

Estrutura de um projeto MVC

* App\_Data

Diretorio para arquivos de fontes de dados (.mdf ou xml)

* Content

Diretório padrão do projeto MVC para arquivos CSS

* Controllers

Diretório para armazenar classes de controle em projetos MVC

* Models

Diretório para armazenar classes de modelo

* Scripts

Diretório para armazenar arquivos javascript (jquery)

* Views

Diretório para armazenar páginas web

* + Shared

Diretório para páginas de layout (template).

ASPX 🡪 extensão padrão de páginas web

CSHTML 🡪 padrão @Razor

MVC (Model View e Controller)

* **Model** 🡪 Camada do MVC responsável pela representação dos dados de entrada / saída do projeto
* **Controller** 🡪 Camada do MVC similar ao CodeBehind, pois é responsavel pelo gerenciamento das ações realizadas pelo usuário da aplicação
* **Views** 🡪 Págima Web (Interface do sistema com usuário da aplicação)

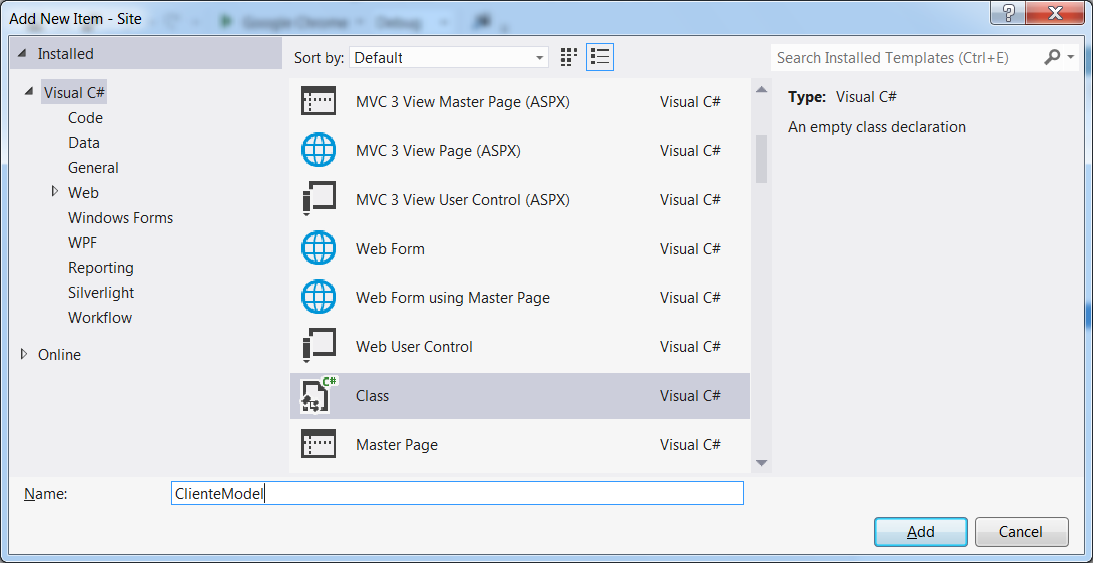
Objetivo:

Definir uma página para cadastro de Clientes contendo:

* Nome do Cliente
* Email
* Logradouro
* Cidade
* Estado

Passo 1: Criação do Modelo

Classe para representar os dados capturados pelo projeto MVC no cadastro de Clientes...



using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Site.Models

{

public class ClienteModel

{

#region Propriedades

public string Nome { get; set; }

public string Email { get; set; }

public string Logradouro { get; set; }

public string Cidade { get; set; }

public string Estado { get; set; }

#endregion

}

}

Mapeamentos na Classe de Modelo

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

**using System.ComponentModel;**

**using System.ComponentModel.DataAnnotations;**

namespace Site.Models

{

public class ClienteModel

{

#region Propriedades

**[Required(ErrorMessage = "Por favor, informe o nome do cliente.")]**

**[Display(Name = "Nome do Cliente:")]**

public string Nome { get; set; }

**[Required(ErrorMessage = "Por favor, informe o email do cliente.")]**

**[Display(Name = "Email:")]**

public string Email { get; set; }

**[Required(ErrorMessage = "Por favor, informe o logradouro.")]**

**[Display(Name = "Logradouro:")]**

public string Logradouro { get; set; }

**[Required(ErrorMessage = "Por favor, informe a cidade.")]**

**[Display(Name = "Cidade:")]**

public string Cidade { get; set; }

**[Required(ErrorMessage = "Por favor, informe o estado.")]**

**[Display(Name = "Estado:")]**

public string Estado { get; set; }

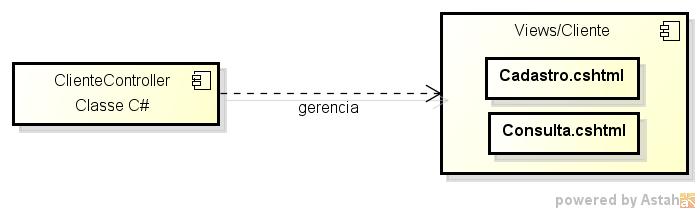
#endregion

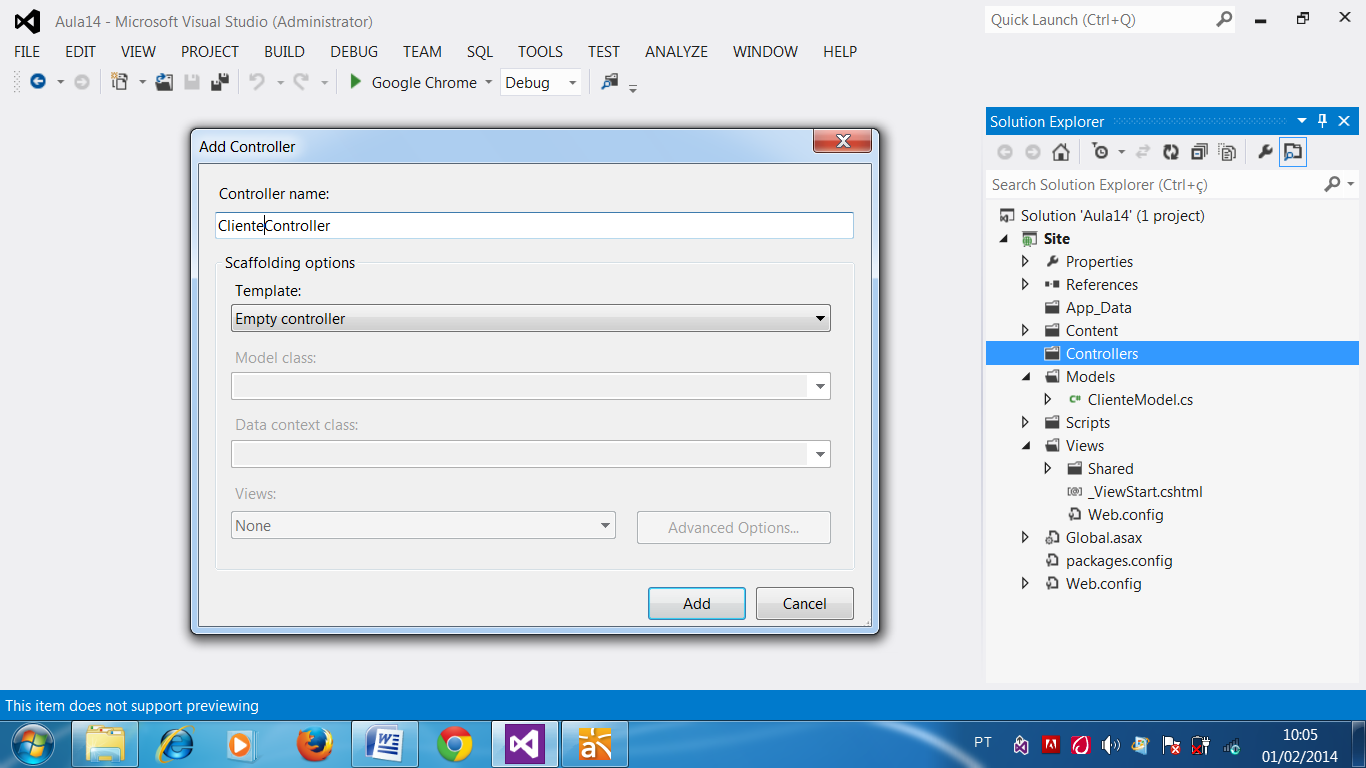
}

}

Passo 2: Criação do Controle

Em MVC, O Controle deverá ser criado antes das páginas.





using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Site.Controllers

{

public class ClienteController : Controller

{

/\*

\* A primeira responsabilidade de uma classe de controle

\* é definir ROTAS para

\* acesso às páginas que estarão subordinadas a este controle

\*/

#region Métodos para definição de rotas de páginas

//ROTA => /Cliente/Cadastro

public ActionResult Cadastro() //Page\_Load

{

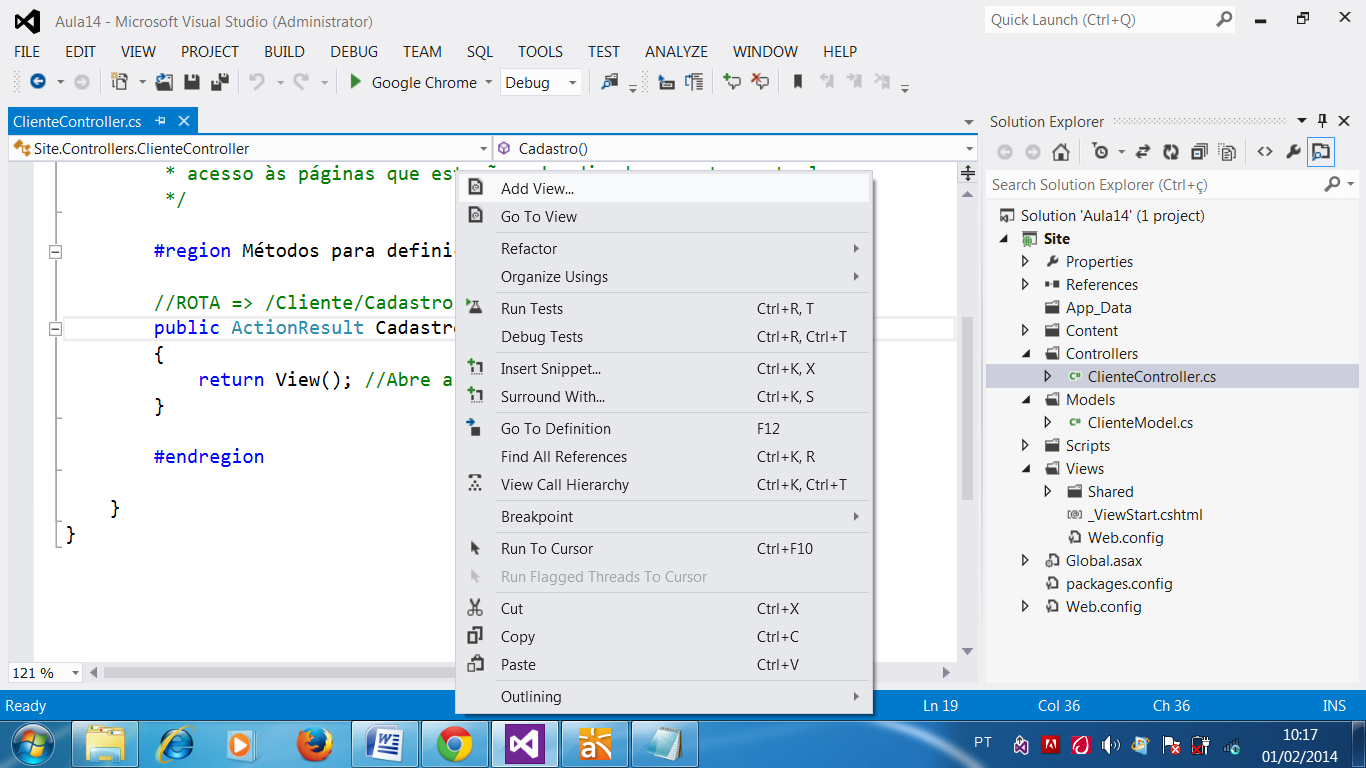
return View(); //Abre a página

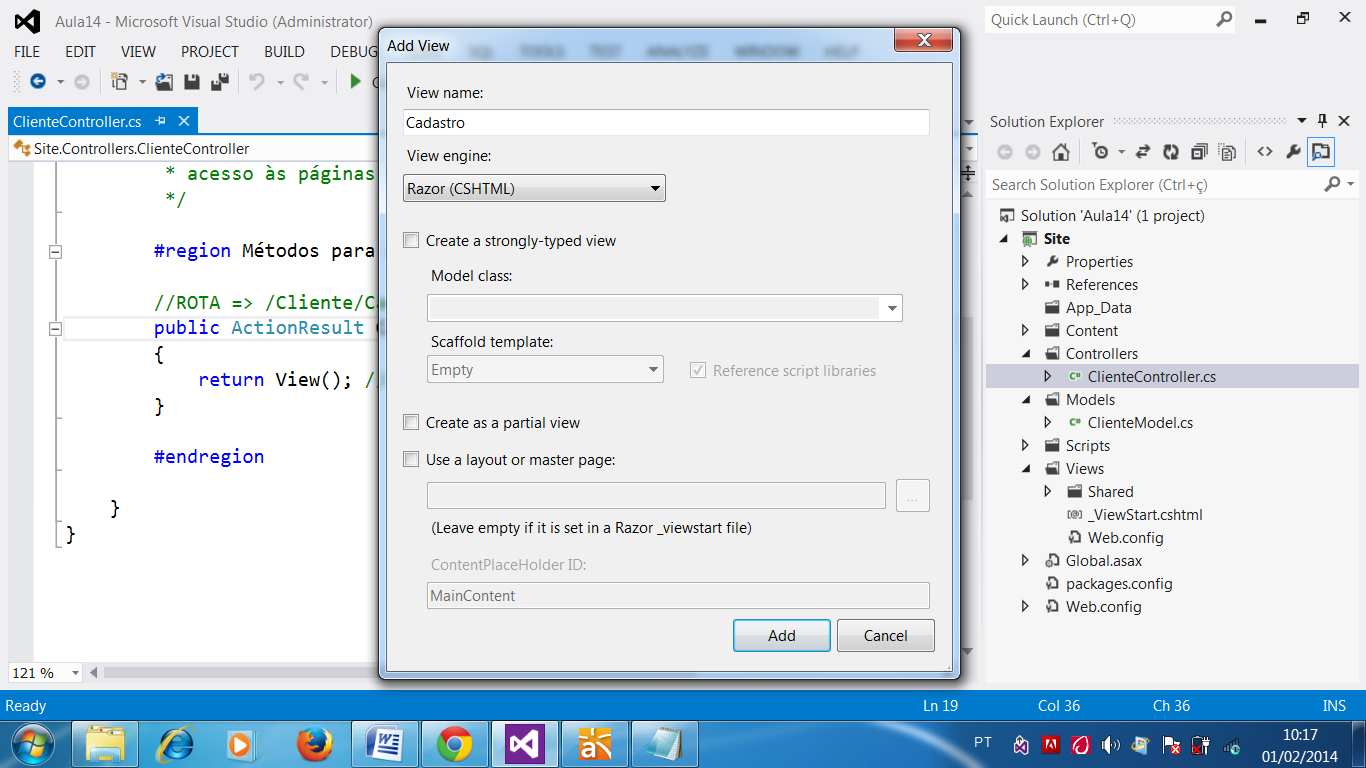
}

#endregion

}

}





<!DOCTYPE html>

<html>

<head>

<title>Cadastro</title>

</head>

<body>

<div>

<h3>Cadastro de Clientes</h3>

<hr />

</div>

</body>

</html>

Para definir a página inicial de um projeto MVC3

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 = "Cliente", 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");

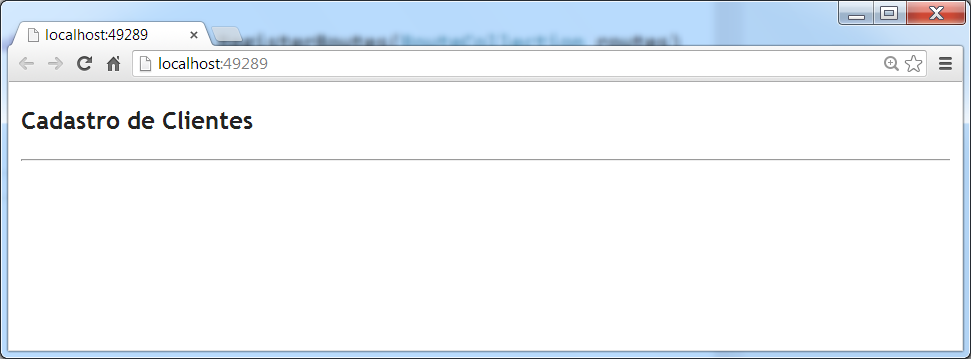
RegisterGlobalFilters(GlobalFilters.Filters);

RegisterRoutes(RouteTable.Routes);

}

}

}



@Razor

Linguagem utilizada para definição dos componentes das páginas em um projeto MVC que estarão vinculados ao modelo e ao controle.

<!-- Vinculando a página à Classe de Modelo -->

@model Site.Models.ClienteModel

<!DOCTYPE html>

<html>

<head>

<title>Cadastro</title>

</head>

<body>

<div>

<h3>Cadastro de Clientes</h3>

<hr />

<!-- Formulário padrao Razor (MVC) -->

@using (Html.BeginForm("CadastrarCliente", "Cliente", FormMethod.Post))

{

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

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

<br /><br />

@Html.LabelFor(model => model.Email) <br />

@Html.TextBoxFor(model => model.Email)

<br /><br />

@Html.LabelFor(model => model.Logradouro) <br />

@Html.TextBoxFor(model => model.Logradouro)

<br /><br />

@Html.LabelFor(model => model.Cidade) <br />

@Html.TextBoxFor(model => model.Cidade)

<br /><br />

@Html.LabelFor(model => model.Estado) <br />

@Html.TextBoxFor(model => model.Estado)

<br /><br />

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

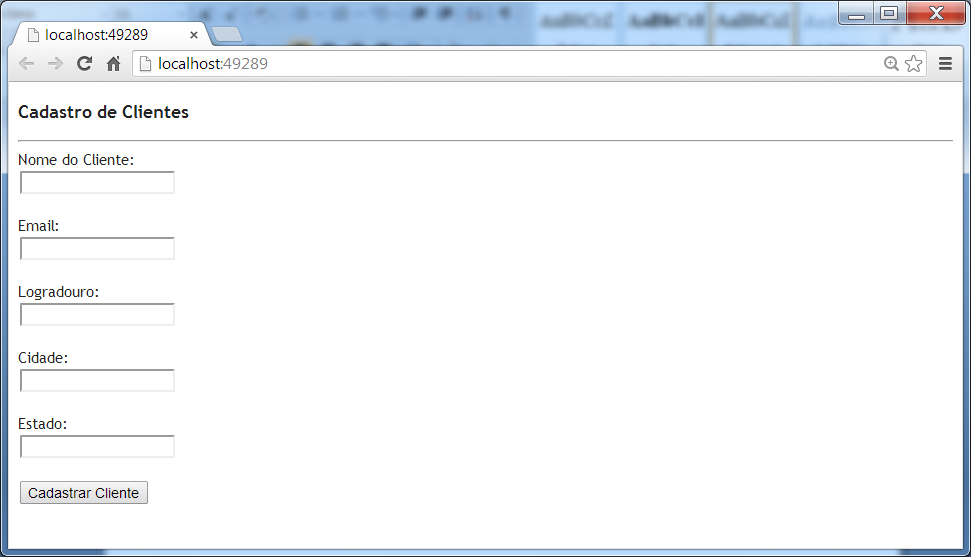
}

</div>

</body>

</html>

Resultado...



using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using Site.Models; //namespace da Classe de Modelo

namespace Site.Controllers

{

public class ClienteController : Controller

{

/\*

\* A primeira responsabilidade de uma classe de controle

é definir ROTAS para

\* acesso às páginas que estarão subordinadas a este controle

\*/

#region Métodos para definição de rotas de páginas

//ROTA => /Cliente/Cadastro

public ActionResult Cadastro() //Page\_Load

{

return View(); //Abre a página

}

#endregion

#region Métodos para resgate de dados de formulários

[HttpPost] //Definir que o método é executado por um formulário POST

public ActionResult CadastrarCliente(ClienteModel model)

//Classe de modelo (dados)

{

//...

return View("Cadastro"); //redirecionamento

}

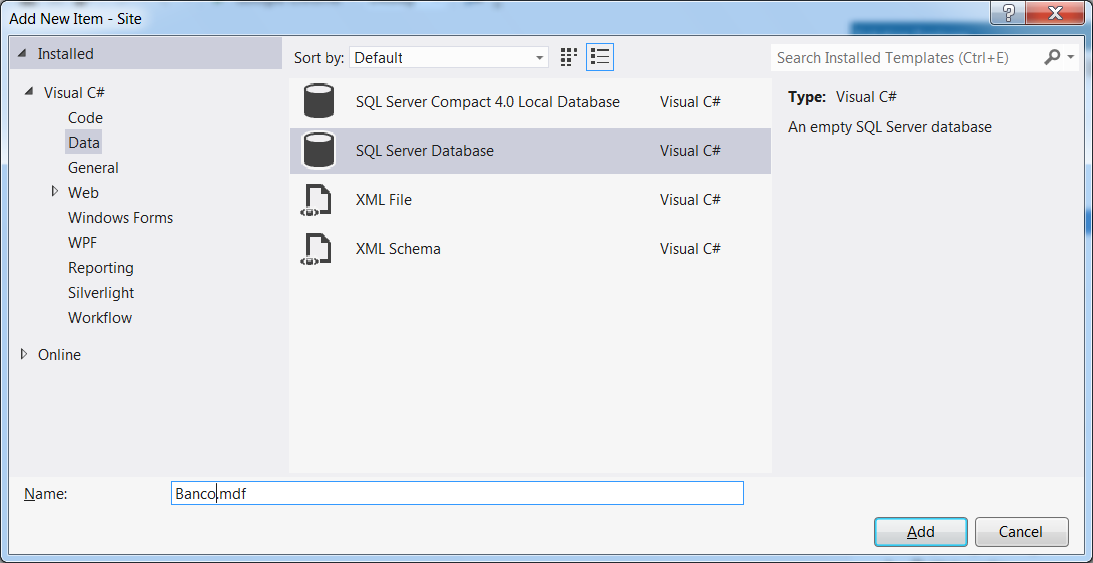
#endregion

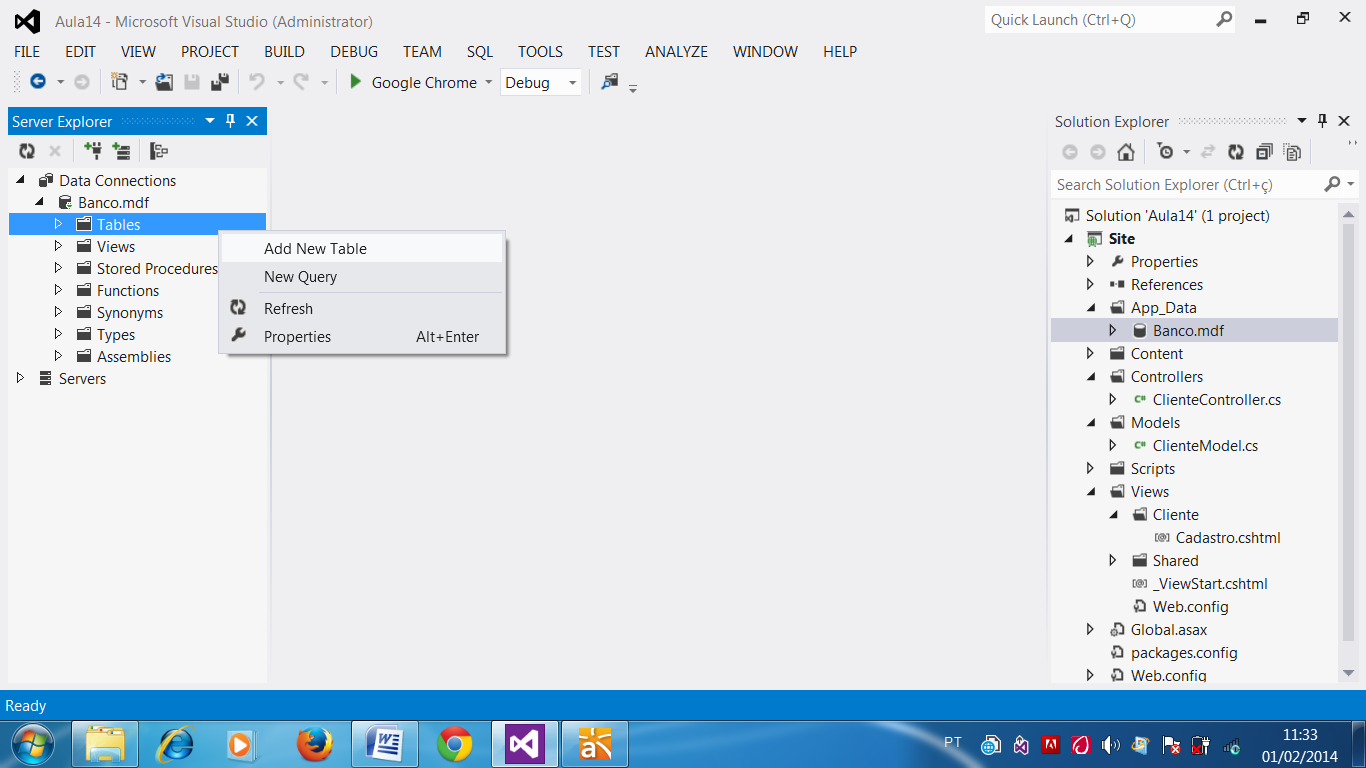
}

}

Criação do banco de dados

**.mdf** (Microsoft Database File)





go

create table Cliente(

IdCliente integer identity,

Nome nvarchar(50) not null,

Email nvarchar(50) not null,

primary key(IdCliente))

go

create table Endereco(

IdEndereco integer identity,

Logradouro nvarchar(50) not null,

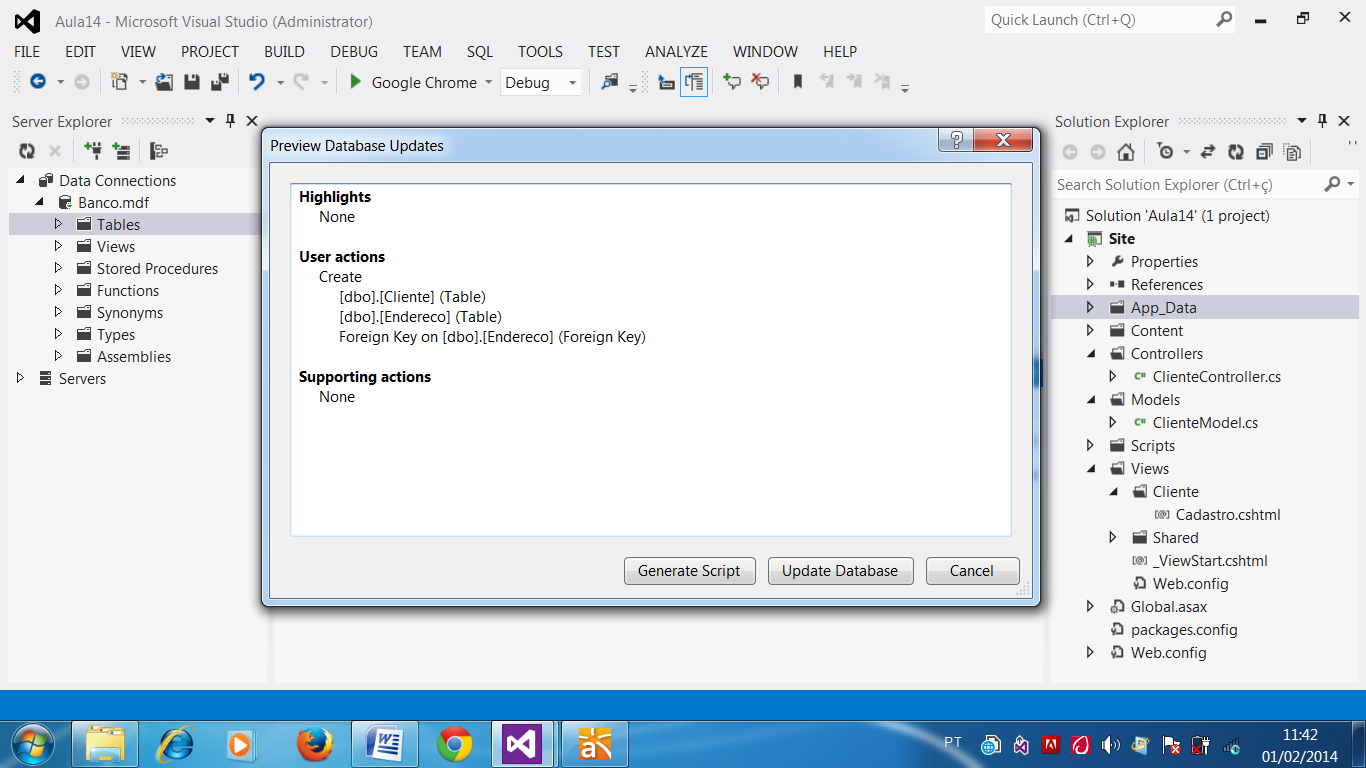
Cidade nvarchar(50) not null,

Estado nvarchar(50) not null,

IdCliente integer not null unique,

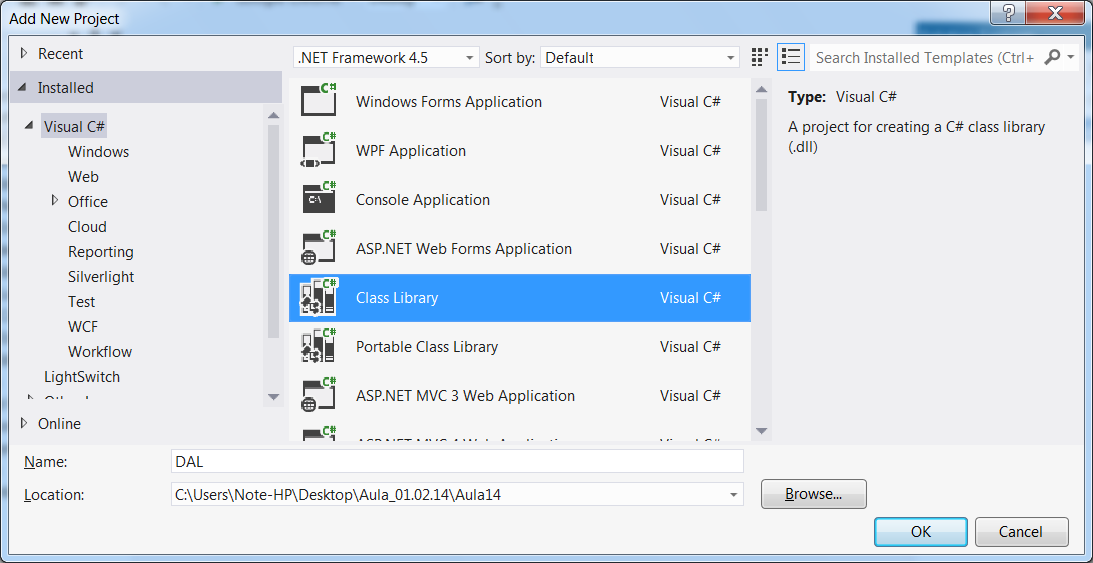
primary key(IdEndereco),

foreign key(IdCliente) references Cliente(IdCliente))



Data Access Layer

Camada de acesso a dados



Classes para representação das   
entidades do banco de dados

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DAL.Entities

{

public class Cliente

{

public int IdCliente { get; set; }

public string Nome { get; set; }

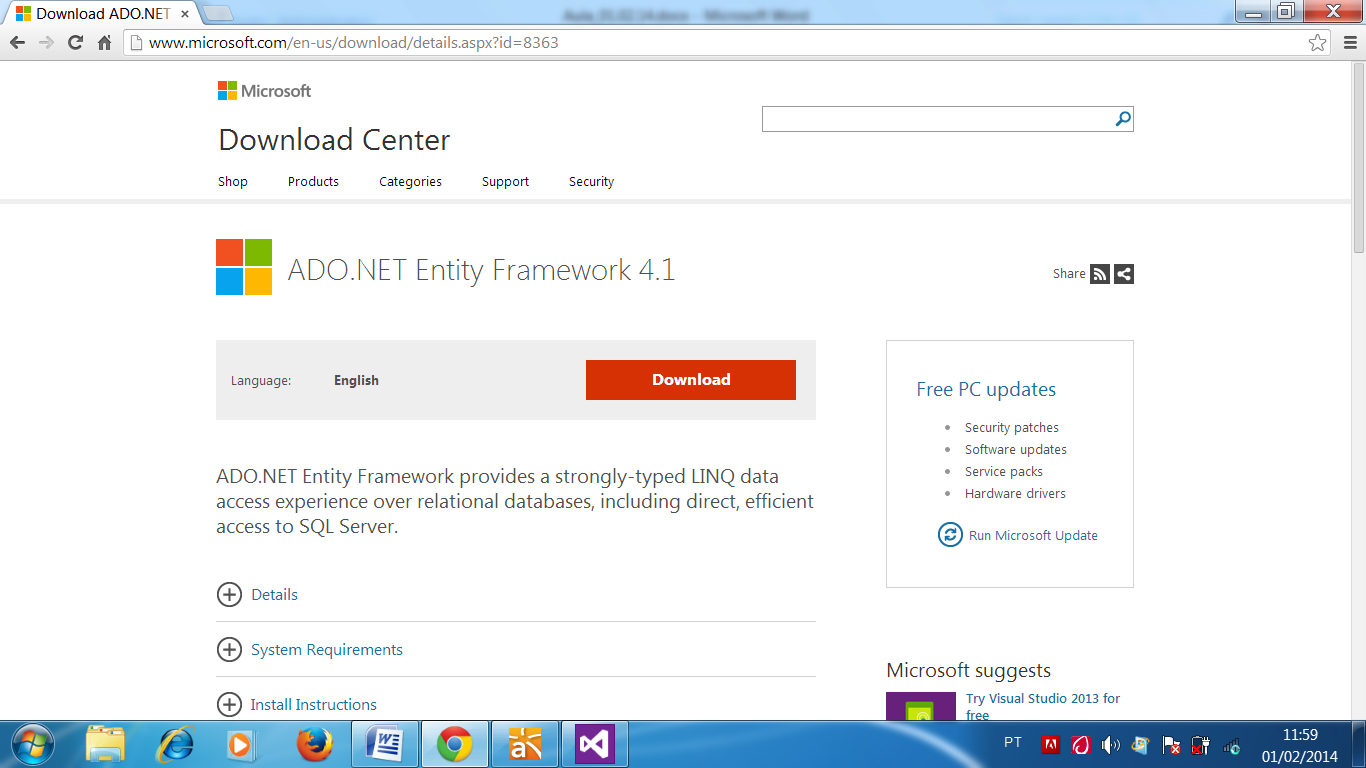
public string Email { get; set; }

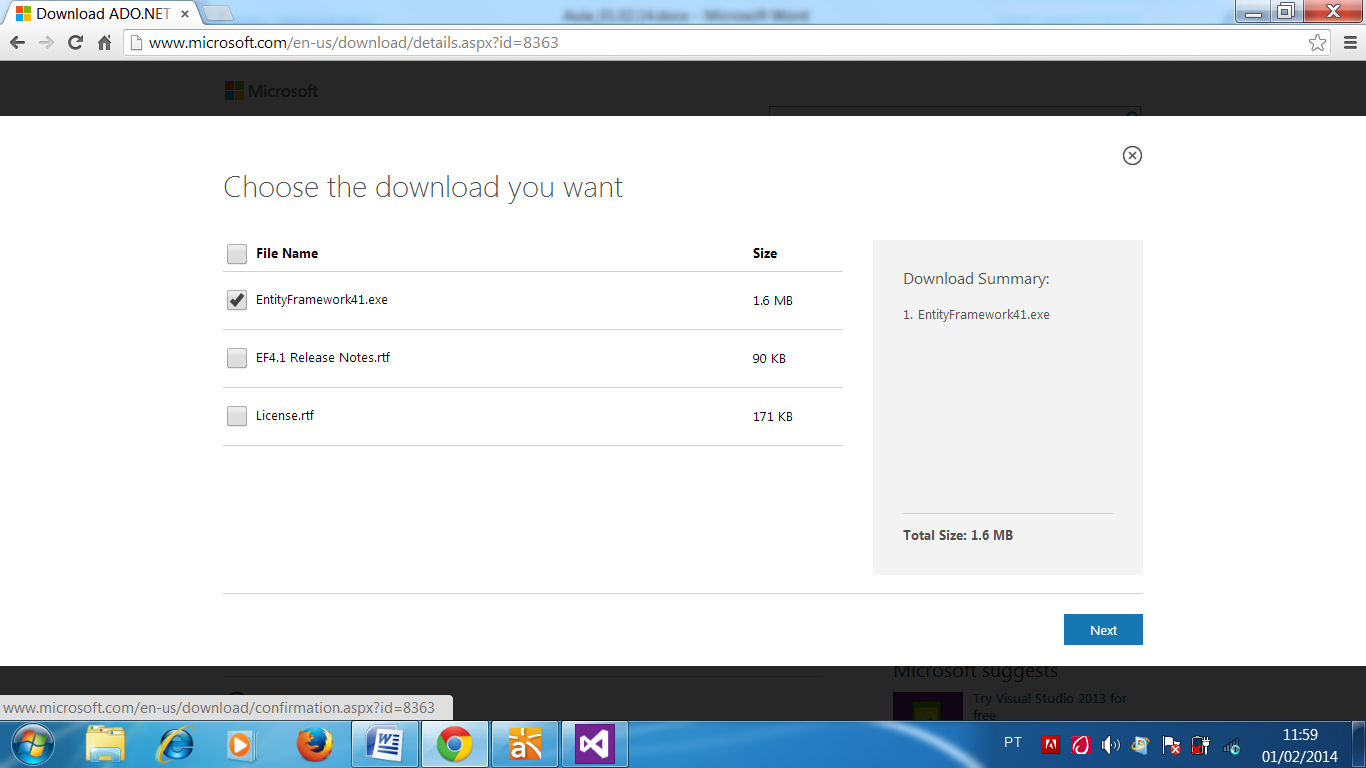
}

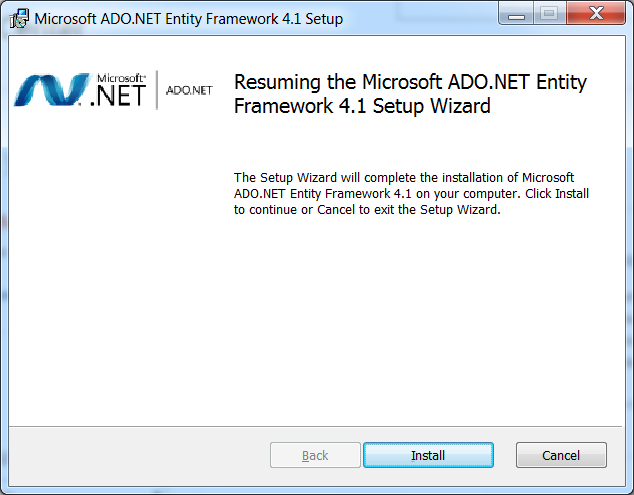
}

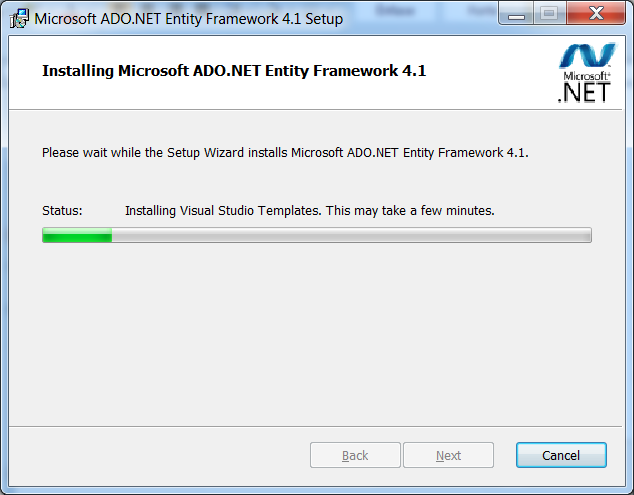
Obtendo o EntityFramework

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

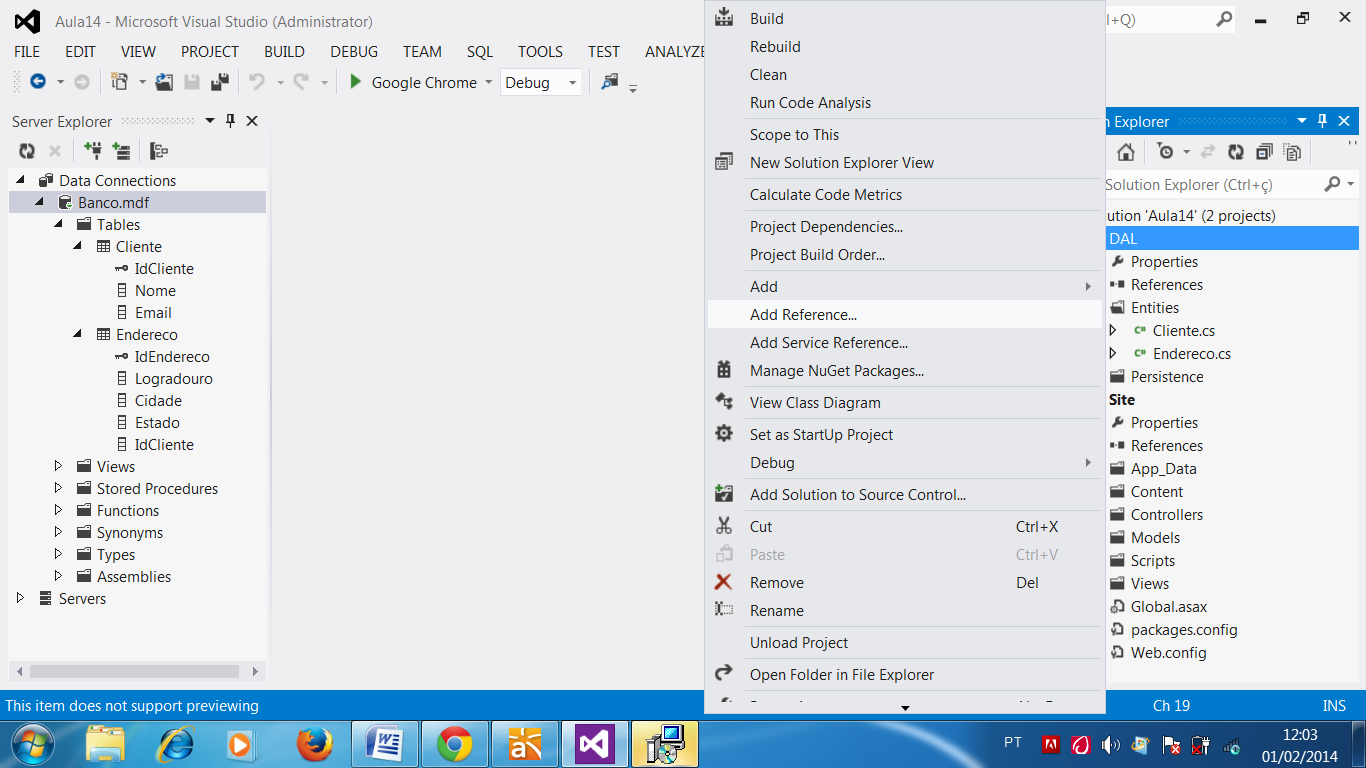


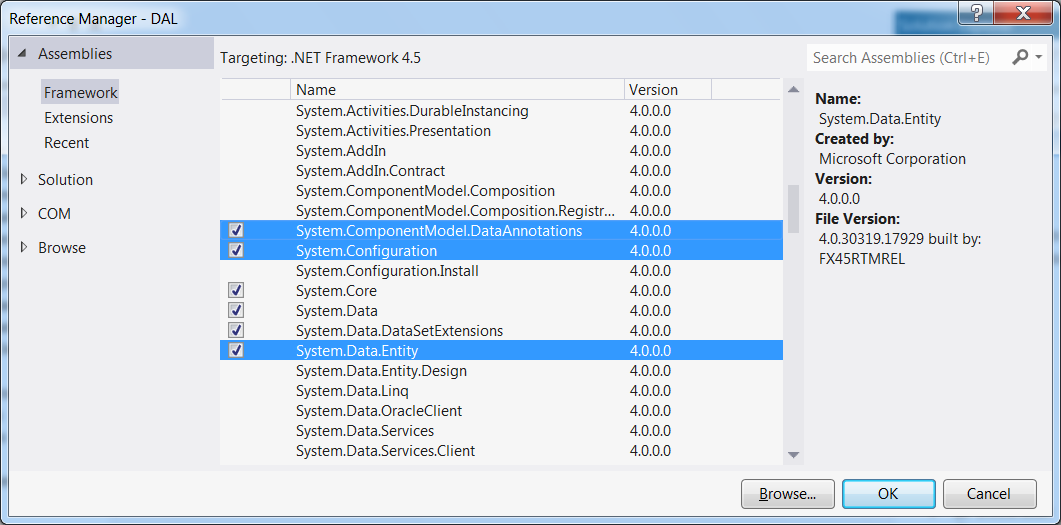


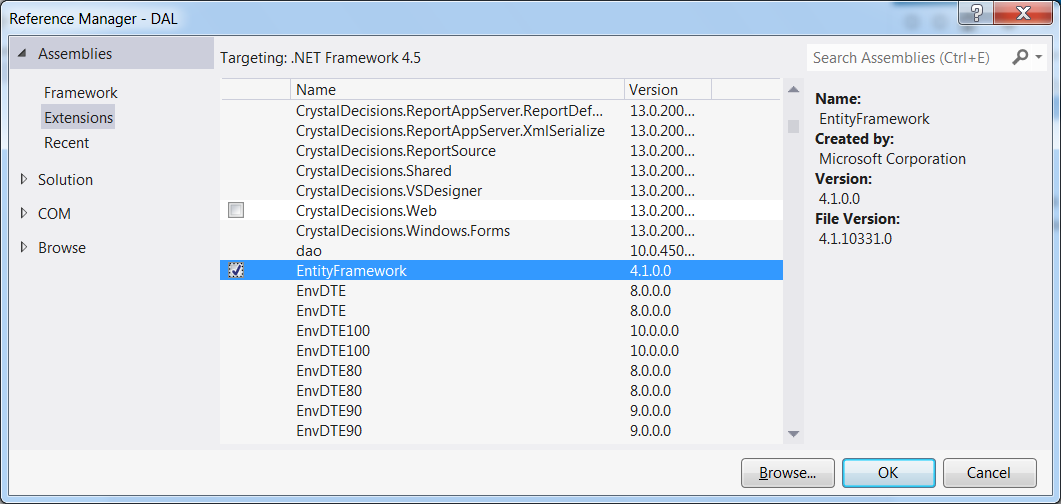




Adicionar referências ao Projeto...







ORM – Mapeamento Objeto Relacional

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.ComponentModel.DataAnnotations; //ORM

namespace DAL.Entities

{

**[Table("Cliente")]**

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; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.ComponentModel.DataAnnotations; //ORM

namespace DAL.Entities

{

**[Table("Endereco")]**

public class Endereco

{

**[Key]**

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

**[Column("IdEndereco")]**

public int IdEndereco { get; set; }

**[Column("Logradouro")]**

public string Logradouro { get; set; }

**[Column("Cidade")]**

public string Cidade { get; set; }

**[Column("Estado")]**

public string Estado { get; set; }

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

public int IdCliente { get; set; }

//Relacionamento (Associação)

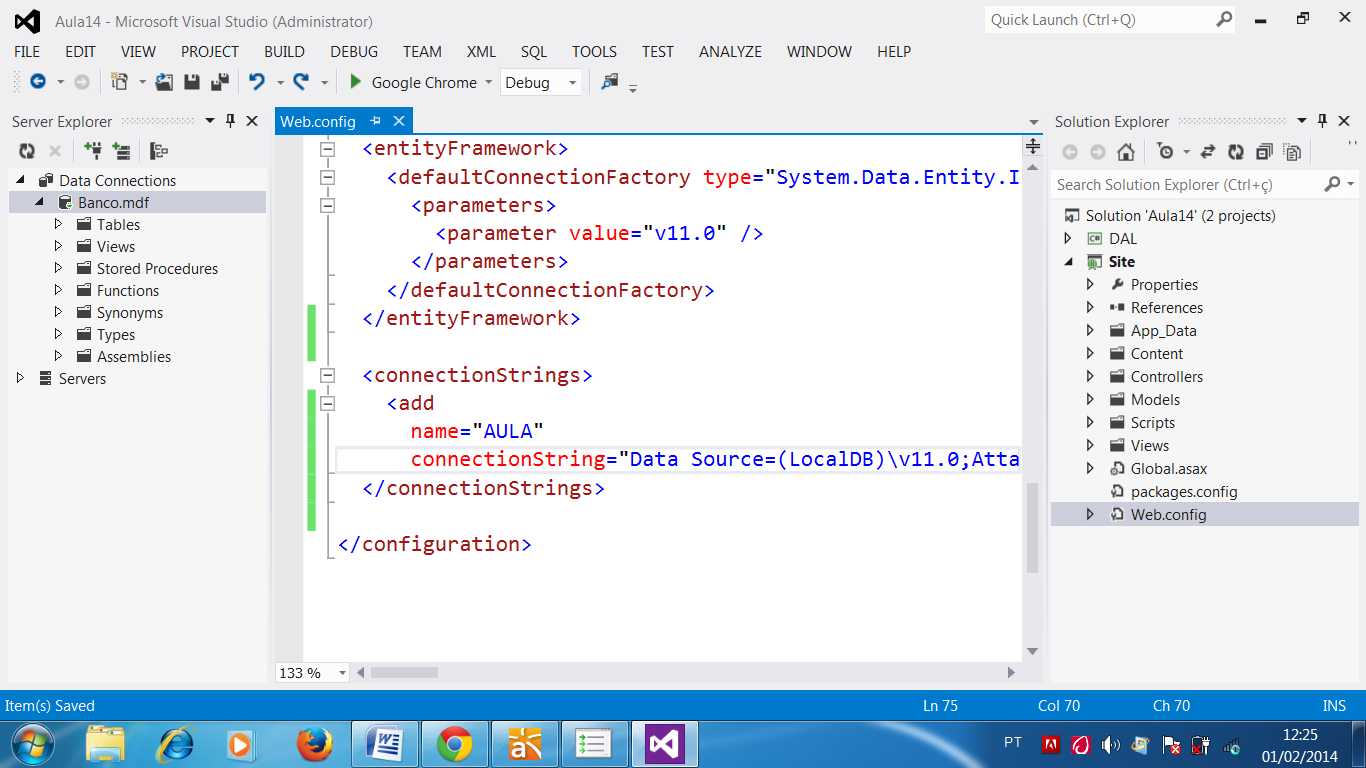
**[ForeignKey("IdCliente")] //nome da propriedade (set/get)**

public Cliente Cliente { get; set; }

}

}

Mapeando a ConnectionString



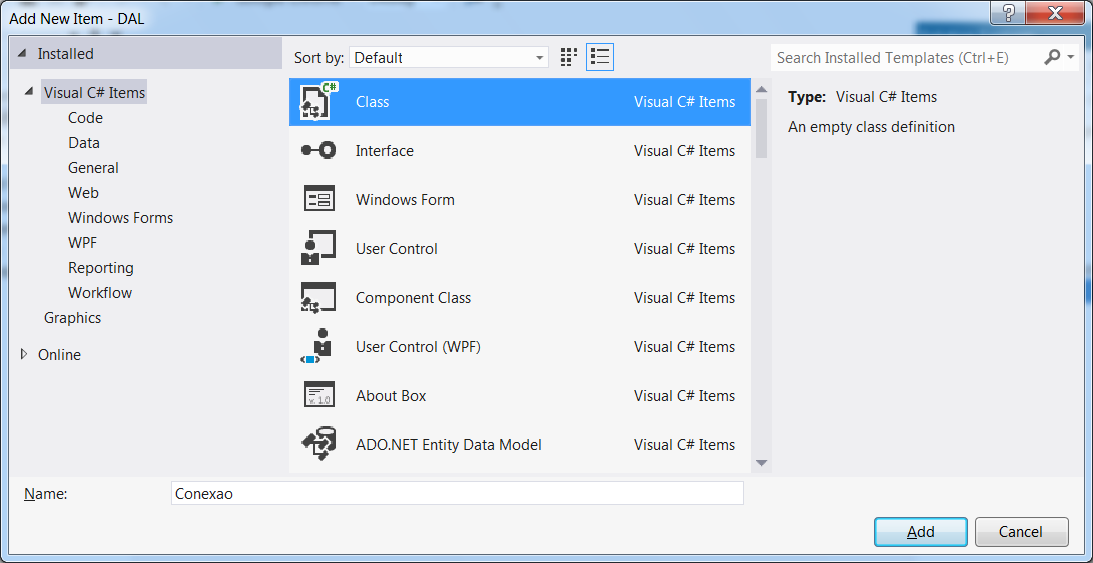
<connectionStrings>

<add

name="AULA"

connectionString="Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Note-HP\Desktop\Aula\_01.02.14\Aula14\Site\App\_Data\Banco.mdf;Integrated Security=True" />

</connectionStrings>



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Configuration; //capturar a connectionstring do web.config.xml

using System.Data.Entity; //entity framework

using DAL.Entities; //classes mapeadas (classes de entidade)

namespace DAL.Persistence

{

//DbContext -> Componente do EF que dará acesso a uma base de dados mapeadas

public class Conexao : DbContext

{

//[ctor] + 2x[tab]

public Conexao()

: base(ConfigurationManager.

ConnectionStrings["AULA"].ConnectionString)

{

//Construtor default

}

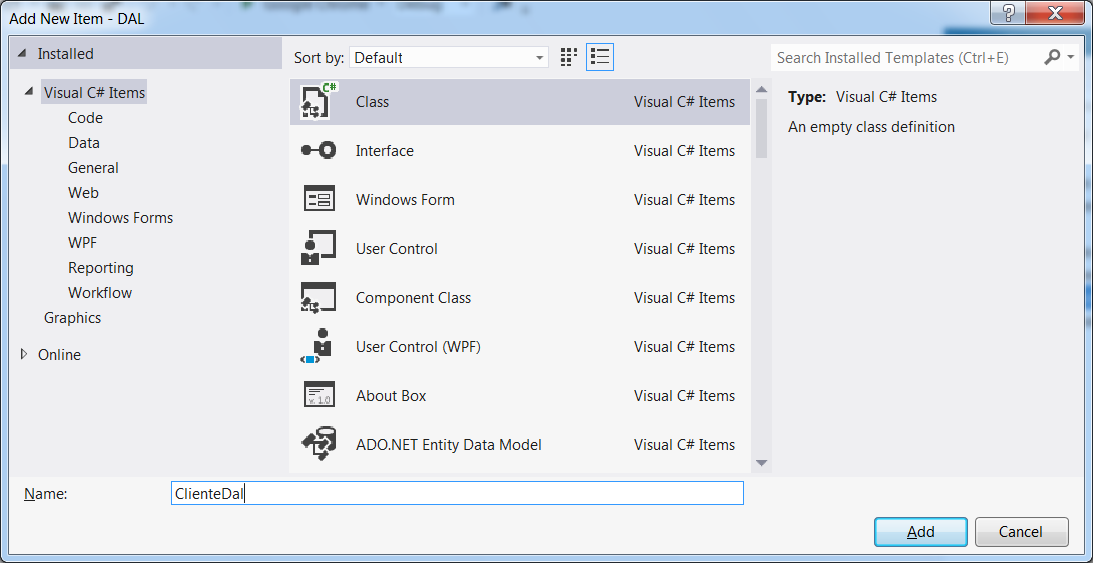
//DbSet -> Componente do EF que referencia as classes mapeadas

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

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

}

}



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DAL.Entities;

namespace DAL.Persistence

{

public class ClienteDal : IDisposable

{

private Conexao Con; //Atributo para a classe de conexão

//[ctor] + 2x[tab]

public ClienteDal()

{

Con = new Conexao(); //inicializando o atributo de conexão

}

//Método para cadastrar Cliente e Endereco

public void Salvar(Cliente c, Endereco e)

{

try

{

Con.Cliente.Add(c); //gravando cliente

Con.Endereco.Add(e); //gravando endereco

Con.SaveChanges();

}

catch(Exception ex)

{

throw new Exception("Erro ao salvar Cliente: " + ex.Message);

}

}

public void Dispose() //Destrutor

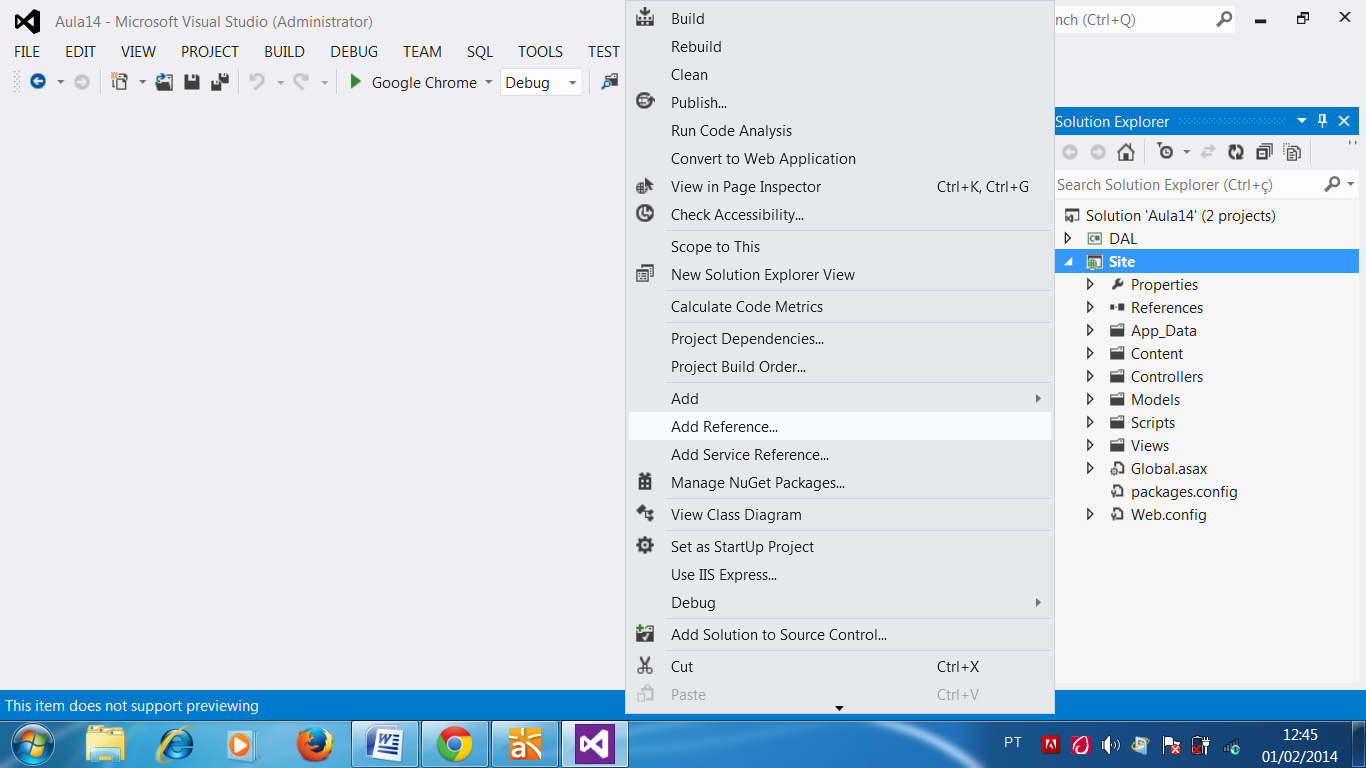
{

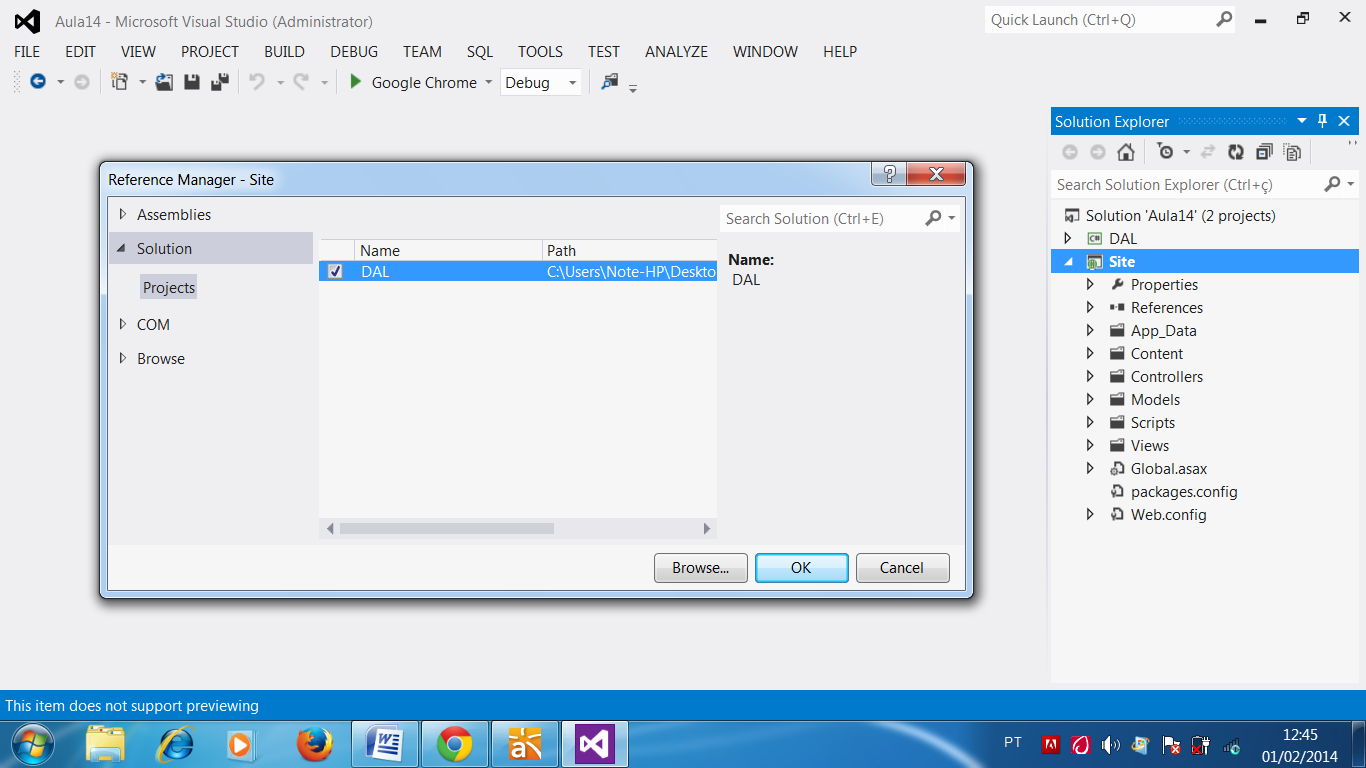
Con.Dispose();

}

}

}





<!-- Vinculando a página à Classe de Modelo -->

@model Site.Models.ClienteModel

<!DOCTYPE html>

<html>

<head>

<title>Cadastro</title>

</head>

<body>

<div>

<h3>Cadastro de Clientes</h3>

<hr />

<!-- Formulário padrao Razor (MVC) -->

@using (Html.BeginForm("CadastrarCliente", "Cliente", FormMethod.Post))

{

<div>

<!-- Mensagens de erro de validação -->

@Html.ValidationSummary()

</div>

<div>

<!-- Mensagem do controle -->

@ViewBag.Mensagem

</div>

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

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

<br /><br />

@Html.LabelFor(model => model.Email) <br />

@Html.TextBoxFor(model => model.Email)

<br /><br />

@Html.LabelFor(model => model.Logradouro) <br />

@Html.TextBoxFor(model => model.Logradouro)

<br /><br />

@Html.LabelFor(model => model.Cidade) <br />

@Html.TextBoxFor(model => model.Cidade)

<br /><br />

@Html.LabelFor(model => model.Estado) <br />

@Html.TextBoxFor(model => model.Estado)

<br /><br />

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

}

</div>

</body>

</html>

