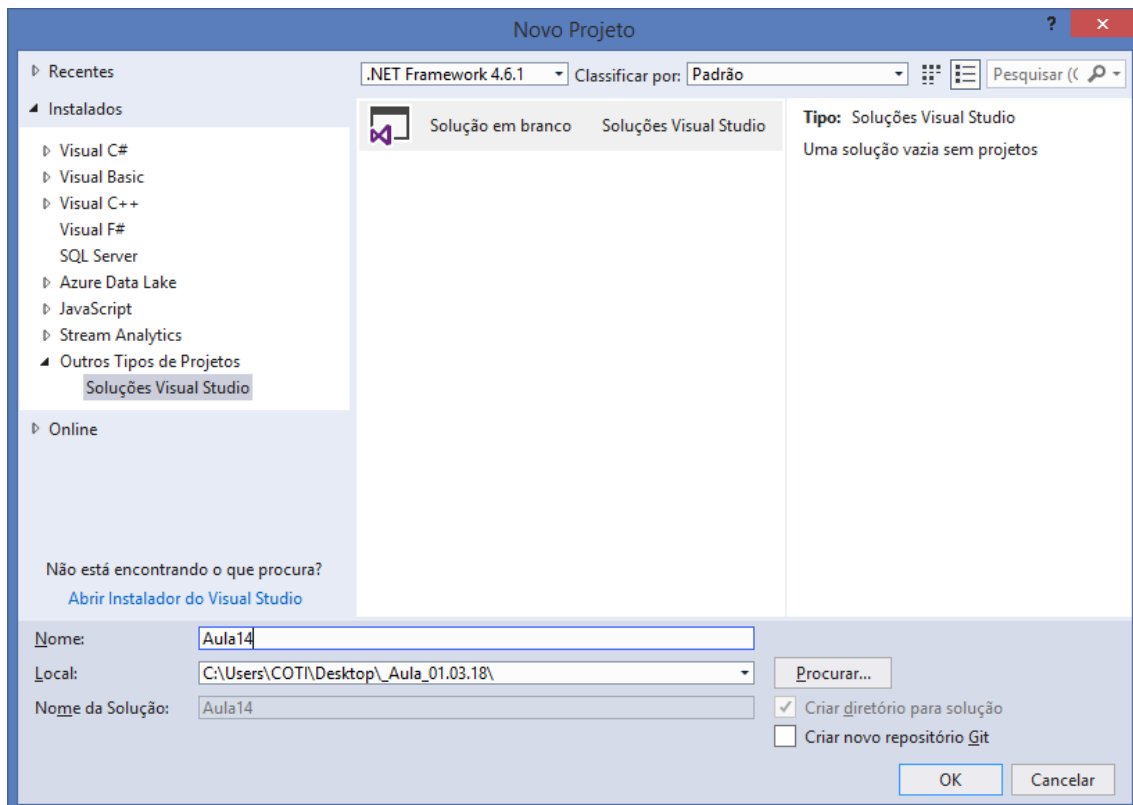
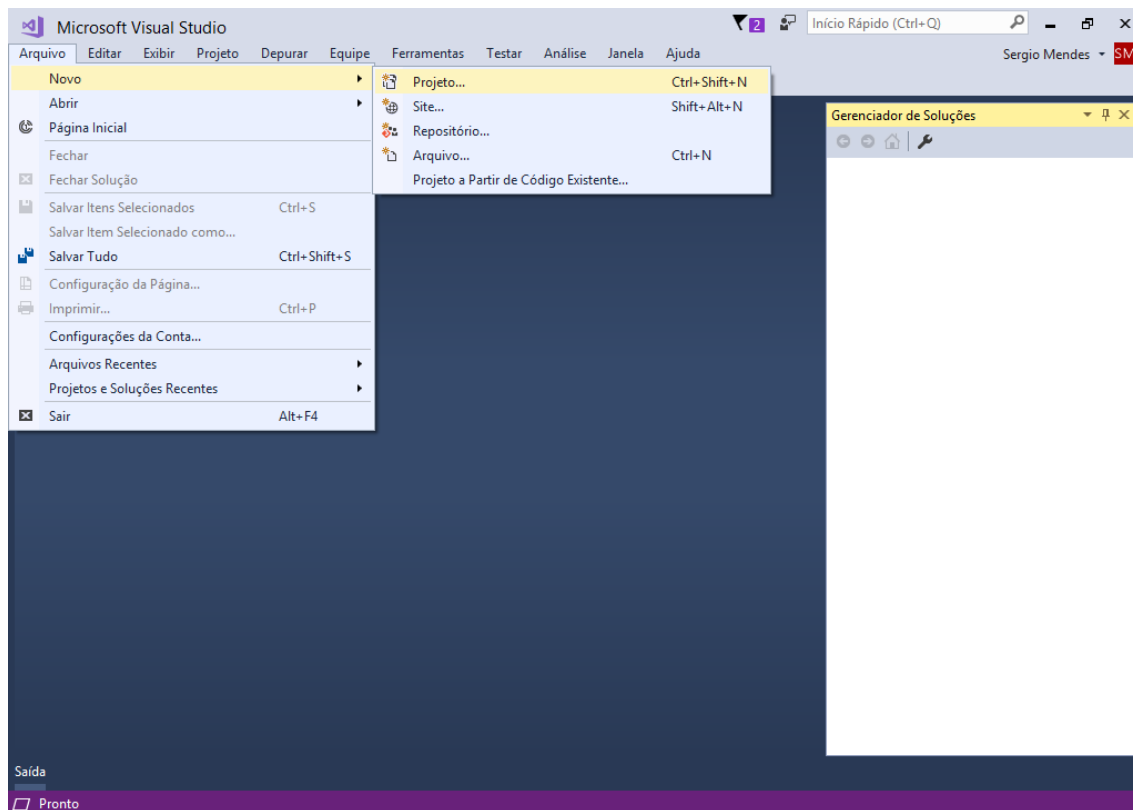
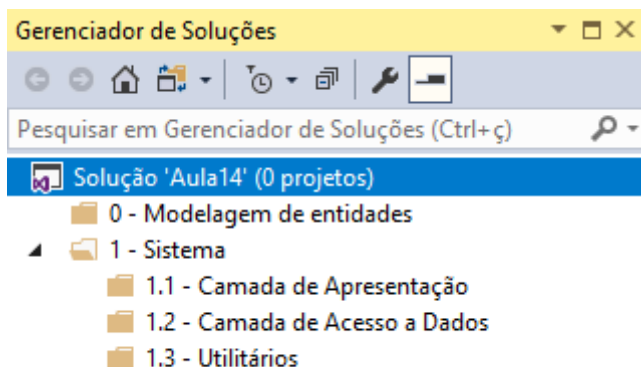


## Nova solução em branco:

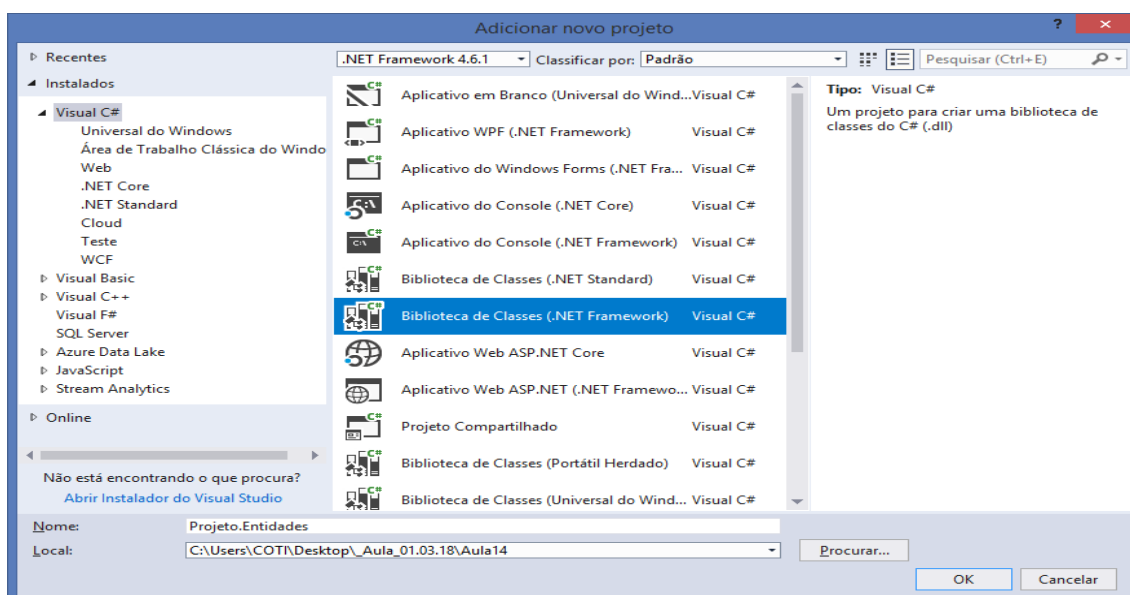


## Organização da Solution



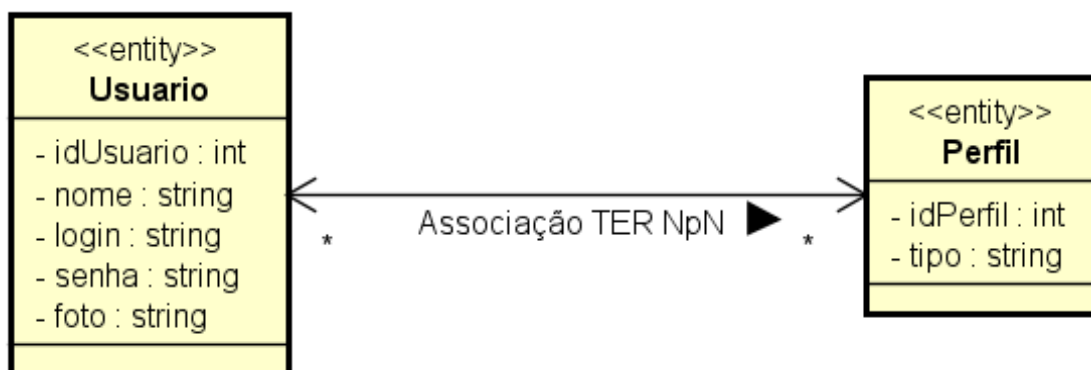
## 0 - Modelagem de entidades

Biblioteca de Classes (.NET Framework)



## Diagrama de Classes

Modelagem de Associação muitos para muitos



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Projeto.Entidades
{
    public class Usuario
    {
        public int IdUsuario { get; set; }
        public string Nome { get; set; }
        public string Login { get; set; }
        public string Senha { get; set; }
        public string Foto { get; set; }
        public DateTime DataCadastro { get; set; }

        //Associação (TER-MUITOS)
        public List<Perfil> Perfis { get; set; }

        public Usuario()
        {
        }

        public Usuario(int idUsuario, string nome, string login,
            string senha, string foto, DateTime dataCadastro)
        {
            IdUsuario = idUsuario;
            Nome = nome;
            Login = login;
            Senha = senha;
            Foto = foto;
            DataCadastro = dataCadastro;
        }

        public override string ToString()
        {
            return $"Id do Usuario: {IdUsuario}, Nome: {Nome},
                Login: {Login}, Data de cadastr: {DataCadastro}";
        }
    }
}
```

```
-----

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Projeto.Entidades
{
    public class Perfil
    {
        //propriedades
        public int IdPerfil { get; set; }
        public string Tipo { get; set; }
    }
}
```

```
//Relacionamento de Associação (TER-MUITOS)
public List<Usuario> Usuario { get; set; }

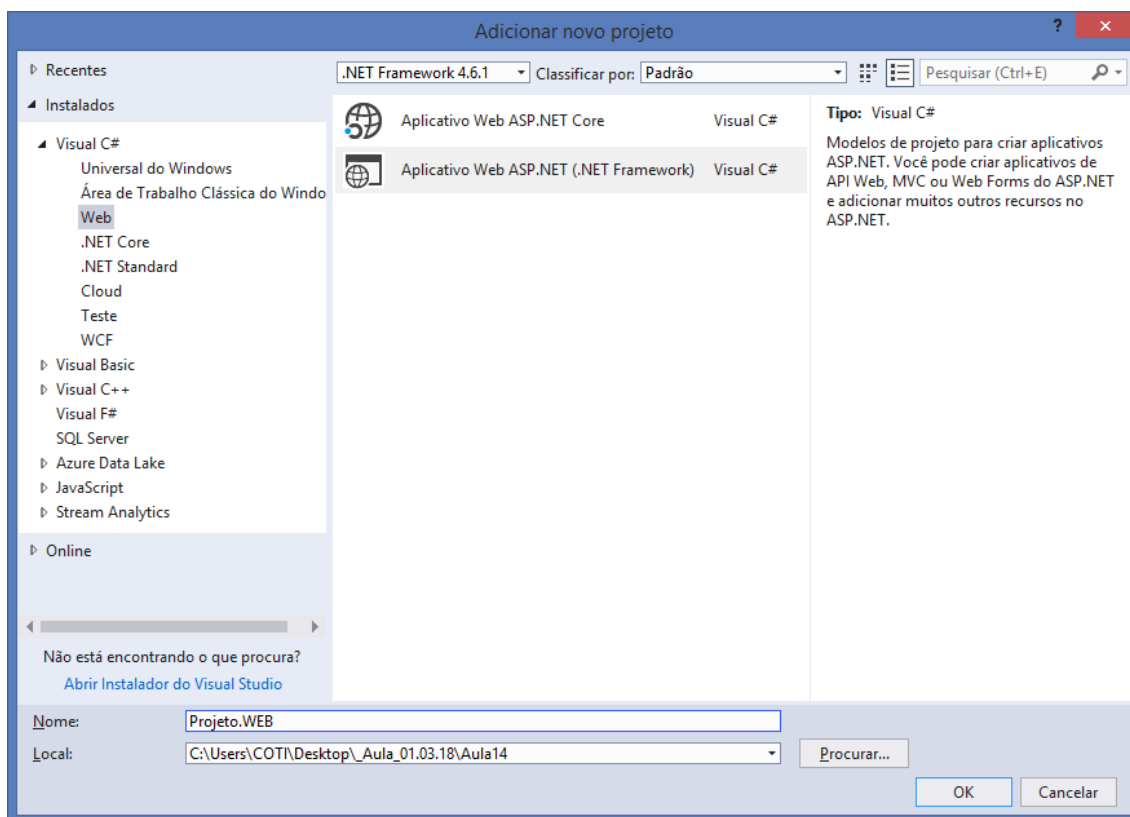
//construtor default..
public Perfil()
{
}

//sobrecarga de construtores
public Perfil(int idPerfil, string tipo)
{
    IdPerfil = idPerfil;
    Tipo = tipo;
}

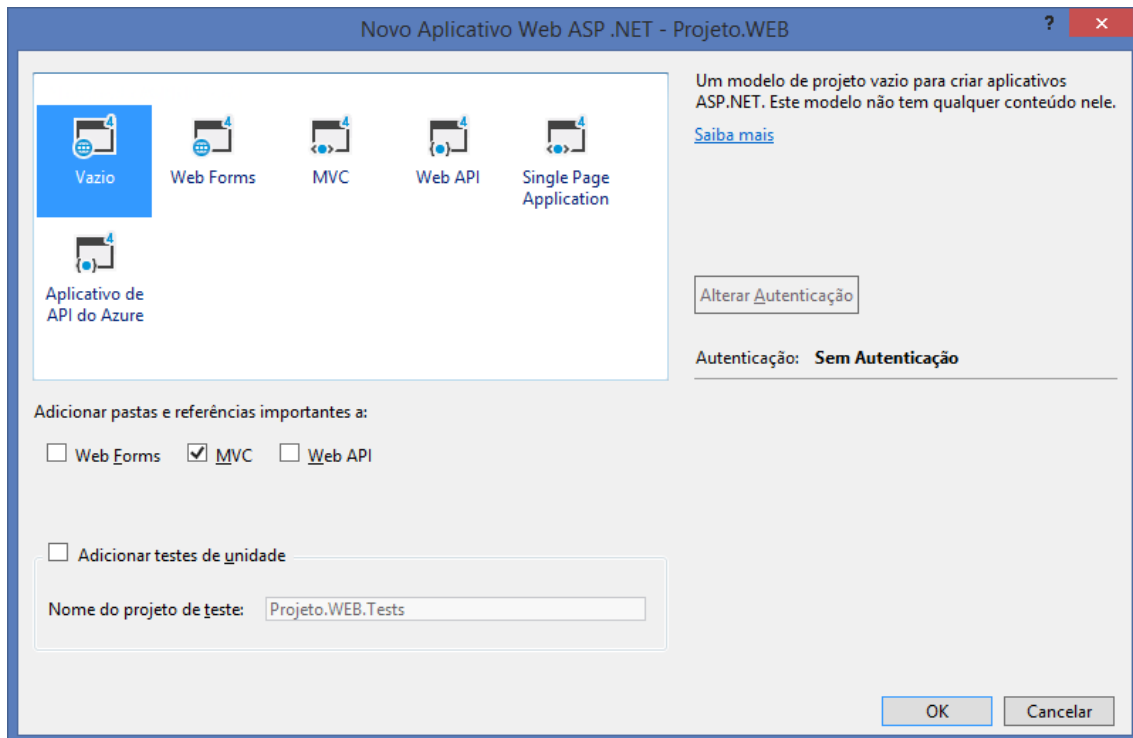
//sobrescrita do método toString..
public override string ToString()
{
    return $"Id do Perfil: {IdPerfil}, Tipo: {Tipo}";
}
}
}
```

## 1.1 - Camada de Apresentação

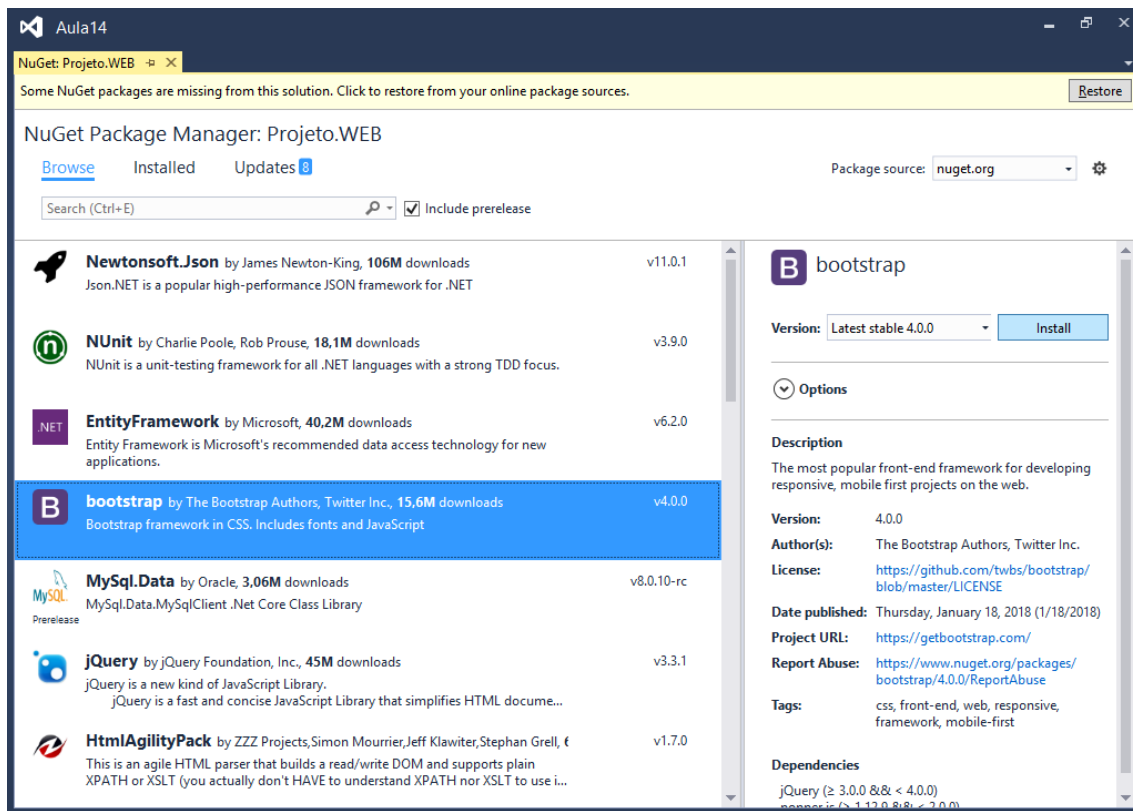
### Asp.Net MVC WebApplication



## Selecione empty / mvc

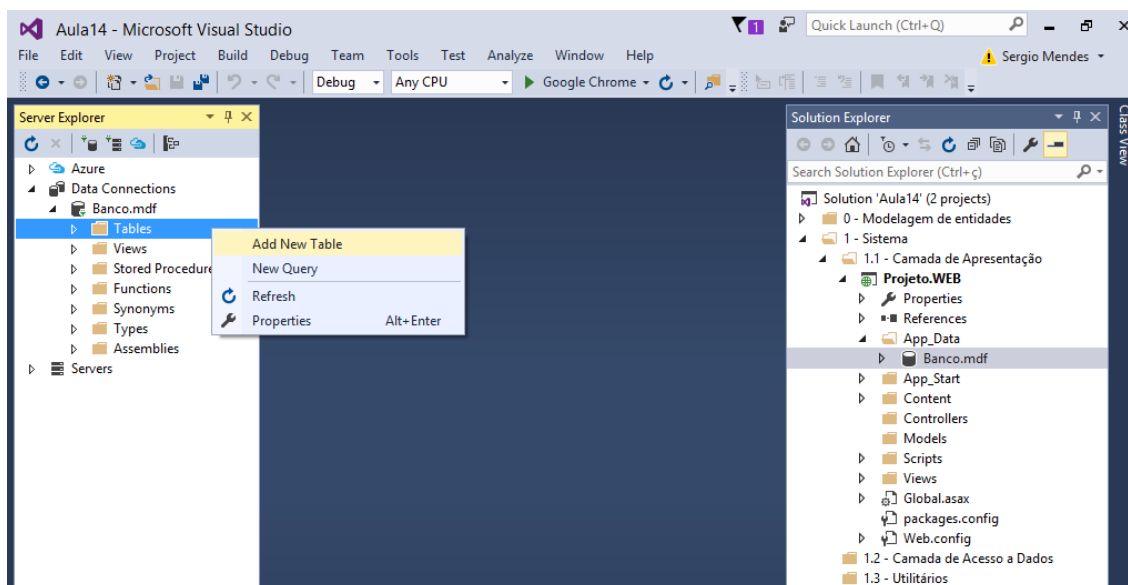
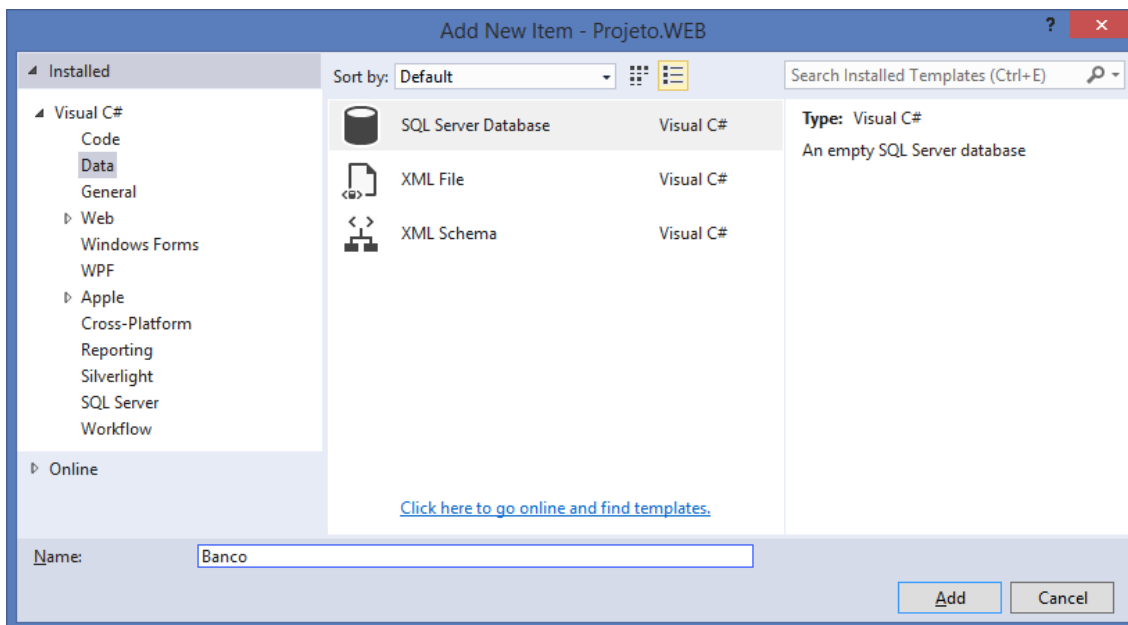


## Instalando o bootstrap: Gerenciador de pacotes do nuget



## Criando a base de dados

### MDF - Master Database File

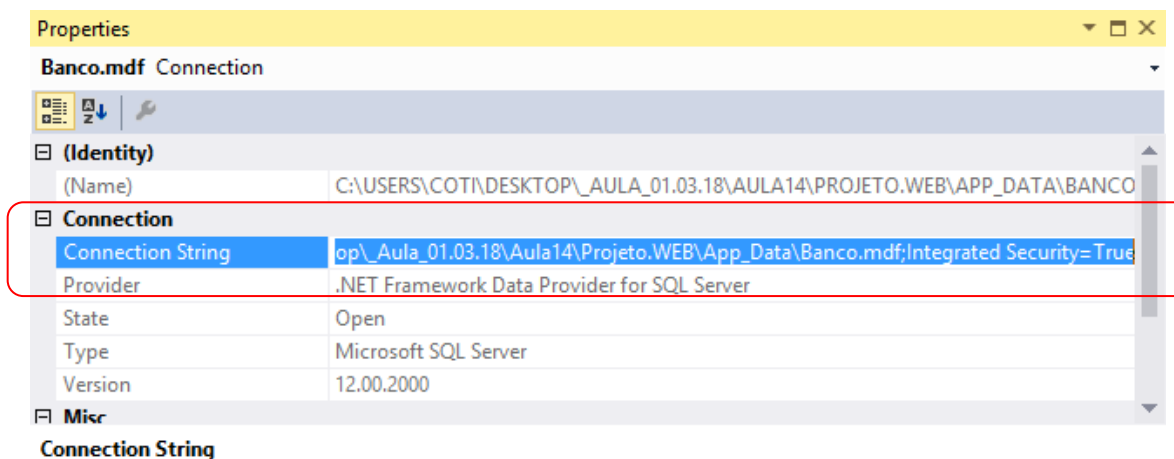
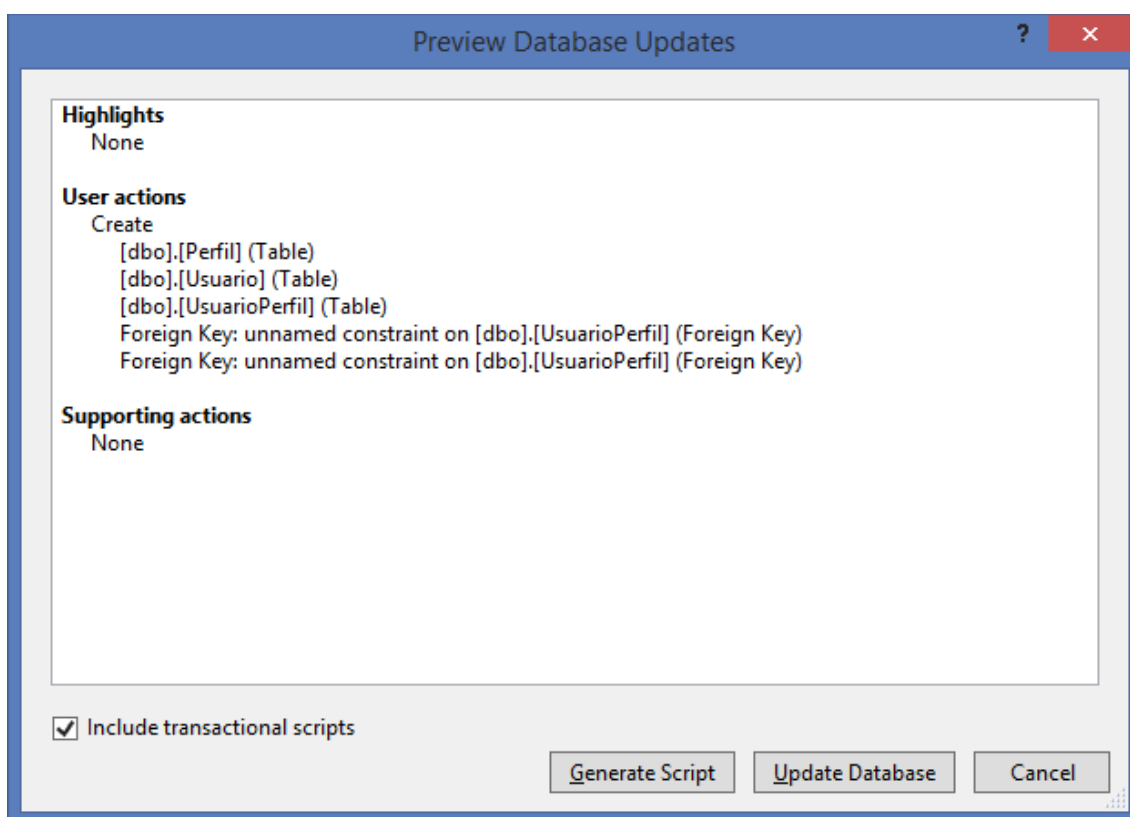


```
create table Perfil(
    IdPerfil      integer                identity(1,1),
    Tipo          nvarchar(50) not null,
    primary key(IdPerfil))
go

create table Usuario(
    IdUsuario     integer                identity(1,1),
    Nome          nvarchar(50) not null,
    [Login]       nvarchar(25) not null unique,
    Senha        nvarchar(50) not null,
    Foto         nvarchar(255) not null,
    primary key (IdUsuario));
go
```

```
--entidade associativa
create table UsuarioPerfil(
    IdUsuario    integer                not null,
    IdPerfil     integer                not null,
    primary key (IdUsuario, IdPerfil),
    foreign key (IdUsuario) references Usuario(IdUsuario),
    foreign key (IdPerfil) references Perfil(IdPerfil))
go
```

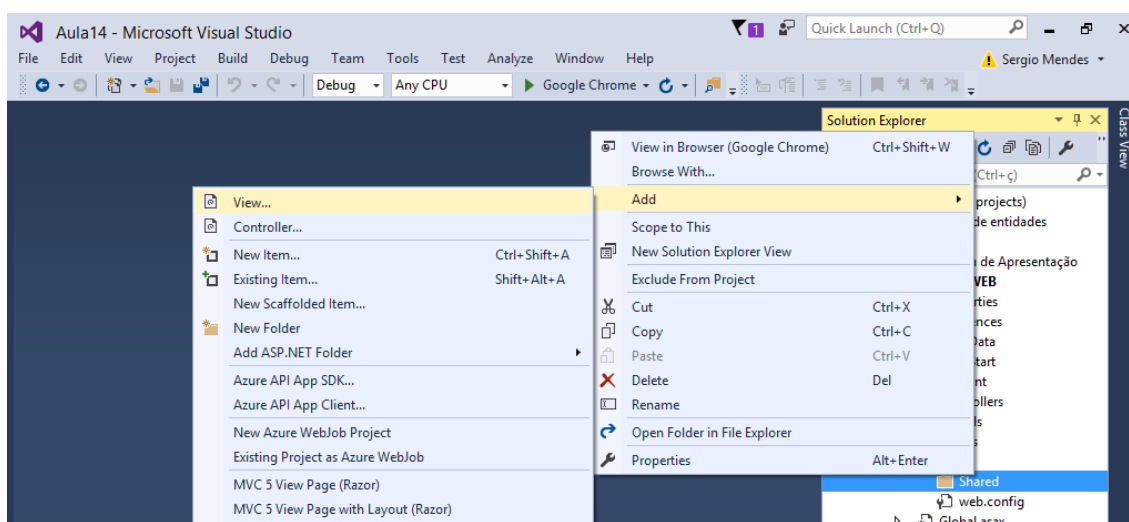
## Executando:



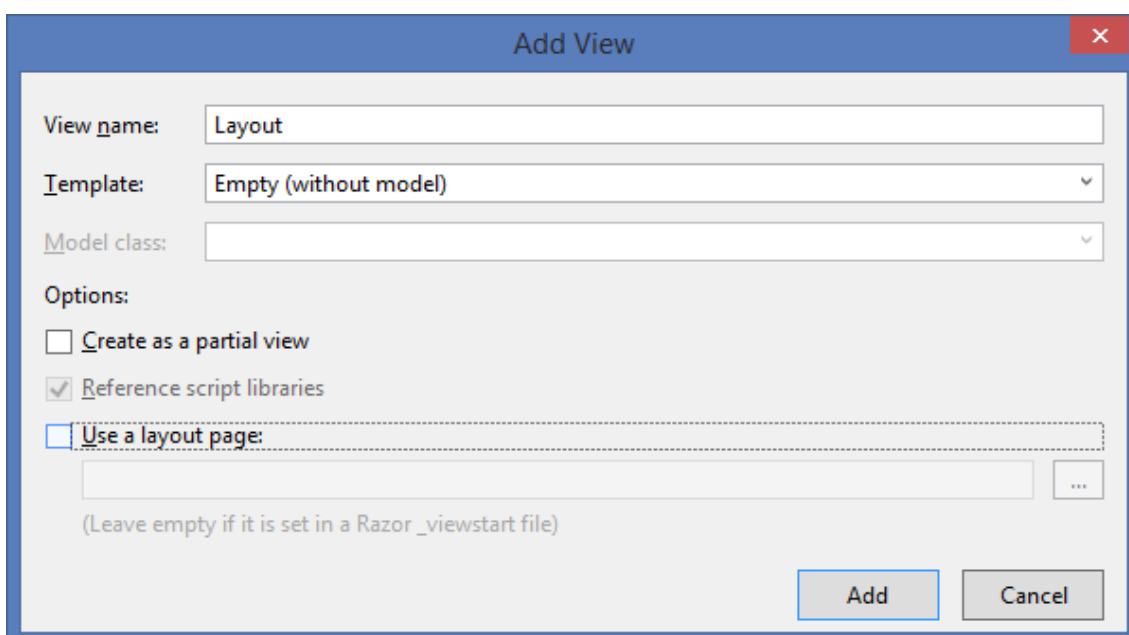
## \Web.config.xml

```
<connectionStrings>
  <add
    name="aula"
    connectionString="Data Source=(LocalDB)\MSSQLLocalDB;
    AttachDbFilename=C:\Users\COTI\Desktop\_Aula_01.03.18\
    Aula14\Projeto.WEB\App_Data\Banco.mdf;Integrated Security=True"
  />
</connectionStrings>
```

Criando uma página de layout (/Views/)



Criando uma página de layout:





```
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>COTI Informatica</title>

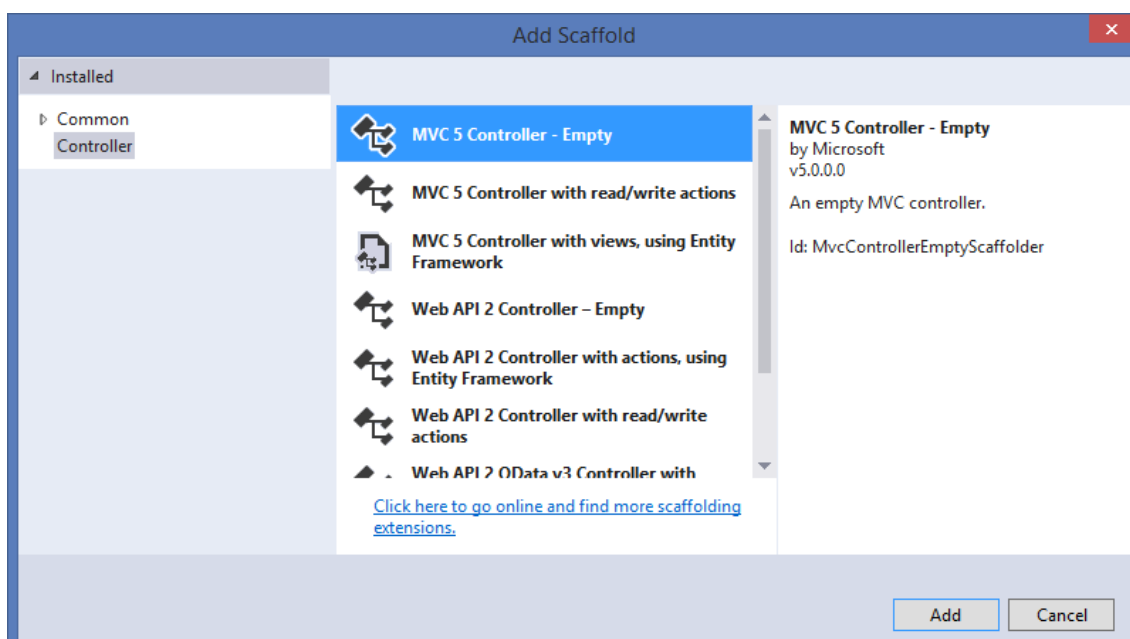
    <!-- folhas de estilo CSS do bootstrap -->
    <link href="~/Content/bootstrap.min.css" rel="stylesheet" />

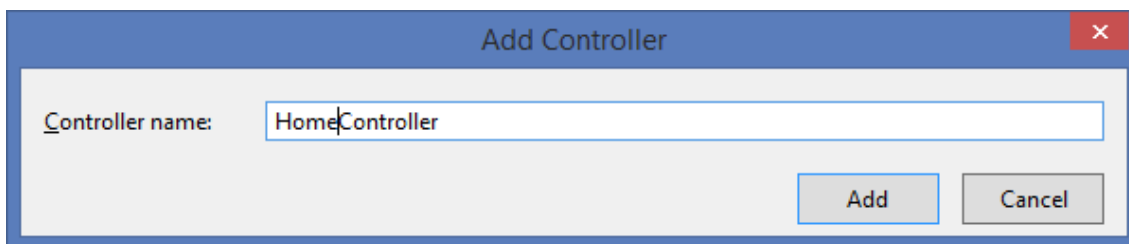
</head>
<body class="container">

    <div class="row">
        <div class="col-md-12">
            <h2>Sistema de Controle de Clientes</h2>
            REALIZE SEU LOGIN OU CADASTRE-SE
        </div>
    </div>

    <div class="row">
        <div class="col-md-12">
            <!-- Local para entrada do conteudo das demais páginas -->
            @RenderBody()
        </div>
    </div>

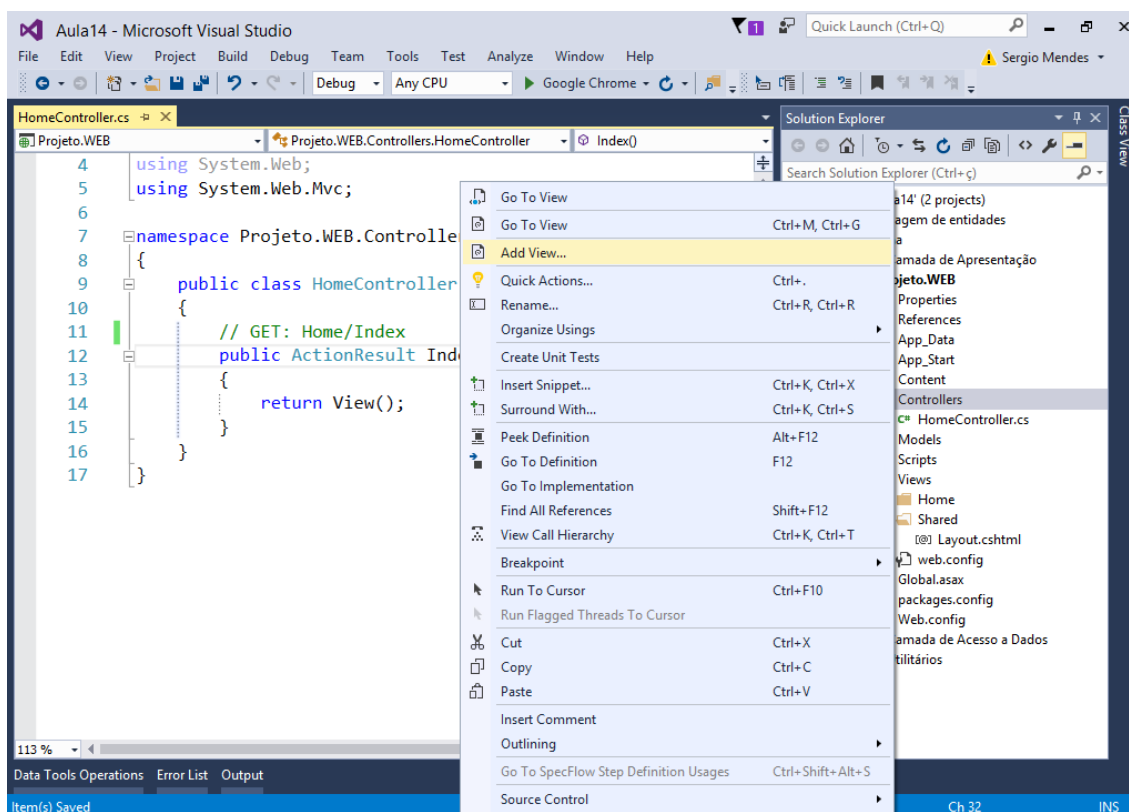
</body>
</html>
```

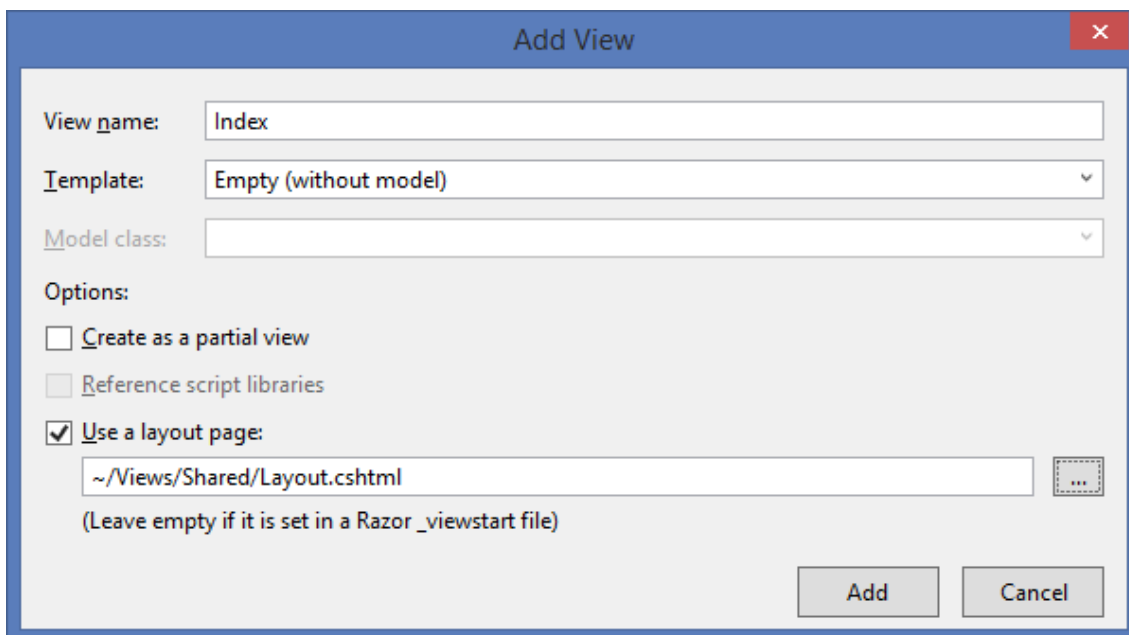




```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Projeto.WEB.Controllers
{
    public class HomeController : Controller
    {
        // GET: Home/Index
        public ActionResult Index()
        {
            return View();
        }
    }
}
```





**Add View**

View name:

Template:

Model class:

Options:

☐ Create as a partial view

☐ Reference script libraries

☒ Use a layout page:

(Leave empty if it is set in a Razor \_viewstart file)

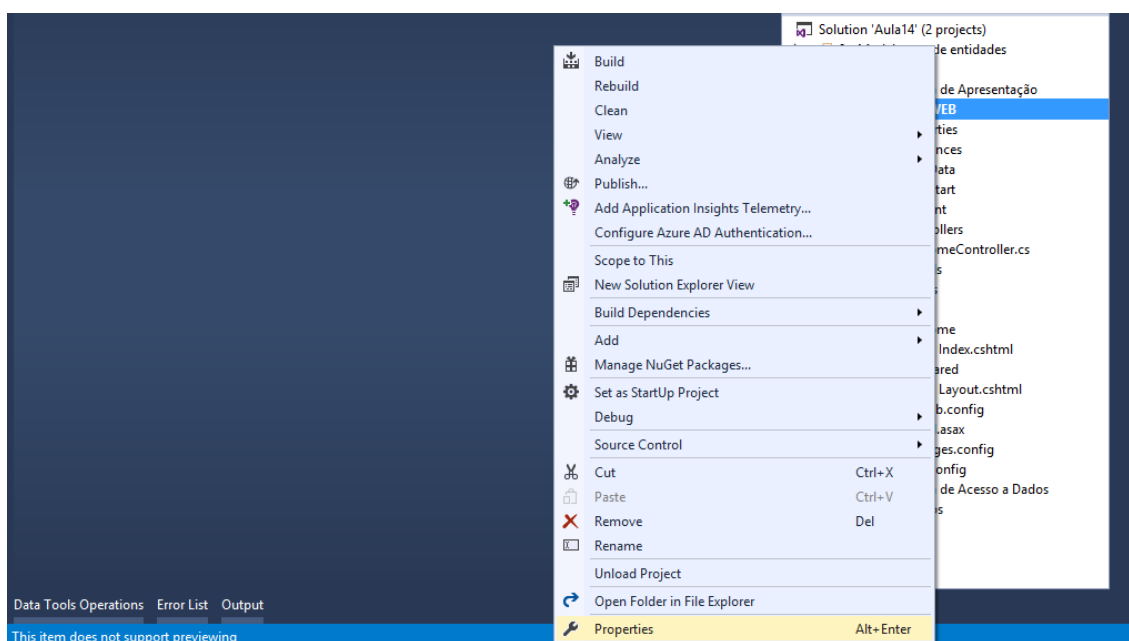
```
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/Layout.cshtml";
}
```

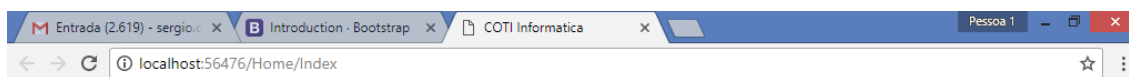
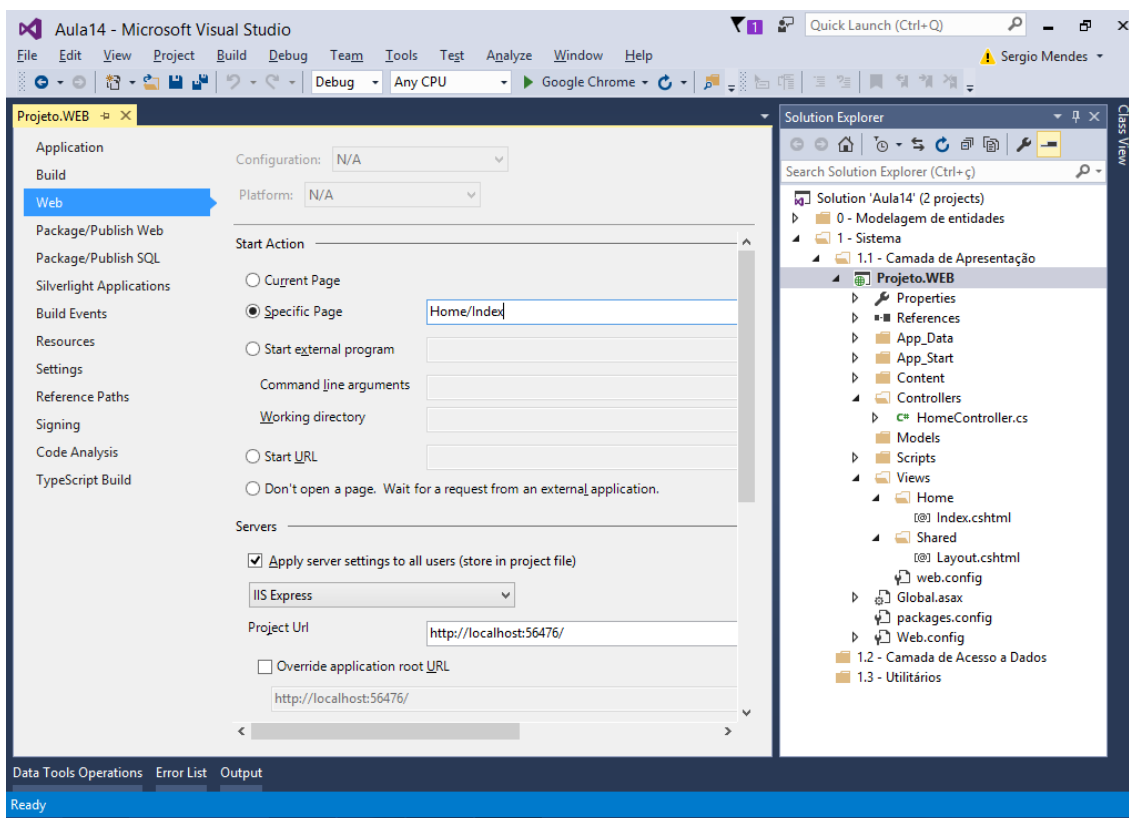
```
<h4>Bem vindo ao projeto</h4>
<hr/>
```

Selecione:

```
<hr />

<ul>
    <li> <a href="/Usuario/Login">Realizar Autenticação</a> </li>
    <li> <a href="/Usuario/Cadastro">Cadastrar conta de Usuário</a> </li>
</ul>
```





## Sistema de Controle de Clientes

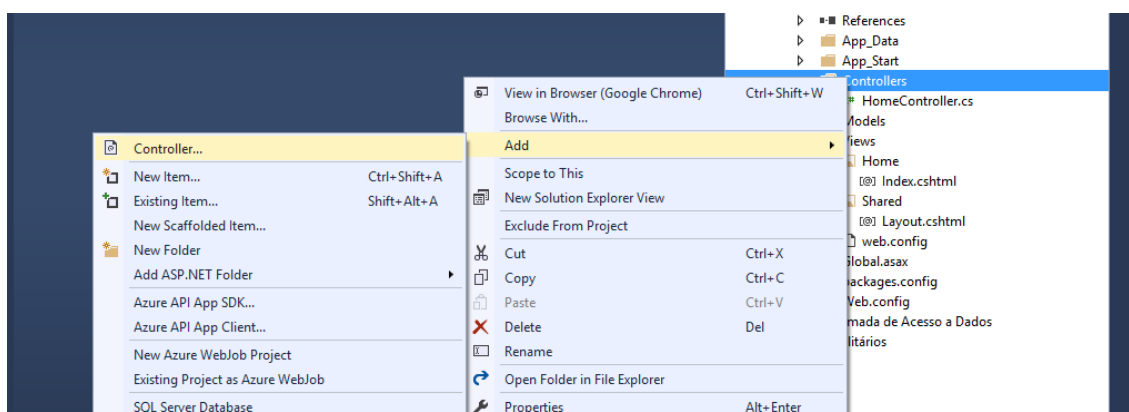
REALIZE SEU LOGIN OU CADASTRE-SE

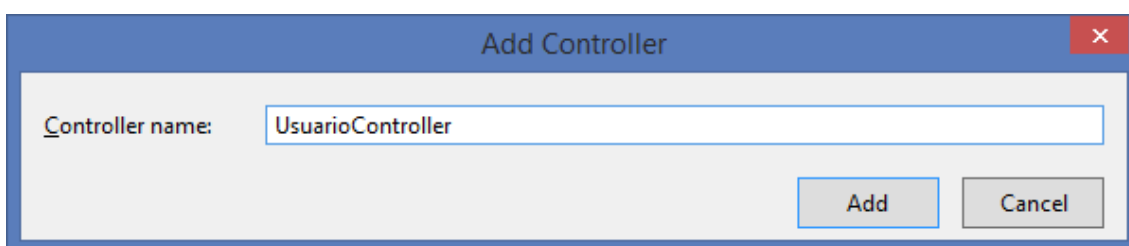
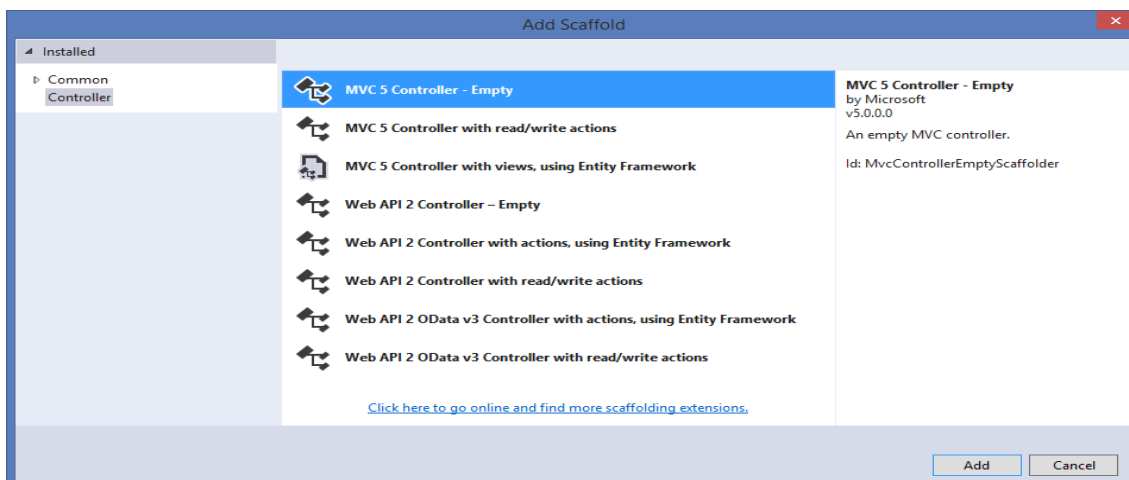
Bem vindo ao projeto

Selecione:

- [Realizar Autenticação](#)
- [Cadastrar conta de Usuário](#)

## Criaremos mais 2 classes de controle além da HomeController



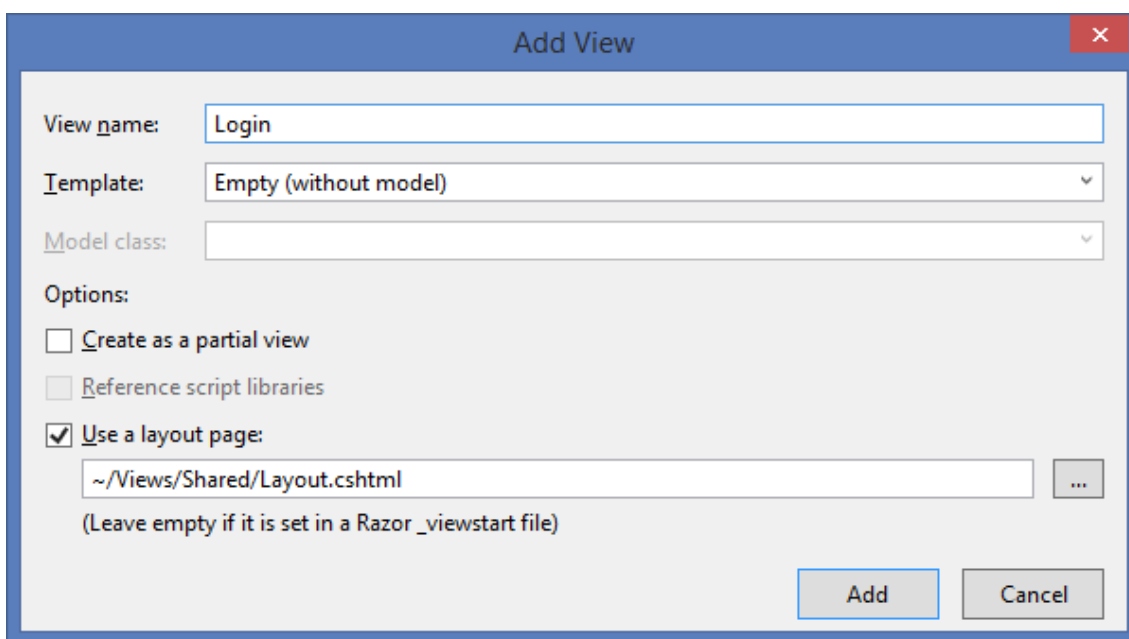
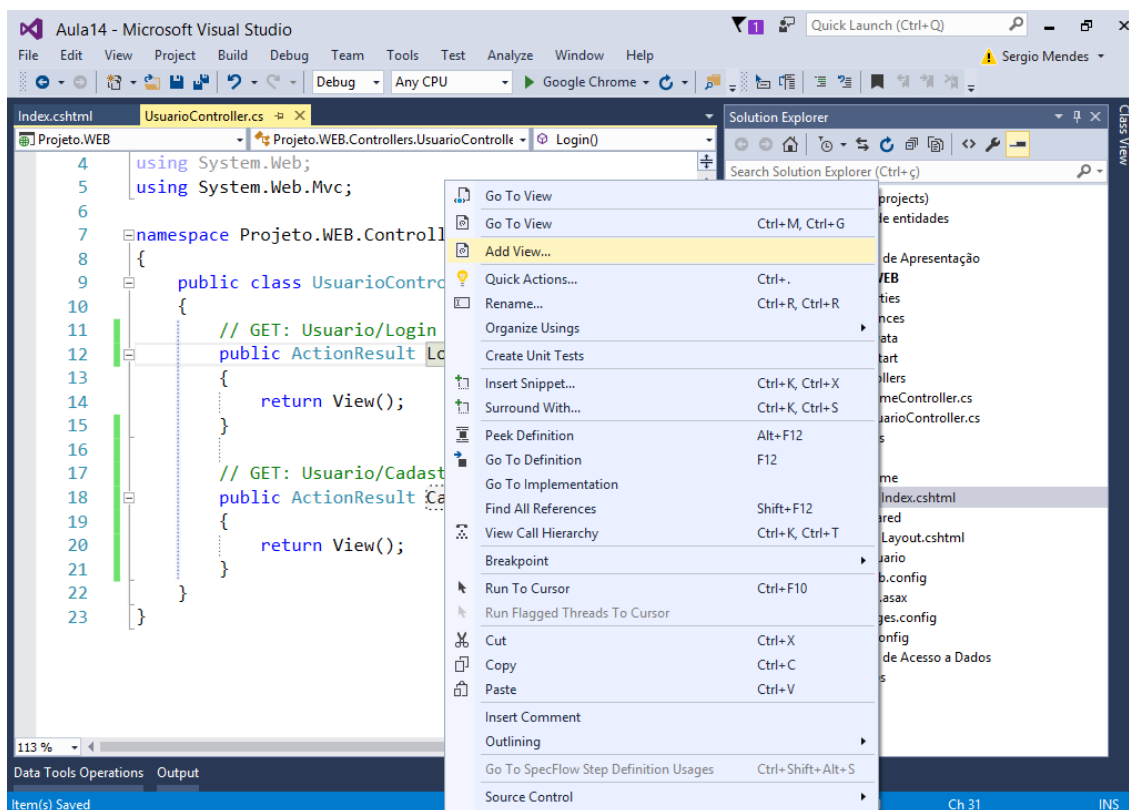


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Projeto.WEB.Controllers
{
    public class UsuarioController : Controller
    {
        // GET: Usuario/Login
        public ActionResult Login()
        {
            return View();
        }

        // GET: Usuario/Cadastro
        public ActionResult Cadastro()
        {
            return View();
        }
    }
}
```

## Criando a página de login:



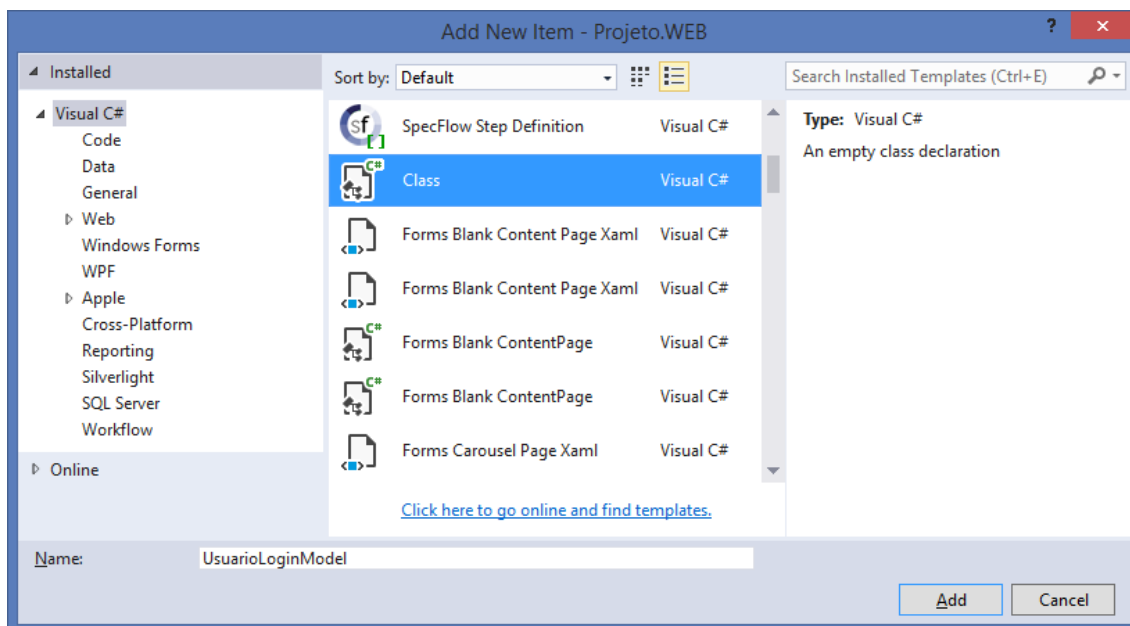
```
@{
    ViewBag.Title = "Login";
    Layout = "~/Views/Shared/Layout.cshtml";
}
```

```
<h4>Login de Usuários</h4>
<a href="/Home/Index">Voltar</a>
<hr/>

<div class="row">
    <div class="col-md-3">

        </div>
    </div>
</div>
```

**Criando a classe de modelo para construirmos o formulario de login de usuario:**



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations; //mapeamentos..

namespace Projeto.WEB.Models
{
    public class UsuarioLoginModel
    {
        [Required(ErrorMessage = "Informe seu login.")]
        public string Login { get; set; }

        [Required(ErrorMessage = "Informe sua senha.")]
        public string Senha { get; set; }
    }
}
```

```
@{
    ViewBag.Title = "Login";
    Layout = "~/Views/Shared/Layout.cshtml";
}

<h4>Login de Usuários</h4>
<a href="/Home/Index">Voltar</a>
<hr/>

<div class="row">
    <div class="col-md-3">

        <!-- Declarando a classe de modelo.. -->
        @model Projeto.WEB.Models.UsuarioLoginModel

        <!-- Abrindo a área do formulário -->
        @using (Html.BeginForm())
        {
            <label>Login de Acesso:</label>
            @Html.TextBoxFor(model => model.Login)
            <br/>

            <label>Senha de Acesso:</label>
            @Html.PasswordFor(model => model.Senha)
            <br />

            <input type="submit" value="Acessar Sistema"/>
        }

    </div>
</div>
```

## Executando:

```
@{
    ViewBag.Title = "Login";
    Layout = "~/Views/Shared/Layout.cshtml";
}

<h4>Login de Usuários</h4>
<a href="/Home/Index">Voltar</a>
<hr />

<div class="row">
    <div class="col-md-3">

        <!-- Declarando a classe de modelo.. -->
        @model Projeto.WEB.Models.UsuarioLoginModel

        <!-- Abrindo a área do formulário -->
        @using (Html.BeginForm())
        {
            <label>Login de Acesso:</label>
            @Html.TextBoxFor(model => model.Login,
                new
                {
                    @class = "form-control",
                    @placeholder = "Login de Acesso"
                })
            <br />
        }
    </div>
</div>
```



```
<label>Senha de Acesso:</label>
@Html.PasswordFor(model => model.Senha,
    new
    {
        @class = "form-control",
        @placeholder = "Sena de Acesso"
    });
<br />

<input type="submit" value="Acessar Sistema"
    class="btn btn-success" />
}

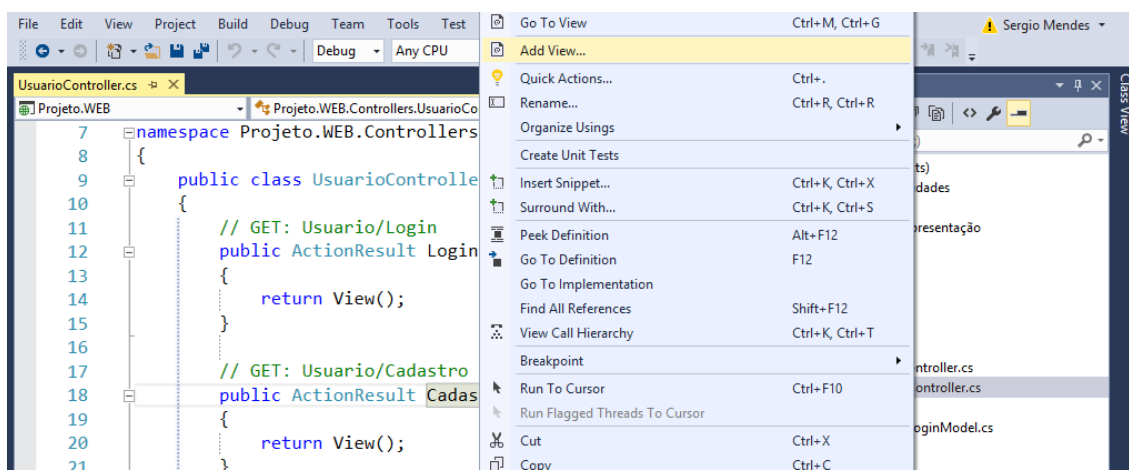
</div>
</div>
```

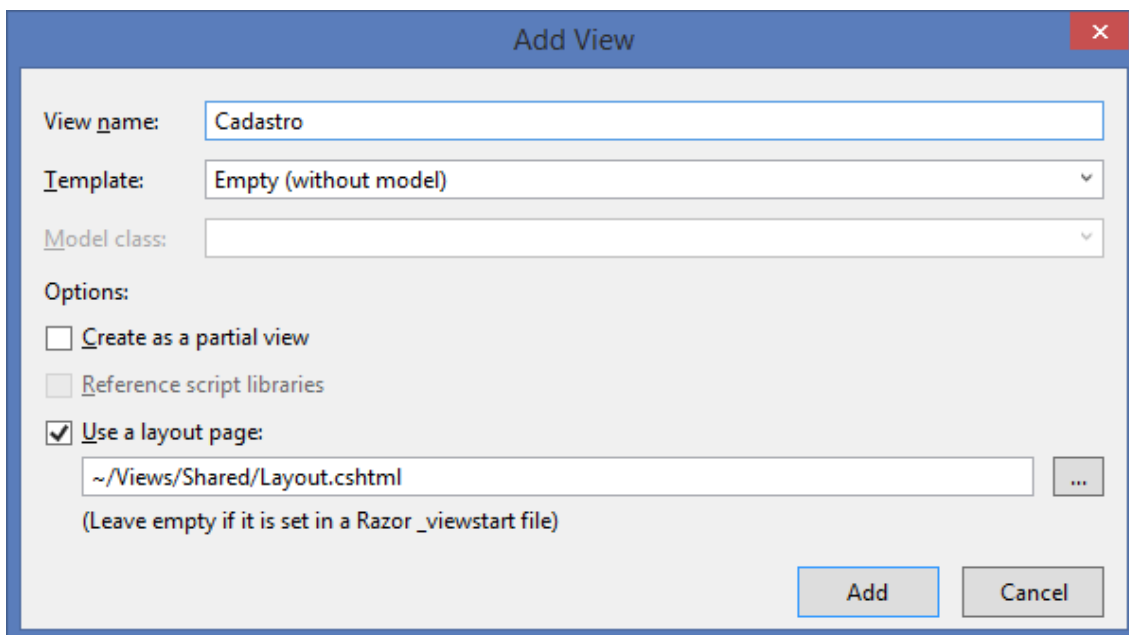
## Executando:



The screenshot shows a web browser window with the URL `localhost:56476/Usuario/Login`. The page title is "Sistema de Controle de Clientes". Below the title, it says "REALIZE SEU LOGIN OU CADASTRE-SE" and "Login de Usuários". There is a link "Voltar". The login form consists of two input fields: "Login de Acesso:" and "Senha de Acesso:". Below these fields is a green button labeled "Acessar Sistema".

## Criando a página de cadastro do usuario:





**Add View**

View name:

Template:

Model class:

Options:

☐ Create as a partial view

☐ Reference script libraries

☒ Use a layout page:

...

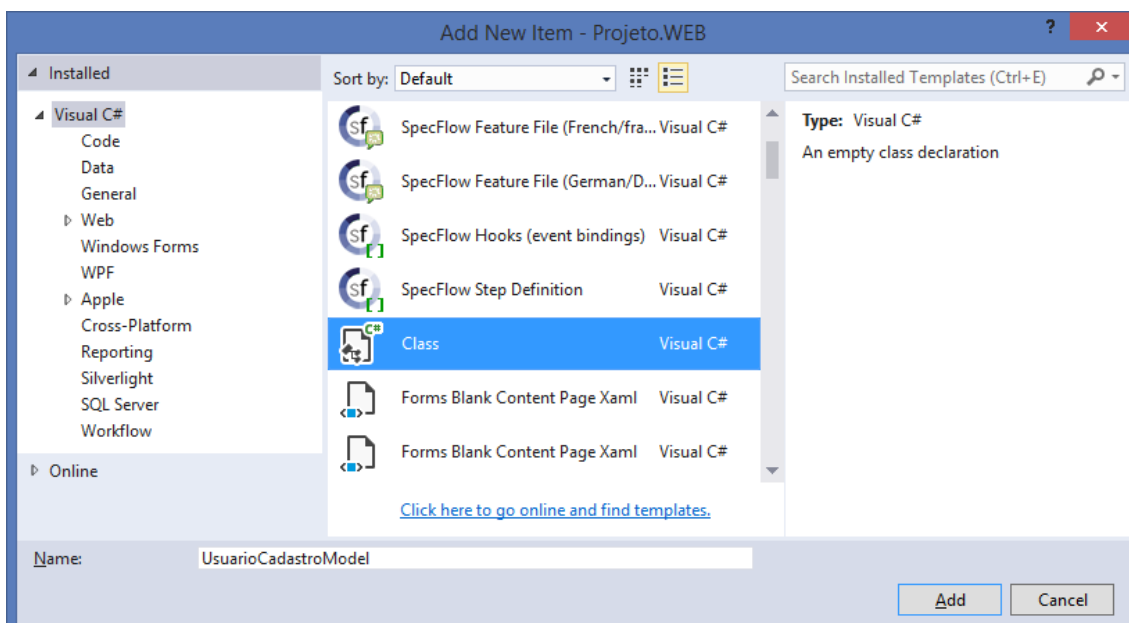
(Leave empty if it is set in a Razor \_viewstart file)

```
@{
    ViewBag.Title = "Cadastro";
    Layout = "~/Views/Shared/Layout.cshtml";
}
```

```
<h4>Criar conta de Usuário</h4>
<a href="/Home/Index">Voltar</a>
<hr/>
```

```
<div class="row">
    <div class="col-md-4">
```

```
    </div>
</div>
```



**Add New Item - Projeto.WEB**

Sort by: Default

Search Installed Templates (Ctrl+E)

Visual C#

- Code
- Data
- General
- Web
  - Windows Forms
  - WPF
- Apple
  - Cross-Platform
  - Reporting
  - Silverlight
  - SQL Server
  - Workflow
- Online

SpecFlow Feature File (French/fra... Visual C#

SpecFlow Feature File (German/D... Visual C#

SpecFlow Hooks (event bindings) Visual C#

SpecFlow Step Definition Visual C#

**Class Visual C#**

Forms Blank Content Page Xaml Visual C#

Forms Blank Content Page Xaml Visual C#

Click here to go online and find templates.

Type: Visual C#

An empty class declaration

Name:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations;

namespace Projeto.WEB.Models
{
    public class UsuarioCadastroModel
    {
        [Required(ErrorMessage = "Informe o nome.")]
        public string Nome { get; set; }

        [Required(ErrorMessage = "Informe o login.")]
        public string Login { get; set; }

        [Required(ErrorMessage = "Informe a senha.")]
        public string Senha { get; set; }

        [Compare("Senha", ErrorMessage = "Senhas não conferem.")]
        [Required(ErrorMessage = "Confirme a senha.")]
        public string SenhaConfirm { get; set; }

        [Required(ErrorMessage = "Envie a foto.")]
        public HttpPostedFileBase Foto { get; set; }
    }
}
```

## Criando o formulário de cadastro de usuario:

```
@{
    ViewBag.Title = "Cadastro";
    Layout = "~/Views/Shared/Layout.cshtml";
}

<h4>Criar conta de Usuário</h4>
<a href="/Home/Index">Voltar</a>
<hr/>

<div class="row">
    <div class="col-md-4">

        <!-- classe de modelo da página -->
        @model Projeto.WEB.Models.UsuarioCadastroModel

        <!-- criando o formulário.. -->
        @using (Html.BeginForm())
        {
            <label>Nome do Usuário:</label>
            @Html.TextBoxFor(model => model.Nome,
                new
                {
                    @class = "form-control",
                    @placeholder = "Nome do Usuário"
                })
            <br />
        }
```

```
<label>Login de Acesso:</label>
@Html.TextBoxFor(model => model.Login,
    new
    {
        @class = "form-control",
        @placeholder = "Login de Acesso"
    })
<br />
<label>Senha de Acesso:</label>
@Html.PasswordFor(model => model.Senha,
    new
    {
        @class = "form-control",
        @placeholder = "Senha de Acesso"
    })
<br />
<label>Confirme sua Senha:</label>
@Html.PasswordFor(model => model.SenhaConfirm,
    new
    {
        @class = "form-control",
        @placeholder = "Confirme sua Senha"
    })
<br />
<label>Envie sua Foto:</label>
<input type="file" name="Foto" class="form-control"/>
<br/>
<input type="submit" value="Cadastrar Usuário"
    class="btn btn-success"/>
}
</div>
</div>
```

## Executando:



## Criando as requisições POST para enviar os dados de cada formulario para o controller:

```
@{
    ViewBag.Title = "Login";
    Layout = "~/Views/Shared/Layout.cshtml";
}

<h4>Login de Usuários</h4>
<a href="/Home/Index">Voltar</a>
<hr />

<div class="row">
    <div class="col-md-3">

        <!-- Declarando a classe de modelo.. -->
        @model Projeto.WEB.Models.UsuarioLoginModel

        <!-- Abrindo a área do formulário -->
        @using (Html.BeginForm("Login", "Usuario", FormMethod.Post))
        {
            <label>Login de Acesso:</label>
            @Html.TextBoxFor(model => model.Login,
                new
                {
                    @class = "form-control",
                    @placeholder = "Login de Acesso"
                })
            <span class="text-danger">
                @Html.ValidationMessageFor(model => model.Login)
            </span>
            <br />

            <label>Senha de Acesso:</label>
            @Html.PasswordFor(model => model.Senha,
                new
                {
                    @class = "form-control",
                    @placeholder = "Senha de Acesso"
                });
            <span class="text-danger">
                @Html.ValidationMessageFor(model => model.Senha)
            </span>
            <br />

            <input type="submit" value="Acessar Sistema"
                class="btn btn-success" />
        }

    </div>

</div>

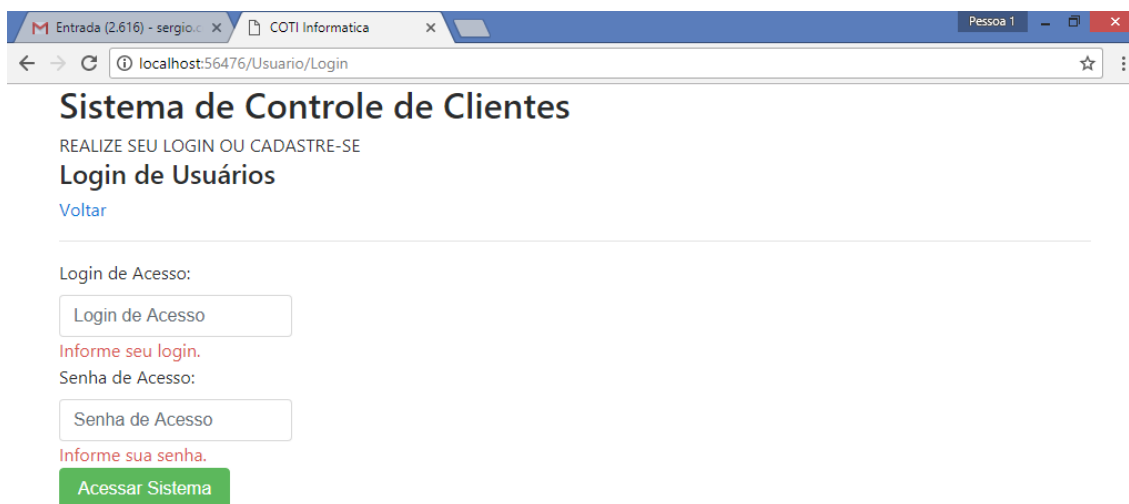
</div>
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Projeto.WEB.Models;

namespace Projeto.WEB.Controllers
{
    public class UsuarioController : Controller
    {
        // GET: Usuario/Login
        public ActionResult Login()
        {
            return View();
        }

        // POST: Usuario/Login
        [HttpPost]
        public ActionResult Login(UsuarioLoginModel model)
        {
            return View();
        }

        // GET: Usuario/Cadastro
        public ActionResult Cadastro()
        {
            return View();
        }
    }
}
```



## Formulário de cadastro:

```
@{
    ViewBag.Title = "Cadastro";
    Layout = "~/Views/Shared/Layout.cshtml";
}

<h4>Criar conta de Usuário</h4>
<a href="/Home/Index">Voltar</a>
```

```
<hr/>

<div class="row">
    <div class="col-md-4">

        <!-- classe de modelo da página -->
        @model Projeto.WEB.Models.UsuarioCadastroModel

        <!-- criando o formulário.. -->
        @using (Html.BeginForm("Cadastro", "Usuario", FormMethod.Post,
                                new { @enctype = "multipart/form-data" }))
        {
            <label>Nome do Usuário:</label>
            @Html.TextBoxFor(model => model.Nome,
                new
                {
                    @class = "form-control",
                    @placeholder = "Nome do Usuário"
                })
            <span class="text-danger">
                @Html.ValidationMessageFor(model => model.Nome)
            </span>
            <br />

            <label>Login de Acesso:</label>
            @Html.TextBoxFor(model => model.Login,
                new
                {
                    @class = "form-control",
                    @placeholder = "Login de Acesso"
                })
            <span class="text-danger">
                @Html.ValidationMessageFor(model => model.Login)
            </span>
            <br />

            <label>Senha de Acesso:</label>
            @Html.PasswordFor(model => model.Senha,
                new
                {
                    @class = "form-control",
                    @placeholder = "Senha de Acesso"
                })
            <span class="text-danger">
                @Html.ValidationMessageFor(model => model.Senha)
            </span>
            <br />

            <label>Confirme sua Senha:</label>
            @Html.PasswordFor(model => model.SenhaConfirm,
                new
                {
                    @class = "form-control",
                    @placeholder = "Confirme sua Senha"
                })
            <span class="text-danger">
                @Html.ValidationMessageFor(model => model.SenhaConfirm)
            </span>
            <br />
        }
    }
</div>
```

```
<label>Envie sua Foto:</label>
<input type="file" name="Foto" class="form-control"/>
<span class="text-danger">
    @Html.ValidationMessageFor(model => model.Foto)
</span>
<br/>

<input type="submit" value="Cadastrar Usuário"
        class="btn btn-success"/>
    }

</div>
</div>
```

## No controller:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Projeto.WEB.Models;

namespace Projeto.WEB.Controllers
{
    public class UsuarioController : Controller
    {
        // GET: Usuario/Login
        public ActionResult Login()
        {
            return View();
        }

        // POST: Usuario/Login
        [HttpPost]
        public ActionResult Login(UsuarioLoginModel model)
        {
            return View();
        }

        // GET: Usuario/Cadastro
        public ActionResult Cadastro()
        {
            return View();
        }

        // POST: Usuario/Cadastro
        [HttpPost]
        public ActionResult Cadastro(UsuarioCadastroModel model)
        {
            return View();
        }
    }
}
```



## Executando:



Entrada (2.616) - sergio.c... x COTI Informatica x COTI Informatica x Pessoa 1

localhost:56476/Usuario/Cadastro

### Sistema de Controle de Clientes

REALIZE SEU LOGIN OU CADASTRE-SE

#### Criar conta de Usuário

[Voltar](#)

Nome do Usuário:

Informe o nome.

Login de Acesso:

Informe o login.

Senha de Acesso:

Informe a senha.

Confirme sua Senha:

Confirme a senha.

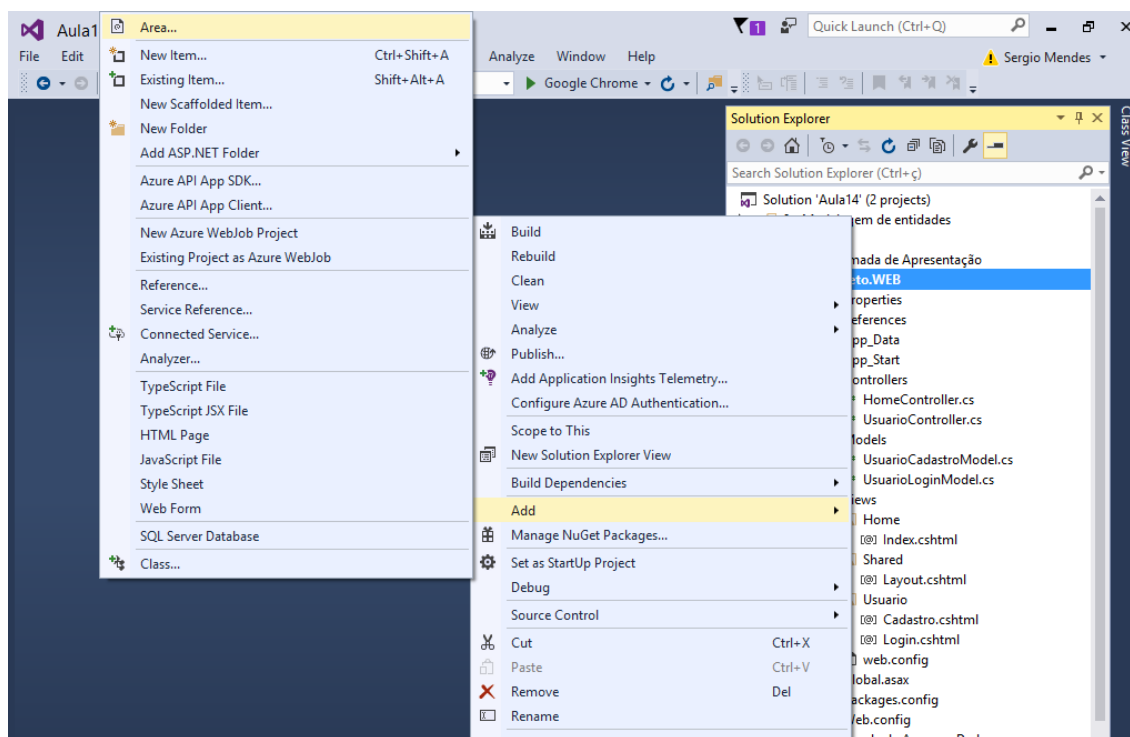
Envie sua Foto:

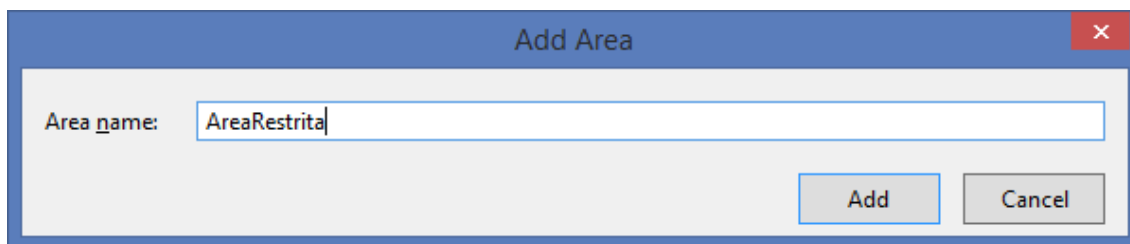
 Nenhum... onado

Envie a foto.

[Cadastrar Usuário](#)

## Criando a area restrita do projeto MVC: (SubProjeto)





Dialog box titled "Add Area". It contains a text input field labeled "Area name:" with the text "AreaRestrita" entered. Below the input field are two buttons: "Add" and "Cancel".

