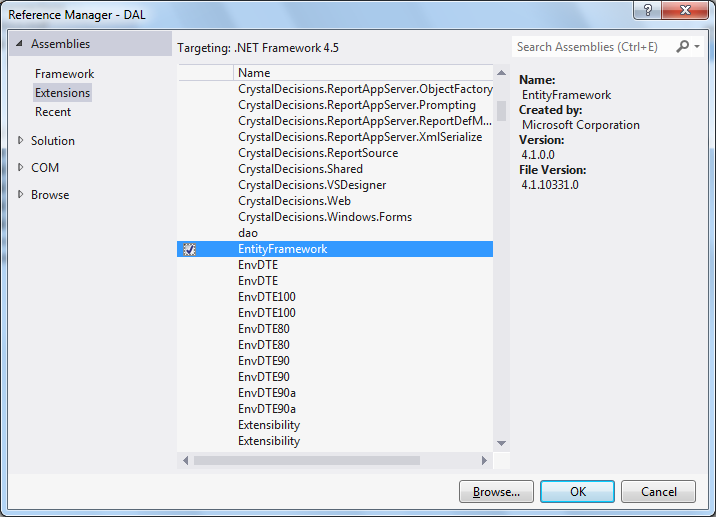




Adicionando referências no projeto DAL

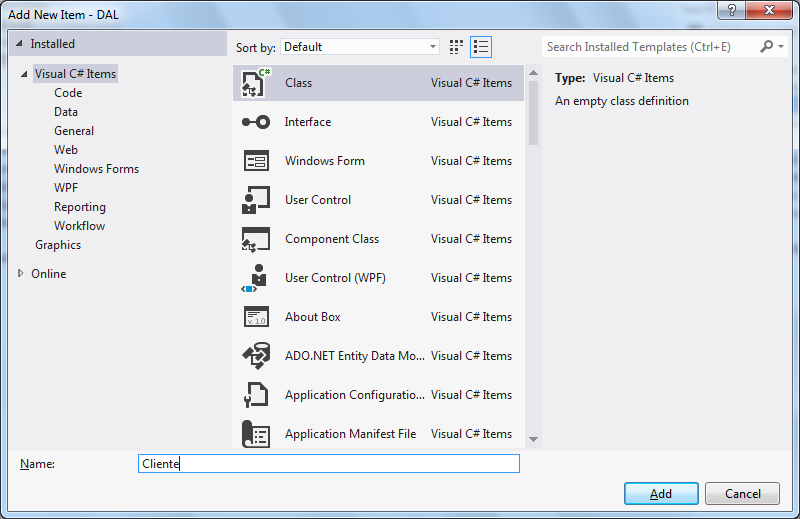






Passo 1: Criação da Classe Cliente

Modelagem das entidades do banco de dados



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DAL.Model

{

public class Cliente

{

public int idCliente { get; set; }

public string Nome { get; set; }

public string Email { get; set; }

}

}

Mapeamento Objeto Relacional (ORM)

Técnica utilizada para descrever Classes de um projeto como representações de entidades (tabelas) de uma base de dados.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.ComponentModel.DataAnnotations; //ORM

namespace DAL.Model

{

**[Table("Cliente")] //Anotação / etiqueta**

public class Cliente

{

**[Key] //chave primária**

**[DatabaseGenerated(DatabaseGeneratedOption.Identity)]**

**[Column("IdCliente")]**

public int IdCliente { get; set; }

**[Column("Nome")]**

public string Nome { get; set; }

**[Column("Email")]**

public string Email { get; set; }

}

}

Mapear a String de Conexão   
no \Web.config.xml

<?xml version="1.0"?>

<configuration>

**<connectionStrings>**

**<add**

**name="AULA"**

**connectionString="Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Coti\Desktop\Aula\_14.12.13\Aula09\Site\App\_Data\Banco.mdf;Integrated Security=True"**

**/>**

**</connectionStrings>**

<system.web>

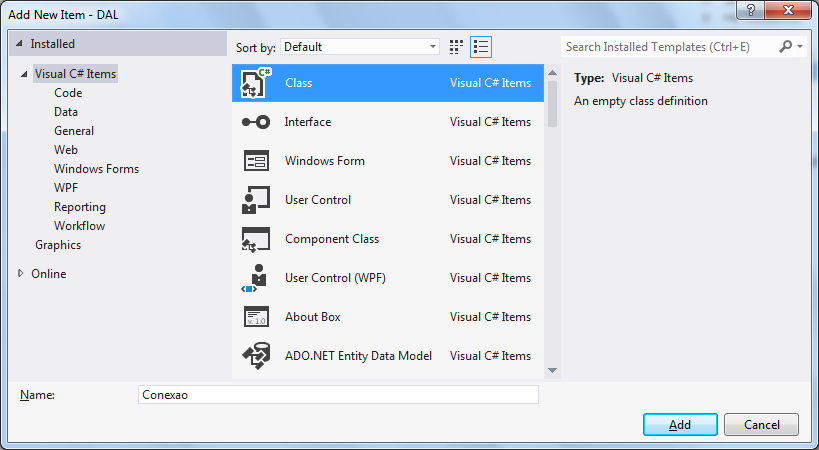
<compilation debug="true" targetFramework="4.5" />

<httpRuntime targetFramework="4.5" />

</system.web>

</configuration>

Classe para Conexão com o banco de dados…



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Configuration; //para pegar a conexão do Web.config.xml

using System.Data.Entity; //Classes do EF => DbContext e DbSet

using DAL.Model; //namespace das classes mapeadas (entidades)

namespace DAL.Persistence

{

//Para que a Classe de conexão possa suportar as operações do EF

//é necessário que esta classe herde System.Data.Entity.DbContext

//é o DbContext quem gerencia conexão e operações na BD

**public class Conexao : DbContext**

{

//Para que a Classe DbContext possa receber o caminho da ConnectionString

//é necessário que passemos essa referência por construtor

//[ctor] + 2x[tab] -> Construtor default (vazio)

public Conexao()

: base(ConfigurationManager.ConnectionStrings

["AULA"].ConnectionString)

{

//Iremos passar para o construtor da Classe PAI (DbContext) o caminho

//do BD, para que quando a Classe Conexao seja instanciada,

//o DbContext imediatamente receba a connectionstring

}

//DbSet -> Permitir que as Classes mapeadas sejam manipuladas pelo EF

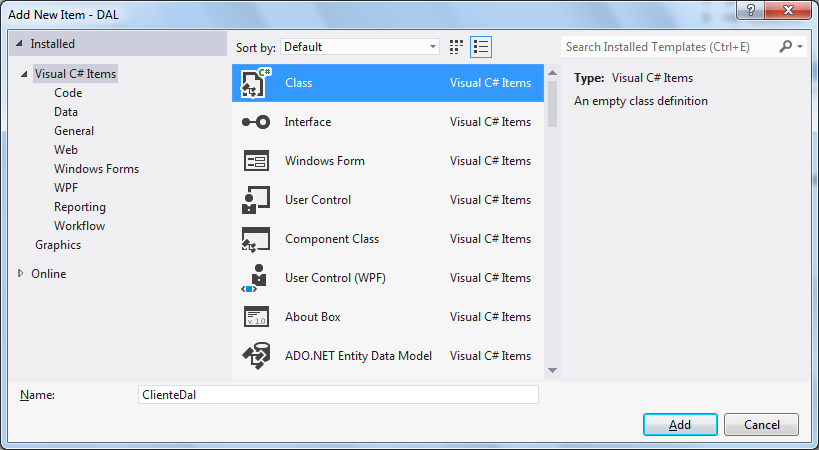
//já com operações de INSERT, DELETE, UPDATE E SELECT definidas

**public DbSet<Cliente> Cliente { get; set; }**

}

}

Classe de persistência...



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DAL.Model;

namespace DAL.Persistence

{

public class ClienteDal

{

public void Salvar(Cliente c)

{

try

{

Conexao Con = new Conexao(); //Classe do DbContext

Con.Cliente.Add(c); //insert do objeto Cliente -> c

Con.SaveChanges(); //executar

}

catch(Exception e)

{

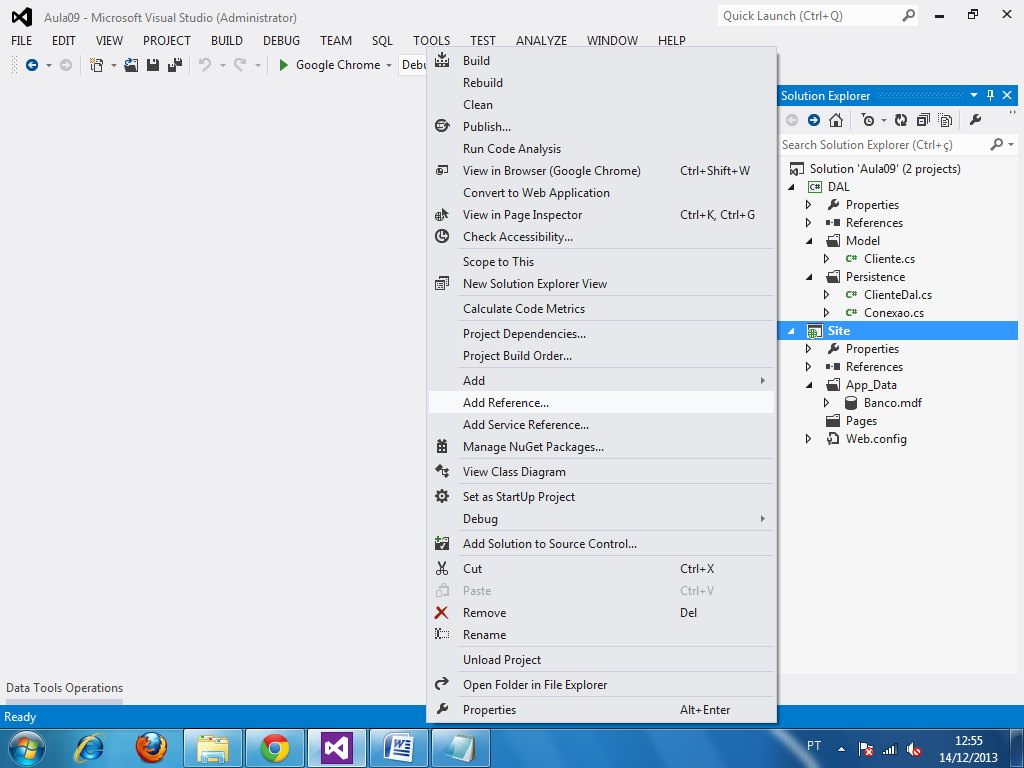
throw new Exception("Erro ao cadastrar Cliente: " + e.Message);

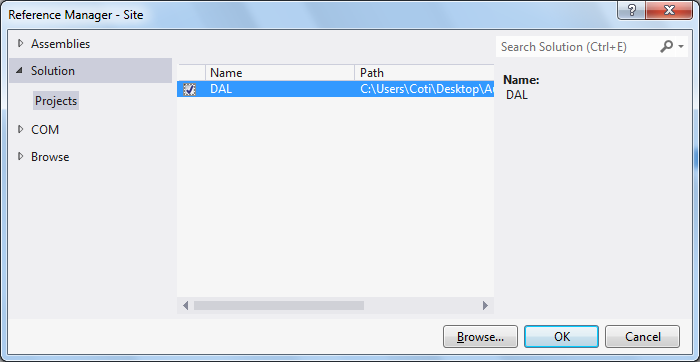
}

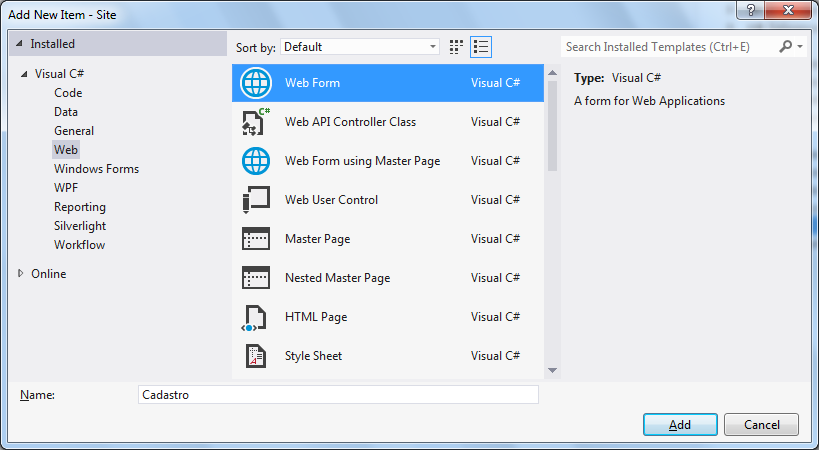
}

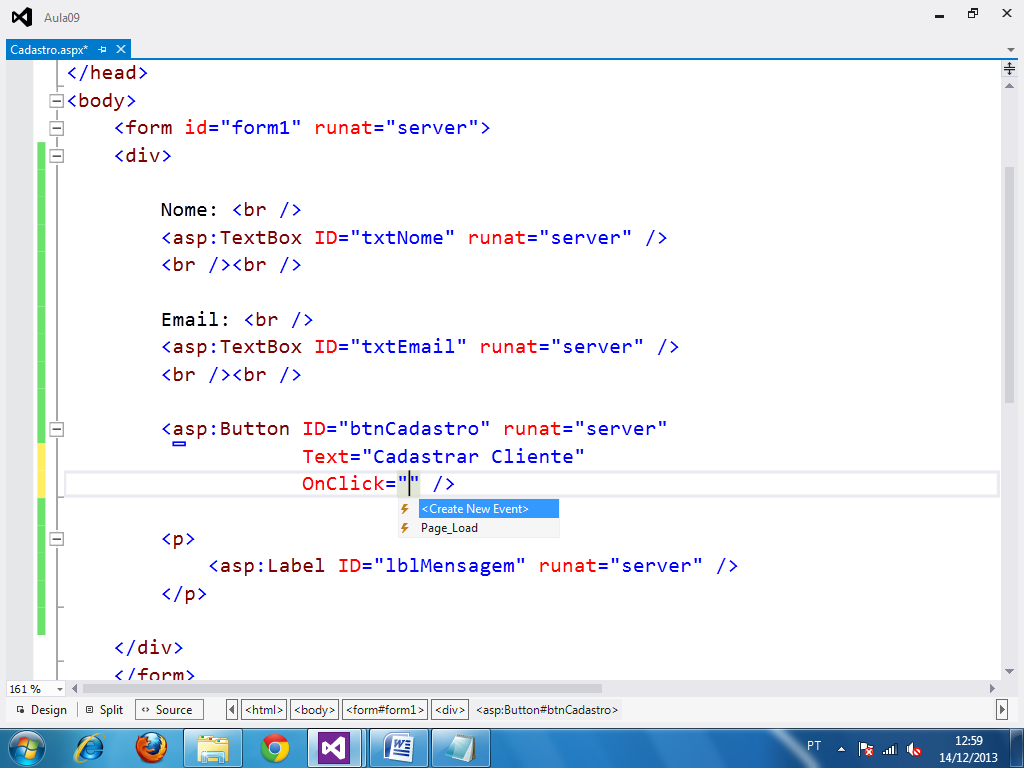
}

}









<%@ 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>

Nome: <br />

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

<br /><br />

Email: <br />

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

<br /><br />

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

Text="Cadastrar Cliente"

**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

{

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

{

}

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

{

try

{

Cliente c = new Cliente();

c.Nome = txtNome.Text;

c.Email = txtEmail.Text;

ClienteDal d = new ClienteDal();

d.Salvar(c);

lblMensagem.Text = "Cliente cadastrado com sucesso.";

}

catch(Exception ex)

{

lblMensagem.Text = ex.Message;

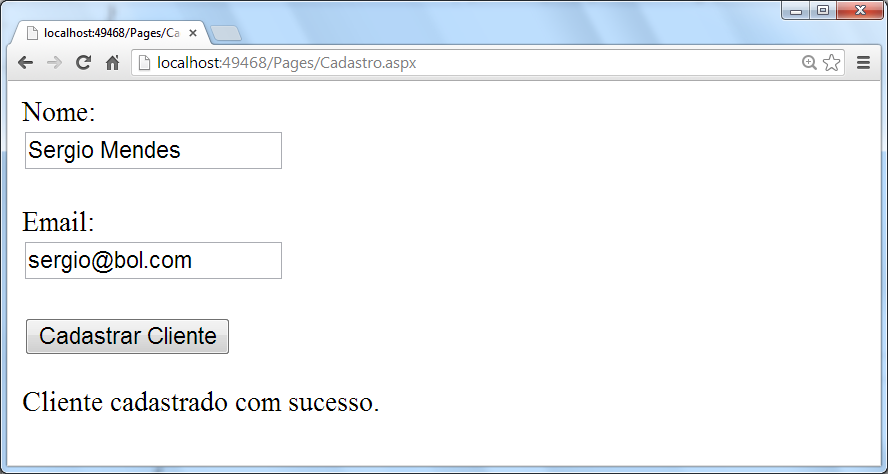
}

}

}

}

Executando...



No SqlServer...

