

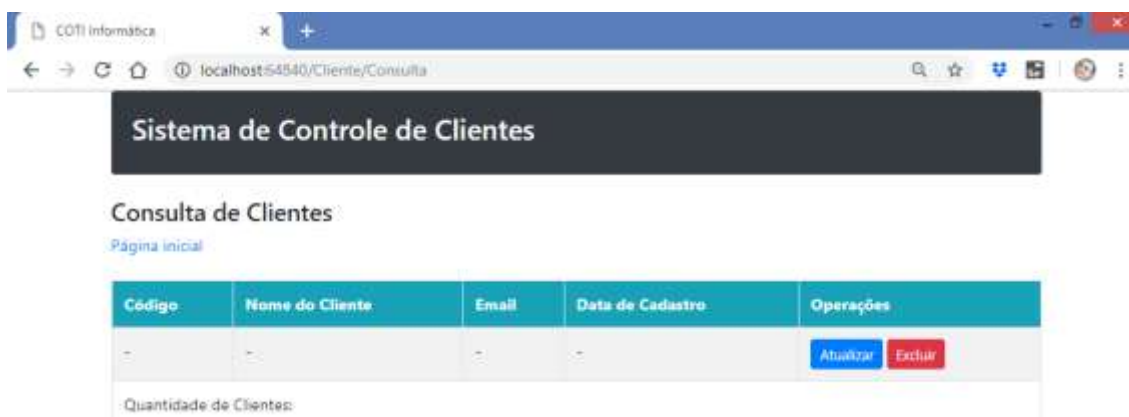
Desenhando a página de consulta de clientes:
/Views/Cliente/Consulta.cshtml

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

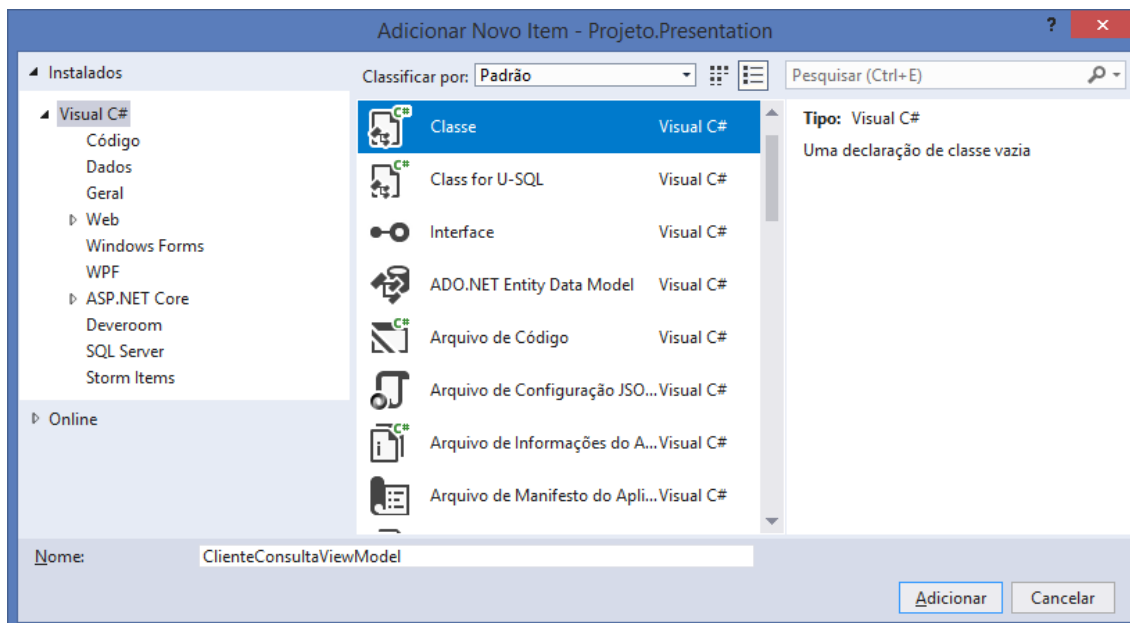
<h4>Consulta de Clientes</h4>
<a href="/Home/Index">Página inicial</a>
<br/>
<br/>

<table class="table table-bordered table-striped table-hover">
    <thead>
        <tr>
            <th class="bg-info text-white">Código</th>
            <th class="bg-info text-white">Nome do Cliente</th>
            <th class="bg-info text-white">Email</th>
            <th class="bg-info text-white">Data de Cadastro</th>
            <th class="bg-info text-white">Operações</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td> - </td>
            <td> - </td>
            <td> - </td>
            <td> - </td>
            <td>
                <a href="/Cliente/Edicao"
                    class="btn btn-primary btn-sm">Atualizar</a>

                <a href="/Cliente/Exclusao"
                    class="btn btn-danger btn-sm">Excluir</a>
            </td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td colspan="5">Quantidade de Clientes:</td>
        </tr>
    </tfoot>
</table>
```



Criando uma classe ViewModel para a consulta de Clientes:



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

namespace Projeto.Presentation.Models
{
    public class ClienteConsultaViewModel
    {
        public int IdCliente { get; set; }
        public string Nome { get; set; }
        public string Email { get; set; }
        public DateTime DataCadastro { get; set; }
    }
}
```

Voltando na classe de controle
/Controllers/ClienteController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Projeto.DAL.Entities; //importando..
using Projeto.BLL.Business; //importando..
using Projeto.Presentation.Models; //importando..

namespace Projeto.Presentation.Controllers
{
}
```

```
public class ClienteController : Controller
{
    // GET: Cliente/Cadastro
    public ActionResult Cadastro()
    {
        return View();
    }

    // POST: Cliente/Cadastro
    [HttpPost] //requisições de formulário
    public ActionResult Cadastro(ClienteCadastroViewModel model)
    {
        //verificar se os campos da model passaram
        //nas regras de validação
        if(ModelState.IsValid)
        {
            try
            {
                Cliente cliente = new Cliente();
                cliente.Nome = model.Nome;
                cliente.Email = model.Email;
                cliente.DataCadastro = DateTime.Now;

                ClienteBusiness business = new ClienteBusiness();
                business.CadastrarCliente(cliente);

                ViewData["Mensagem"] = "Cliente cadastrado com sucesso.";
                ModelState.Clear(); //limpar os campos do formulário
            }
            catch(Exception e)
            {
                ViewData["Mensagem"] = e.Message;
            }
        }

        return View();
    }

    // GET: Cliente/Consulta
    public ActionResult Consulta()
    {
        //declarar uma lista da classe ViewModel..
        List<ClienteConsultaViewModel> lista
            = new List<ClienteConsultaViewModel>();

        try
        {
            //executando a consulta de clientes..
            ClienteBusiness business = new ClienteBusiness();
            List<Cliente> consulta = business.ConsultarTodos();
        }
    }
}
```

```
//varrer a consulta de clientes obtido..
foreach(Cliente cliente in consulta)
{
    ClienteConsultaViewModel model
        = new ClienteConsultaViewModel();
    model.IdCliente = cliente.IdCliente;
    model.Nome = cliente.Nome;
    model.Email = cliente.Email;
    model.DataCadastro = cliente.DataCadastro;

    lista.Add(model);
}
}
catch(Exception e)
{
    //exibir mensagem de erro..
    ViewData["Mensagem"] = e.Message;
}

//abre a página
return View(lista);
}
}
```

Exibindo a lista na página:

```
@model List<Projeto.Presentation.Models.ClienteConsultaViewModel>

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

<h4>Consulta de Clientes</h4>
<a href="/Home/Index">Página inicial</a>
<br/>
<br/>

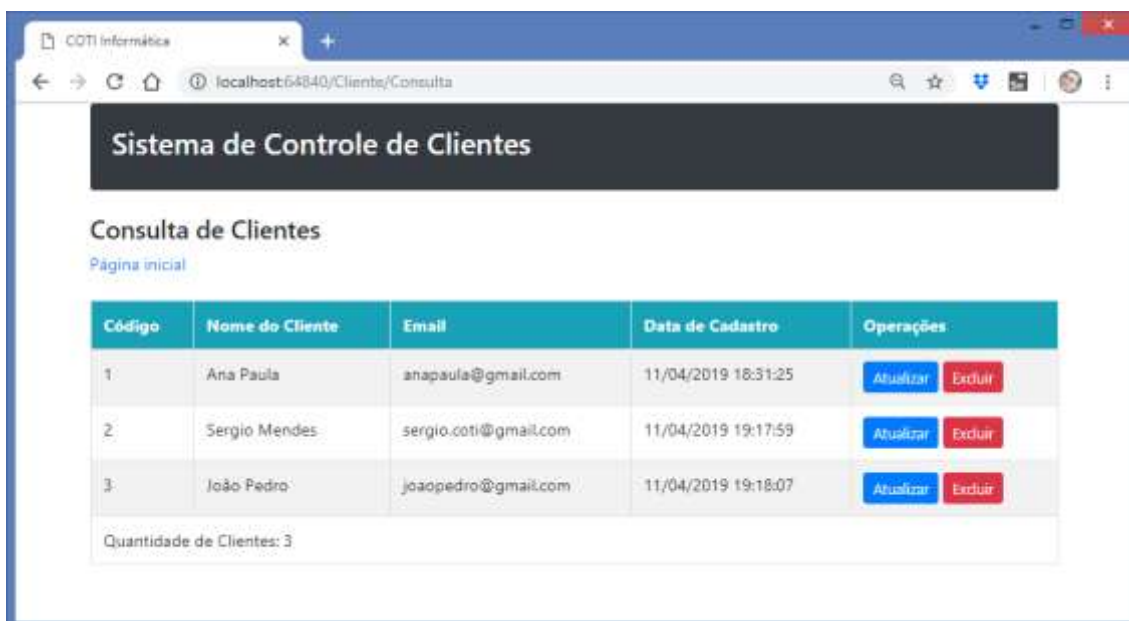
<div>
    <strong>@ViewData["Mensagem"]</strong>
</div>

<table class="table table-bordered table-striped table-hover">
    <thead>
        <tr>
            <th class="bg-info text-white">Código</th>
            <th class="bg-info text-white">Nome do Cliente</th>
            <th class="bg-info text-white">Email</th>
            <th class="bg-info text-white">Data de Cadastro</th>
            <th class="bg-info text-white">Operações</th>
        </tr>
    </thead>
    <tbody>
```

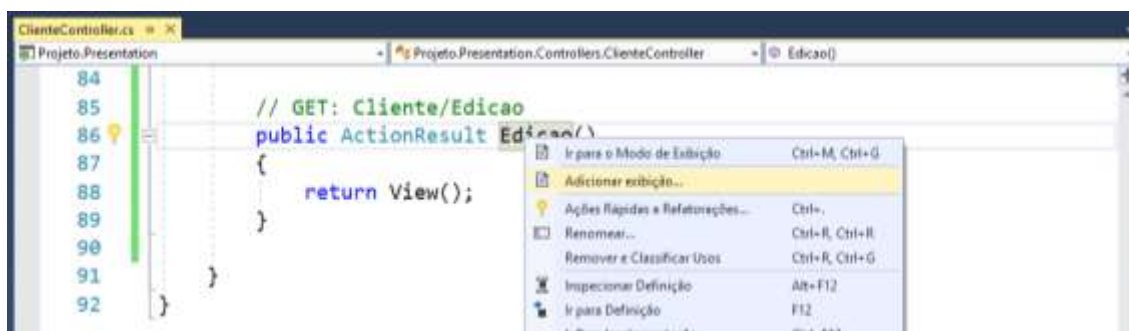
```
@foreach (var item in Model)
{
    <tr>
        <td> @item.IdCliente </td>
        <td> @item.Nome </td>
        <td> @item.Email </td>
        <td> @item.DataCadastro.ToString("dd/MM/yyyy") </td>
        <td>
            <a href="/Cliente/Edicao"
                class="btn btn-primary btn-sm">Atualizar</a>

            <a href="/Cliente/Exclusao"
                class="btn btn-danger btn-sm">Excluir</a>
        </td>
    </tr>
}

</tbody>
<tfoot>
    <tr>
        <td colspan="5">Quantidade de Clientes: @Model.Count</td>
    </tr>
</tfoot>
</table>
```



Criando a página de edição:



Adicionar modo de exibição

Nome do modo de exibição:

Modelo:

Classe do modelo:

Opções:

☐ Criar como um Modo de exibição parcial

☐ Bibliotecas de scripts de referência

☒ Usar uma página de layout:

...

(deixe em branco se ele estiver definido em um arquivo Razor _viewstart)

Adicionar Cancelar

@{

```
ViewBag.Title = "Edicao";
Layout = "~/Views/Shared/Layout.cshtml";
```

}

```
<h4>Atualizar dados de Cliente</h4>
<a href="/Cliente/Consulta">Consulta de Clientes</a>
<br />
<br />
```

Criando uma classe ViewModel para a página de Edição:

Adicionar Novo Item - Projeto.Presentation

Classificar por: Padrão

Pesquisar (Ctrl+E)

Visual C#

- Código
- Dados
- Geral
- Web
- Windows Forms
- WPF
- ASP.NET Core
- Deveroom
- SQL Server
- Storm Items
- Online

Ícone	Nome	Tipo
	Classe	Visual C#
	Class for U-SQL	Visual C#
	Interface	Visual C#
	ADO.NET Entity Data Model	Visual C#
	Arquivo de Código	Visual C#
	Arquivo de Configuração JSO...	Visual C#
	Arquivo de Informações do A...	Visual C#
	Arquivo de Manifesto do Apli...	Visual C#

Tipo: Visual C#

Uma declaração de classe vazia

Nome:

Adicionar Cancelar

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

namespace Projeto.Presentation.Models
{
    public class ClienteEdicaoViewModel
    {
        public int IdCliente { get; set; }

        [MinLength(6, ErrorMessage = "Informe no mínimo {1} caracteres.")]
        [MaxLength(100, ErrorMessage = "Informe no máximo {1} caracteres.")]
        [Required(ErrorMessage = "Por favor, informe o nome do cliente.")]
        public string Nome { get; set; }

        [EmailAddress(ErrorMessage = "Informe um endereço de email válido.")]
        [Required(ErrorMessage = "Por favor, informe o email do cliente.")]
        public string Email { get; set; }

        public DateTime DataCadastro { get; set; }
    }
}
```

Página de edição:

```
@model Projeto.Presentation.Models.ClienteEdicaoViewModel
```

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

```
<h4>Atualizar dados de Cliente</h4>
<a href="/Cliente/Consulta">Consulta de Clientes</a>
<br />
<br />
```

```
@using (Html.BeginForm())
{
    <div class="text-danger">
        @Html.ValidationSummary()
    </div>

    <label>Código do Cliente:</label>
    @Html.TextBoxFor(model => model.IdCliente,
        new { @class = "form-control col-md-4", @readonly = "readonly" })
    <br />

    <label>Nome do Cliente:</label>
    @Html.TextBoxFor(model => model.Nome,
        new { @class = "form-control col-md-4" })
    <br />

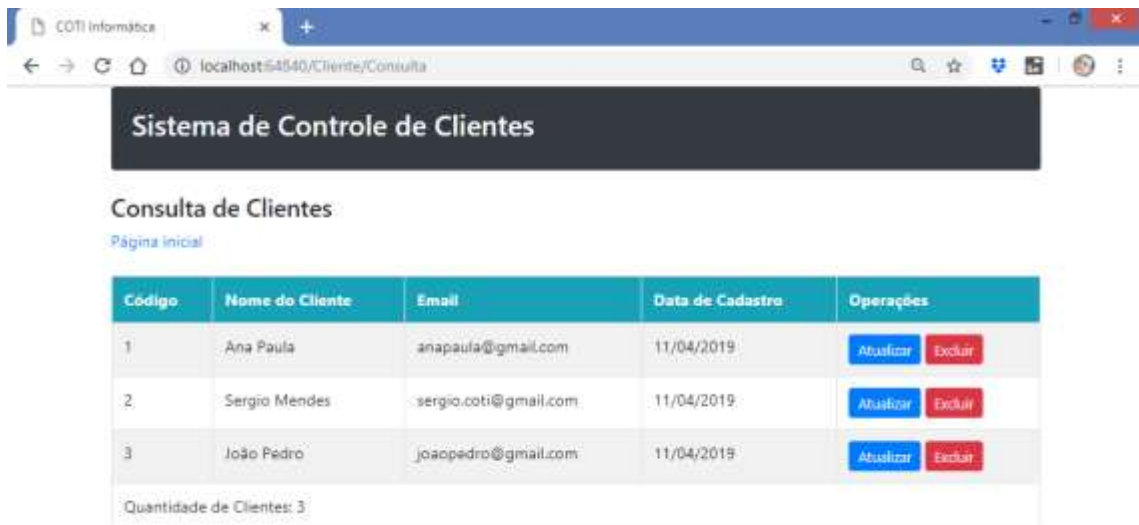
    <label>Email do Cliente:</label>
    @Html.TextBoxFor(model => model.Email,
        new { @class = "form-control col-md-4" })
    <br />
}
```

```
<label>Data de Cadastro:</label>
@Html.TextBoxFor(model => model.DataCadastro,
    new { @class = "form-control col-md-4", @readonly = "readonly" })
<br />

<input type="submit" value="Atualizar Cliente" class="btn btn-primary"/>
<br/>
<br/>

<strong>@ViewData["Mensagem"]</strong>

}
```



QueryString

Passagem de parâmetros por URL:

```
@model List<Projeto.Presentation.Models.ClienteConsultaViewModel>

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

<h4>Consulta de Clientes</h4>
<a href="/Home/Index">Página inicial</a>
<br/>
<br/>

<div>
    <strong>@ViewData["Mensagem"]</strong>
</div>

<table class="table table-bordered table-striped table-hover">
    <thead>
        <tr>
            <th class="bg-info text-white">Código</th>
            <th class="bg-info text-white">Nome do Cliente</th>
            <th class="bg-info text-white">Email</th>
            <th class="bg-info text-white">Data de Cadastro</th>
            <th class="bg-info text-white">Operações</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td> @item.IdCliente </td>
                <td> @item.Nome </td>
                <td> @item.Email </td>
                <td> @item.DataCadastro.ToString("dd/MM/yyyy") </td>
                <td>
                    <a href="/Cliente/Edicao/@item.IdCliente"
                       class="btn btn-primary btn-sm">Atualizar</a>

                    <a href="/Cliente/Exclusao/@item.IdCliente"
                       class="btn btn-danger btn-sm">Excluir</a>
                </td>
            </tr>
        }
    </tbody>
    <tfoot>
        <tr>
            <td colspan="5">Quantidade de Clientes: @Model.Count</td>
        </tr>
    </tfoot>
</table>
```

Exemplo:

<http://localhost:64840/Cliente/Edicao/1>

[Controller] [View] [Id]

Recuperando o id do cliente:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Projeto.DAL.Entities; //importando..
using Projeto.BLL.Business; //importando..
using Projeto.Presentation.Models; //importando..

namespace Projeto.Presentation.Controllers
{
    public class ClienteController : Controller
    {
        // GET: Cliente/Cadastro
        public ActionResult Cadastro()
        {
            return View();
        }

        // POST: Cliente/Cadastro
        [HttpPost] //requisições de formulário
        public ActionResult Cadastro(ClienteCadastroViewModel model)
        {
            //verificar se os campos da model passaram
            //nas regras de validação
            if(ModelState.IsValid)
            {
                try
                {
                    Cliente cliente = new Cliente();
                    cliente.Nome = model.Nome;
                    cliente.Email = model.Email;
                    cliente.DataCadastro = DateTime.Now;

                    ClienteBusiness business = new ClienteBusiness();
                    business.CadastrarCliente(cliente);

                    ViewData["Mensagem"] = "Cliente cadastrado com sucesso.";
                    ModelState.Clear(); //limpar os campos do formulário
                }
                catch(Exception e)
                {
                    ViewData["Mensagem"] = e.Message;
                }
            }

            return View();
        }

        // GET: Cliente/Consulta
        public ActionResult Consulta()
        {

```

```
//declarar uma lista da classe ViewModel..
List<ClienteConsultaViewModel> lista
    = new List<ClienteConsultaViewModel>();

try
{
    //executando a consulta de clientes..
    ClienteBusiness business = new ClienteBusiness();
    List<Cliente> consulta = business.ConsultarTodos();

    //varrer a consulta de clientes obtido..
    foreach(Cliente cliente in consulta)
    {
        ClienteConsultaViewModel model
            = new ClienteConsultaViewModel();
        model.IdCliente = cliente.IdCliente;
        model.Nome = cliente.Nome;
        model.Email = cliente.Email;
        model.DataCadastro = cliente.DataCadastro;

        lista.Add(model);
    }
}
catch(Exception e)
{
    //exibir mensagem de erro..
    ViewData["Mensagem"] = e.Message;
}

//abre a página
return View(lista);
}

// GET: Cliente/Edicao/id
public ActionResult Edicao(int id)
{
    ClienteEdicaoViewModel model = new ClienteEdicaoViewModel();

    try
    {
        //buscando o cliente pelo id..
        ClienteBusiness business = new ClienteBusiness();
        Cliente cliente = business.ConsultarPorId(id);

        model.IdCliente = cliente.IdCliente;
        model.Nome = cliente.Nome;
        model.Email = cliente.Email;
        model.DataCadastro = cliente.DataCadastro;
    }
    catch(Exception e)
    {
        ViewData["Mensagem"] = e.Message;
    }

    return View(model);
}
}
```

Executando:



Sistema de Controle de Clientes

Atualizar dados de Cliente

[Consulta de Clientes](#)

Código do Cliente:

1

Nome do Cliente:

Ana Paula

Email do Cliente:

anapaula@gmail.com

Data de Cadastro:

11/04/2019 18:31:25

Atualizar Cliente

Criando um método HttpPost para atualizar os dados do cliente:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Projeto.DAL.Entities; //importando..
using Projeto.BLL.Business; //importando..
using Projeto.Presentation.Models; //importando..

namespace Projeto.Presentation.Controllers
{
    public class ClienteController : Controller
    {
        // GET: Cliente/Cadastro
        public ActionResult Cadastro()
        {
            return View();
        }

        // POST: Cliente/Cadastro
        [HttpPost] //requisições de formulário
        public ActionResult Cadastro(ClienteCadastroViewModel model)
        {
            //verificar se os campos da model passaram
            //nas regras de validação
        }
    }
}
```

```
if(ModelState.IsValid)
{
    try
    {
        Cliente cliente = new Cliente();
        cliente.Nome = model.Nome;
        cliente.Email = model.Email;
        cliente.DataCadastro = DateTime.Now;

        ClienteBusiness business = new ClienteBusiness();
        business.CadastrarCliente(cliente);

        ViewData["Mensagem"] = "Cliente cadastrado com sucesso.";
        ModelState.Clear(); //limpar os campos do formulário
    }
    catch(Exception e)
    {
        ViewData["Mensagem"] = e.Message;
    }
}

return View();
}

// GET: Cliente/Consulta
public ActionResult Consulta()
{
    //declarar uma lista da classe ViewModel..
    List<ClienteConsultaViewModel> lista
        = new List<ClienteConsultaViewModel>();

    try
    {
        //executando a consulta de clientes..
        ClienteBusiness business = new ClienteBusiness();
        List<Cliente> consulta = business.ConsultarTodos();

        //varrer a consulta de clientes obtido..
        foreach(Cliente cliente in consulta)
        {
            ClienteConsultaViewModel model
                = new ClienteConsultaViewModel();
            model.IdCliente = cliente.IdCliente;
            model.Nome = cliente.Nome;
            model.Email = cliente.Email;
            model.DataCadastro = cliente.DataCadastro;

            lista.Add(model);
        }
    }
    catch(Exception e)
    {
        //exibir mensagem de erro..
        ViewData["Mensagem"] = e.Message;
    }

    //abre a página
    return View(lista);
}
```

```
// GET: Cliente/Edicao/id
public ActionResult Edicao(int id)
{
    ClienteEdicaoViewModel model = new ClienteEdicaoViewModel();

    try
    {
        //buscando o cliente pelo id..
        ClienteBusiness business = new ClienteBusiness();
        Cliente cliente = business.ConsultarPorId(id);

        model.IdCliente = cliente.IdCliente;
        model.Nome = cliente.Nome;
        model.Email = cliente.Email;
        model.DataCadastro = cliente.DataCadastro;
    }
    catch(Exception e)
    {
        ViewData["Mensagem"] = e.Message;
    }

    return View(model);
}

[HttpPost] //recebe o SUBMIT do formulário
public ActionResult Edicao(ClienteEdicaoViewModel model)
{
    //verificar se os dados da model passaram
    //nas regras de validação..
    if(ModelState.IsValid)
    {
        try
        {
            Cliente cliente = new Cliente();

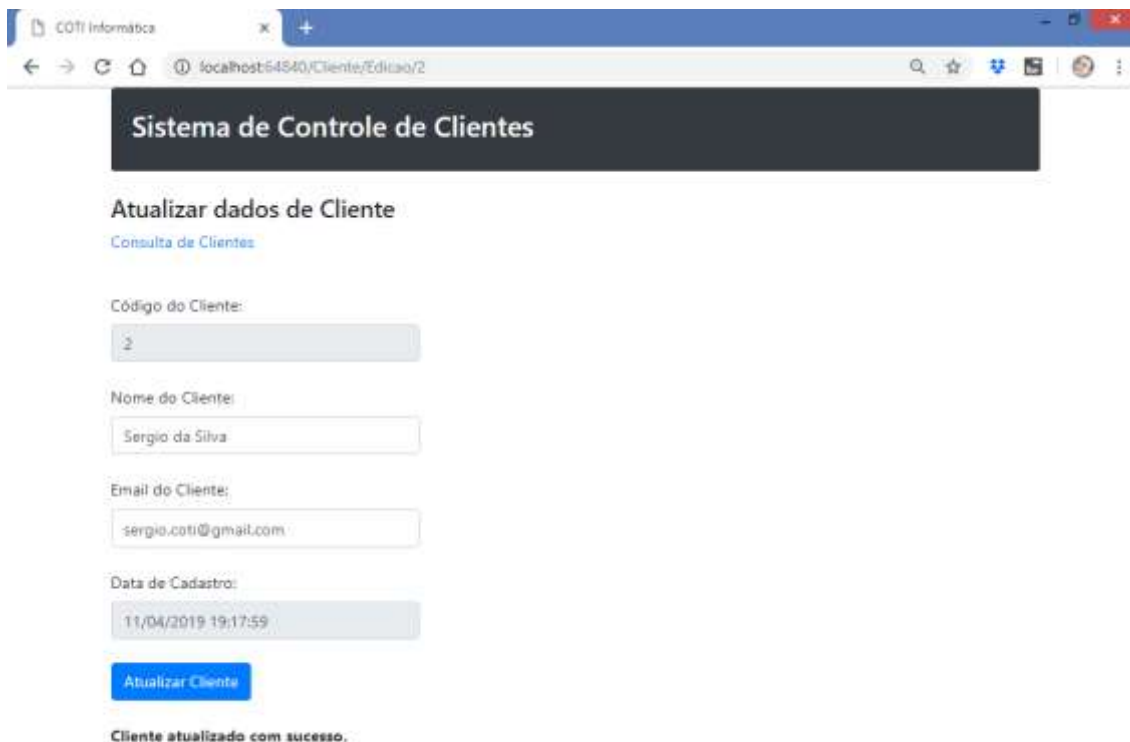
            cliente.IdCliente = model.IdCliente;
            cliente.Nome = model.Nome;
            cliente.Email = model.Email;

            ClienteBusiness business = new ClienteBusiness();
            business.AtualizarCliente(cliente);

            ViewData["Mensagem"] = "Cliente atualizado com sucesso.";
        }
        catch(Exception e)
        {
            ViewData["Mensagem"] = e.Message;
        }
    }

    return View();
}
}
```

Executando:



Implementando a exclusão:

```

28
29 @foreach (var item in Model)
30 {
31     <tr>
32         <td> @item.IdCliente </td>
33         <td> @item.Nome </td>
34         <td> @item.Email </td>
35         <td> @item.DataCadastro.ToString("dd/MM/yyyy") </td>
36         <td>
37             <a href="/Cliente/Edicao/@item.IdCliente"
38                 class="btn btn-primary btn-sm">Atualizar</a>
39
40             <a href="/Cliente/Exclusao/@item.IdCliente"
41                 class="btn btn-danger btn-sm">Excluir</a>
42         </td>
43     </tr>
44 }
45
46 </tbody>
47 <tfoot>
48     <tr>
49         <td colspan="5">Quantidade de Clientes: @Model.Count</td>
50     </tr>
51 </tfoot>
52 </table>

```

```
@model List<Projeto.Presentation.Models.ClienteConsultaViewModel>

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

<h4>Consulta de Clientes</h4>
<a href="/Home/Index">Página inicial</a>
<br/>
<br/>

<div>
    <strong>@ViewData["Mensagem"]</strong>
</div>

<table class="table table-bordered table-striped table-hover">
    <thead>
        <tr>
            <th class="bg-info text-white">Código</th>
            <th class="bg-info text-white">Nome do Cliente</th>
            <th class="bg-info text-white">Email</th>
            <th class="bg-info text-white">Data de Cadastro</th>
            <th class="bg-info text-white">Operações</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td> @item.IdCliente </td>
                <td> @item.Nome </td>
                <td> @item.Email </td>
                <td> @item.DataCadastro.ToString("dd/MM/yyyy") </td>
                <td>
                    <a href="/Cliente/Edicao/@item.IdCliente"
                       class="btn btn-primary btn-sm">Atualizar</a>

                    <a href="/Cliente/Exclusao/@item.IdCliente"
                       class="btn btn-danger btn-sm"
                       onclick="return confirm('Deseja excluir
                                   o Cliente @item.Nome?');">Excluir</a>
                </td>
            </tr>
        }
    </tbody>
</table>

<tfoot>
    <tr>
        <td colspan="5">Quantidade de Clientes: @Model.Count</td>
    </tr>
</tfoot>
</table>
```


Implementando a exclusão:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Projeto.DAL.Entities; //importando..
using Projeto.BLL.Business; //importando..
using Projeto.Presentation.Models; //importando..

namespace Projeto.Presentation.Controllers
{
    public class ClienteController : Controller
    {
        // GET: Cliente/Cadastro
        public ActionResult Cadastro()
        {
            return View();
        }

        // POST: Cliente/Cadastro
        [HttpPost] //requisições de formulário
        public ActionResult Cadastro(ClienteCadastroViewModel model)
        {
            //verificar se os campos da model passaram
            //nas regras de validação
            if(ModelState.IsValid)
            {
                try
                {
                    Cliente cliente = new Cliente();
                    cliente.Nome = model.Nome;
                    cliente.Email = model.Email;
                    cliente.DataCadastro = DateTime.Now;

                    ClienteBusiness business = new ClienteBusiness();
                    business.CadastrarCliente(cliente);

                    ViewData["Mensagem"] = "Cliente cadastrado com sucesso.";
                    ModelState.Clear(); //limpar os campos do formulário
                }
                catch(Exception e)
                {
                    ViewData["Mensagem"] = e.Message;
                }
            }

            return View();
        }

        // GET: Cliente/Consulta
        public ActionResult Consulta()
        {
            //declarar uma lista da classe ViewModel..
            List<ClienteConsultaViewModel> lista
                = new List<ClienteConsultaViewModel>();
        }
    }
}
```

```
try
{
    //executando a consulta de clientes..
    ClienteBusiness business = new ClienteBusiness();
    List<Cliente> consulta = business.ConsultarTodos();

    //varrer a consulta de clientes obtido..
    foreach(Cliente cliente in consulta)
    {
        ClienteConsultaViewModel model
            = new ClienteConsultaViewModel();

        model.IdCliente = cliente.IdCliente;
        model.Nome = cliente.Nome;
        model.Email = cliente.Email;
        model.DataCadastro = cliente.DataCadastro;

        lista.Add(model);
    }
}
catch(Exception e)
{
    //exibir mensagem de erro..
    ViewData["Mensagem"] = e.Message;
}

//abre a página
return View(lista);
}

// GET: Cliente/Edicao/id
public ActionResult Edicao(int id)
{
    ClienteEdicaoViewModel model = new ClienteEdicaoViewModel();

    try
    {
        //buscando o cliente pelo id..
        ClienteBusiness business = new ClienteBusiness();
        Cliente cliente = business.ConsultarPorId(id);

        model.IdCliente = cliente.IdCliente;
        model.Nome = cliente.Nome;
        model.Email = cliente.Email;
        model.DataCadastro = cliente.DataCadastro;
    }
    catch(Exception e)
    {
        ViewData["Mensagem"] = e.Message;
    }

    return View(model);
}

[HttpPost] //recebe o SUBMIT do formulário
public ActionResult Edicao(ClienteEdicaoViewModel model)
{
    //verificar se os dados da model passaram
    //nas regras de validação..
}
```

```

        if(ModelState.IsValid)
        {
            try
            {
                Cliente cliente = new Cliente();

                cliente.IdCliente = model.IdCliente;
                cliente.Nome = model.Nome;
                cliente.Email = model.Email;

                ClienteBusiness business = new ClienteBusiness();
                business.AtualizarCliente(cliente);

                ViewData["Mensagem"] = "Cliente atualizado com sucesso.";
            }
            catch(Exception e)
            {
                ViewData["Mensagem"] = e.Message;
            }
        }

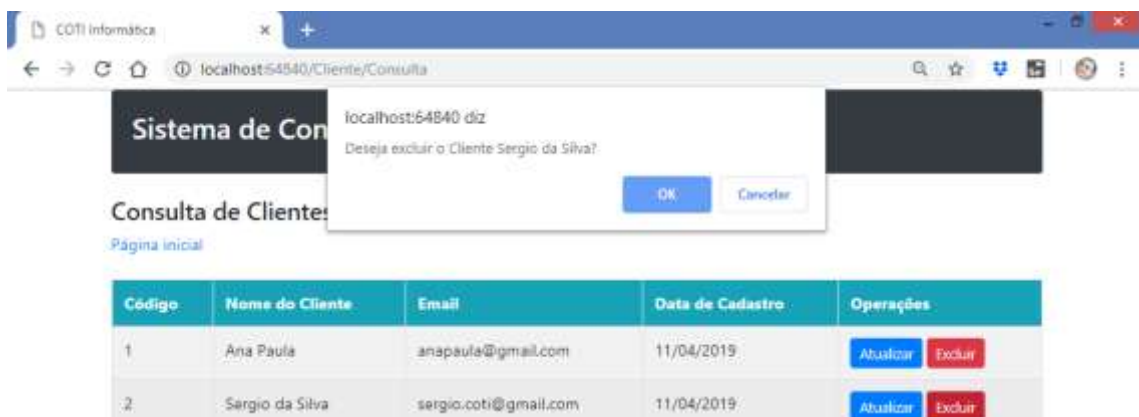
        return View();
    }

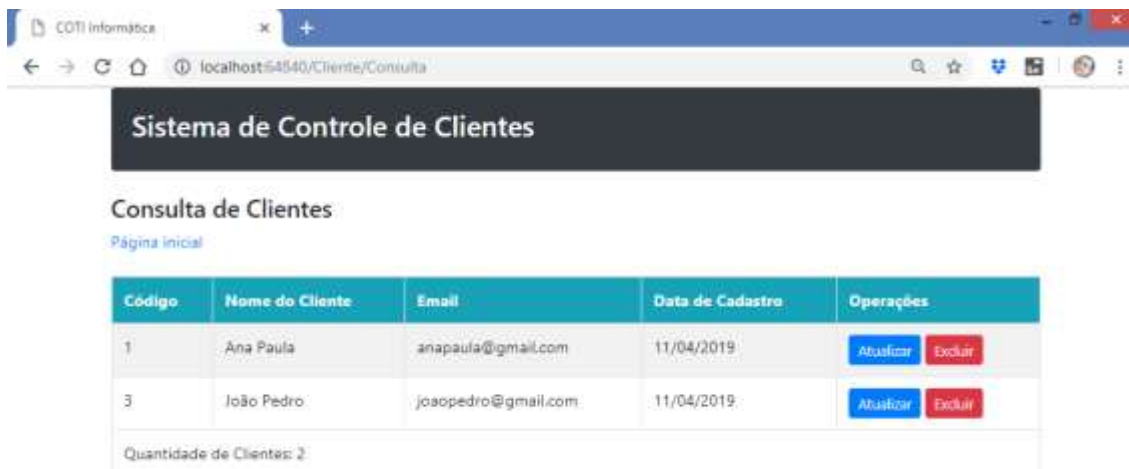
    public ActionResult Exclusao(int id)
    {
        try
        {
            ClienteBusiness business = new ClienteBusiness();
            business.ExcluirCliente(id);

            ViewData["Mensagem"] = "Cliente excluído com sucesso.";
        }
        catch(Exception e)
        {
            ViewData["Mensagem"] = e.Message;
        }

        //redirecionar para a página da consulta..
        return RedirectToAction("Consulta");
    }
}
}

```



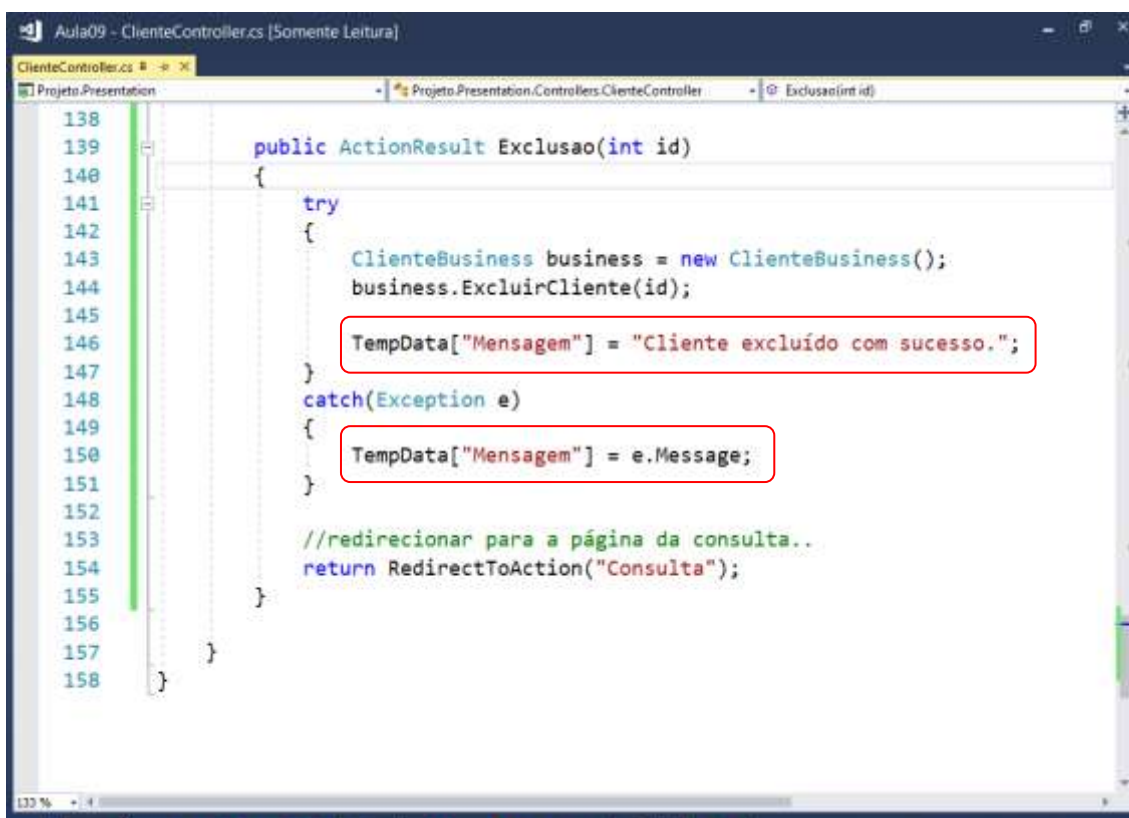


Código	Nome do Cliente	Email	Data de Cadastro	Operações
1	Ana Paula	anapaula@gmail.com	11/04/2019	Atualizar Excluir
3	João Pedro	joaopedro@gmail.com	11/04/2019	Atualizar Excluir

Quantidade de Clientes: 2

TempData

É capaz de transportar um conteúdo para uma página após a execução do comando **RedirectToAction**



```

138
139 public ActionResult Exclusao(int id)
140 {
141     try
142     {
143         ClienteBusiness business = new ClienteBusiness();
144         business.ExcluirCliente(id);
145         TempData["Mensagem"] = "Cliente excluído com sucesso.";
146     }
147     catch (Exception e)
148     {
149         TempData["Mensagem"] = e.Message;
150     }
151
152     //redirecionar para a página da consulta..
153     return RedirectToAction("Consulta");
154 }
155
156
157
158

```

```
@model List<Projeto.Presentation.Models.ClienteConsultaViewModel>
```

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

```
<h4>Consulta de Clientes</h4>
```

```
<a href="/Home/Index">Página inicial</a>
<br/>
<br/>

<div>
    <strong>@TempData["Mensagem"]</strong>
</div>

<table class="table table-bordered table-striped table-hover">
    <thead>
        <tr>
            <th class="bg-info text-white">Código</th>
            <th class="bg-info text-white">Nome do Cliente</th>
            <th class="bg-info text-white">Email</th>
            <th class="bg-info text-white">Data de Cadastro</th>
            <th class="bg-info text-white">Operações</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td> @item.IdCliente </td>
                <td> @item.Nome </td>
                <td> @item.Email </td>
                <td> @item.DataCadastro.ToString("dd/MM/yyyy") </td>
                <td>
                    <a href="/Cliente/Edicao/@item.IdCliente"
                       class="btn btn-primary btn-sm">Atualizar</a>

                    <a href="/Cliente/Exclusao/@item.IdCliente"
                       class="btn btn-danger btn-sm"
                       onclick="return confirm('Deseja excluir
                                   o Cliente @item.Nome?');" >Excluir</a>
                </td>
            </tr>
        }
    </tbody>
</table>

<tfoot>
    <tr>
        <td colspan="5">Quantidade de Clientes: @Model.Count</td>
    </tr>
</tfoot>
</table>
```

Sistema de Controle de Clientes

Consulta de Clientes

[Página inicial](#)

Cliente excluído com sucesso.

Código	Nome do Cliente	Email	Data de Cadastro	Operações
3	João Pedro	joaopedro@gmail.com	11/04/2019	Atualizar Excluir

Quantidade de Clientes: 1