

Login

Criar uma pasta **Repositories** na raiz do projeto.

Criar uma classe **Funcoes.cs** dentro da pasta Repositories.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;
using WebSite1.Models;

namespace WebSite1.Repositories
{
    public class Funcoes
    {
        public static bool AutenticarUsuario(string login, string senha)
        {
            Context _db = new Context();
            var query = (from u in _db.Clientes
                        where u.Email == login &&
                               u.Senha == senha
                        select u).SingleOrDefault();
            if (query == null)
            {
                return false;
            }
            FormsAuthentication.SetAuthCookie(query.Email, false);
            //HttpContext.Current.Response.Cookies["Usuario"].Value = query.Email;
            //HttpContext.Current.Response.Cookies["Usuario"].Expires = DateTime.Now.AddDays(10);

            HttpContext.Current.Session["Usuario"] = query.Email;
            return true;
        }

        public static Cliente GetUsuario()
        {
            string _login = HttpContext.Current.User.Identity.Name;

            //if (HttpContext.Current.Request.Cookies.Count > 0 ||
            HttpContext.Current.Request.Cookies["Usuario"] != null)
                if (HttpContext.Current.Session.Count > 0 ||
                HttpContext.Current.Session["Usuario"] != null)
                {
                    _login = HttpContext.Current.Session["Usuario"].ToString();
                    //_login = HttpContext.Current.Request.Cookies["Usuario"].Value.ToString();
                    if (_login == "")
                    {
                        return null;
                    }
                    else
                    {
                        Context _db = new Context();
                        Cliente cliente = (from u in _db.Clientes
                                        where u.Email == _login
                                        select u).SingleOrDefault();

                        return cliente;
                    }
                }
            else
            {
                return null;
            }
        }
    }
}
```

```

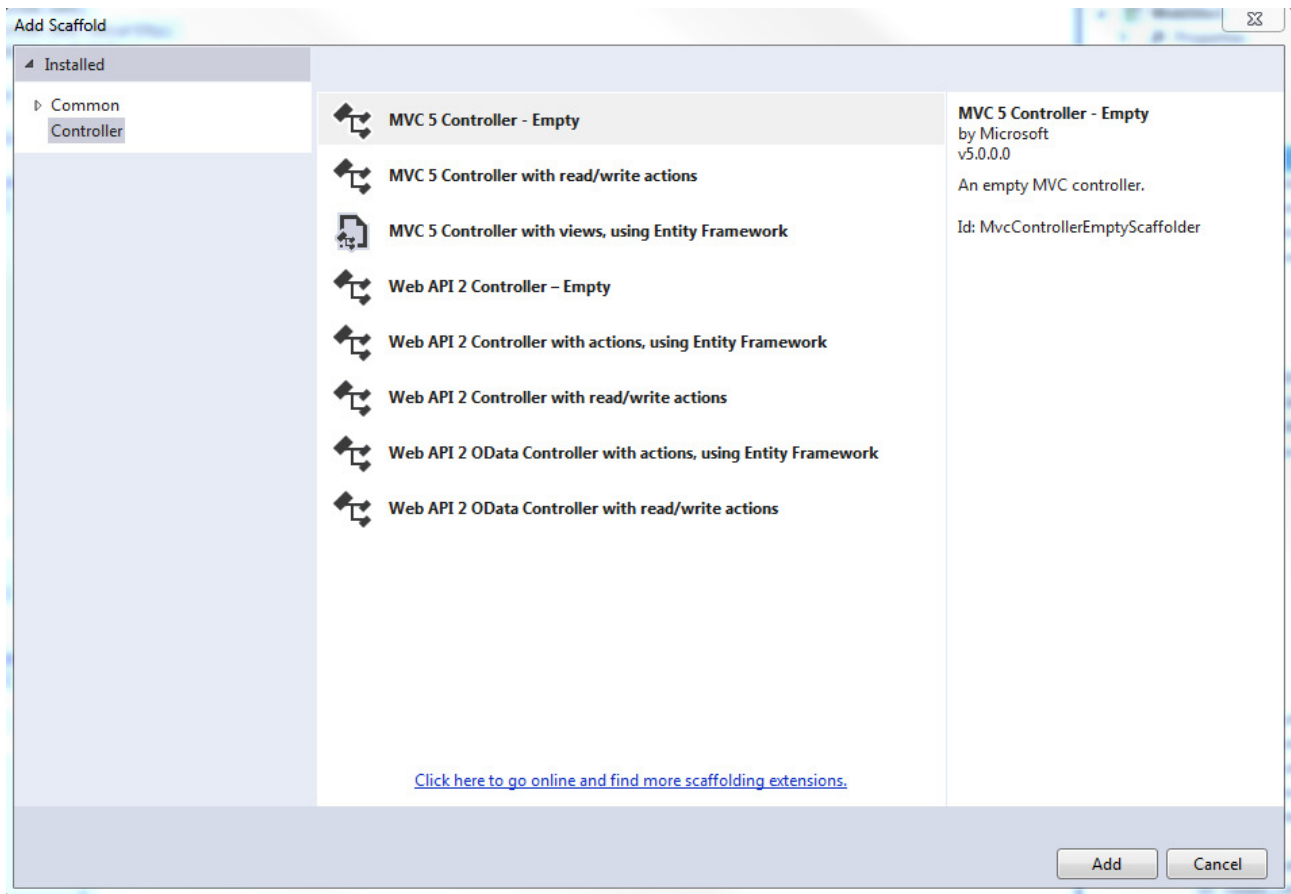
public static Cliente GetUsuario(string _login)
{
    if (_login == "")
    {
        return null;
    }
    else
    {
        Context _db = new Context();
        Cliente cliente = (from u in _db.Clientes
                           where u.Email == _login
                           select u).SingleOrDefault();
        return cliente;
    }
}

public static void Deslogar()
{
    HttpContext.Current.Session["Usuario"] = "";
    //HttpContext.Current.Response.Cookies["Usuario"].Value = "";
    FormsAuthentication.SignOut();
}
}
}

```

Adicionando o Controller

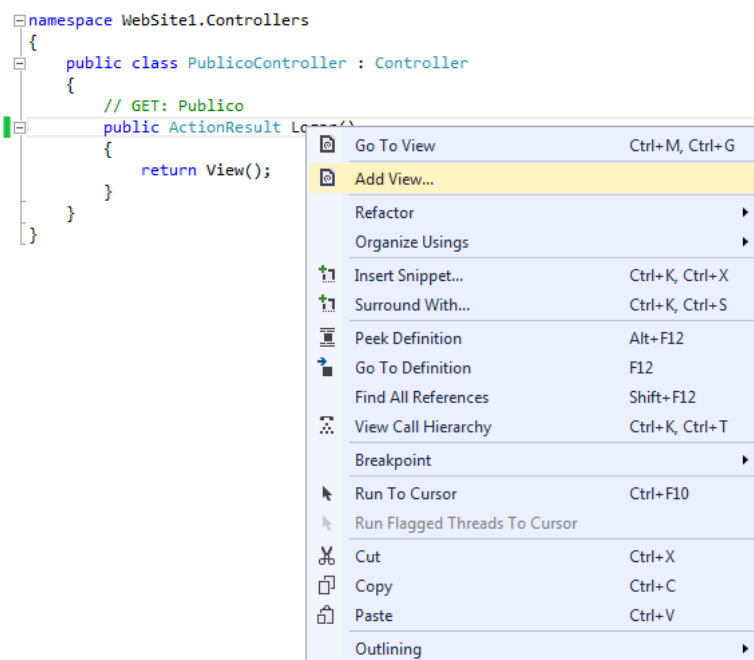
Na pasta Controller, criar um novo Controller
Nome: PublicoController

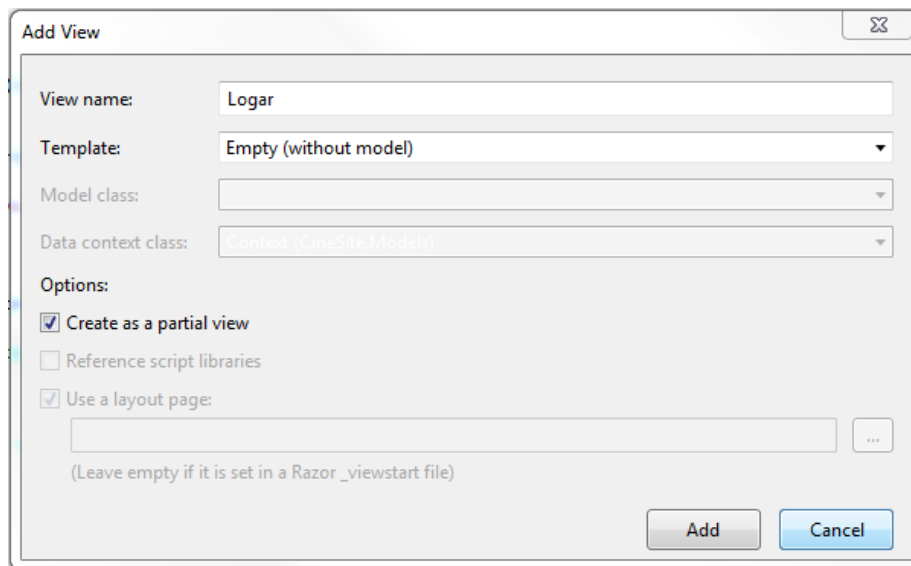


Alterar o nome da Action Index para Logar

```
public ActionResult Logar()  
{  
    return View();  
}
```

Adicionar a View Logar

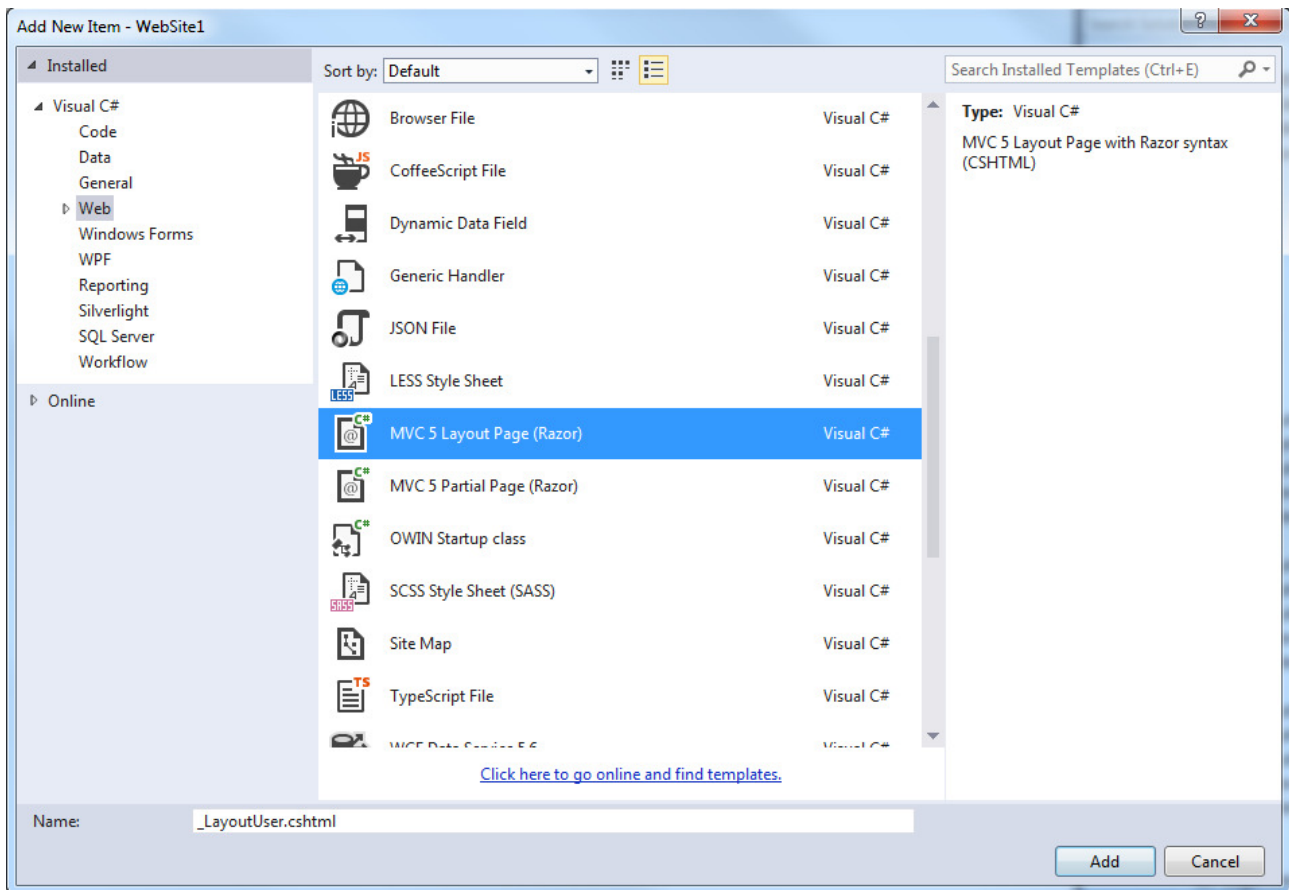




Uma nova página será criada na pasta Views/Publico

Adicionando um novo Layout

Botão direito na pasta Views/Shared > Add New Item



Alterar a página _LayoutUser.cshtml

```

<!DOCTYPE html>

<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>@ViewBag.Title</title>

  @Styles.Render("~/Content/css")
  @Scripts.Render("~/bundles/modernizr")

</head>
<body>
  <div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target=".navbar-collapse">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        @Html.ActionLink("Exemplo", "Logar", "Publico", new { area = "" }, new
{ @class = "navbar-brand" })
      </div>

      <div class="navbar-collapse collapse">
        <ul class="nav navbar-nav">
          @* <li>@Html.ActionLink("Sobre", "About", "Publico")</li> *@
        </ul>
      </div>
    </div>
  </div>

```

```

        </div>
</div>
<div class="container body-content">
    @RenderBody()
    <hr />
    <footer>
        <p>&copy; @DateTime.Now.Year</p>
    </footer>
</div>

@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
@RenderSection("scripts", required: false)
</body>
</html>

```

Alterando a página Views/Publico/Logar.cshtml

```

@model IEnumerable<WebSite1.Models.Cliente>

@{
    Layout = "~/Views/Shared/_LayoutUser.cshtml";
    ViewBag.Title = "Login";
}
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <hr />
        <h3 class="text-center">@ViewBag.Error</h3>
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            <label class="control-label col-md-2">
                Email
            </label>
            <div>
                <div class="col-md-10 ">
                    <input type="text" name="email" id="email" class="form-control" />
                </div>
            </div>
        </div>
        <div class="form-group">
            <label class="control-label col-md-2">
                Senha
            </label>
            <div class="col-md-10">
                <input type="password" name="senha" id="senha" class="form-control" />
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Entrar" class="btn btn-default" />
            </div>
        </div>
    </div>
}

```

Alterando o PublicoController

Adicionar uma Action com HttpPost em PublicoController para receber as informações do Login.

```
[HttpPost]
public ActionResult Logar(string email, string senha)
{
    if (Funcoes.AutenticarUsuario(email, senha) == false)
    {
        ViewBag.Error = "Nome de usuário e/ou senha inválida";
        return View();
    }

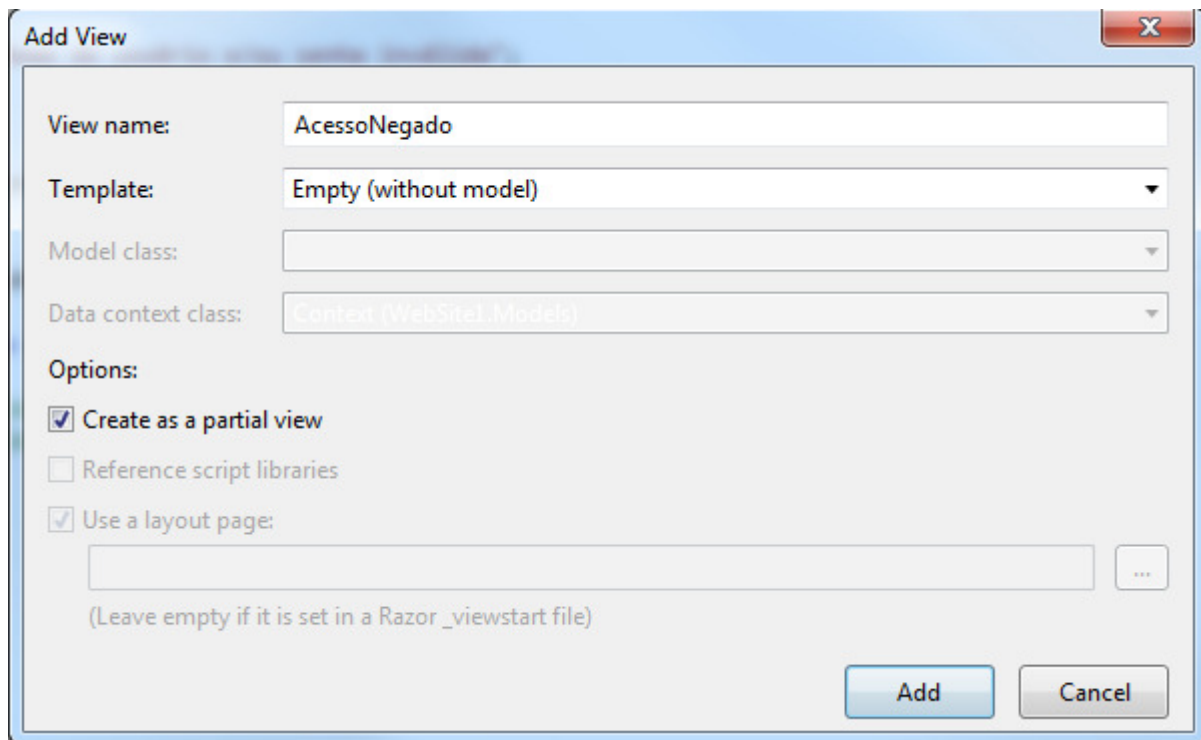
    return RedirectToAction("Index", "Home");
}
```

Adicionar uma Action para AcessoNegado

```
public ActionResult AcessoNegado()
{
    using (Context c = new Context())
    {
        return View();
    }
}
```

Gerar um View para AcessoNegado

Botão direito na Action Acesso Negado



Alterar a Views/Publico/AcessoNegado.cshtml

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

<div class="form-horizontal">
    <br />
    <h2 class="text-center">Acesso Negado</h2>
    @*<h3 class="text-center">@Html.ActionLink("Ir para a tela de Login", "Logar", "Publico")</h3>*@
</div>
```

Alterar o _Layout.cshtml em Views/Shared

Para permitir acesso apenas a quem estiver Logado

```
@{
    WebSite1.Models.Cliente cliente = WebSite1.Repositories.Funcoes.GetUsuario();

    if (cliente != null)
    {
        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav">
                <li>@Html.ActionLink("Home", "Index", "Home")</li>
                <li>@Html.ActionLink("Estados", "Index", "Estados")</li>
                <li>@Html.ActionLink("Cidades", "Index", "Cidades")</li>
                <li>@Html.ActionLink("Clientes", "Index", "Clientes")</li>
                <li>@Html.ActionLink("About", "About", "Home")</li>
                <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
            </ul>
        </div>
    }
    else
    {
        <script type="text/javascript">
            window.location.href = "@Url.Action("AcessoNegado", "Publico")";
        </script>
    }
}
```

Adicionar a opção Logoff

Adicionar uma ao final do menu da tela _Layout.cshtml

```
<li>@Html.ActionLink("Sair", "Logoff", "Publico")</li>
```

Adicionar Logoff no PublicoController

```
public ActionResult Logoff()
{
    WebSite1.Repositories.Funcoes.Deslogar();
    return RedirectToAction("Logar", "Publico");
}
```

Executar a aplicação

Acessar a URL <http://localhost:porta/Publico/Logar>