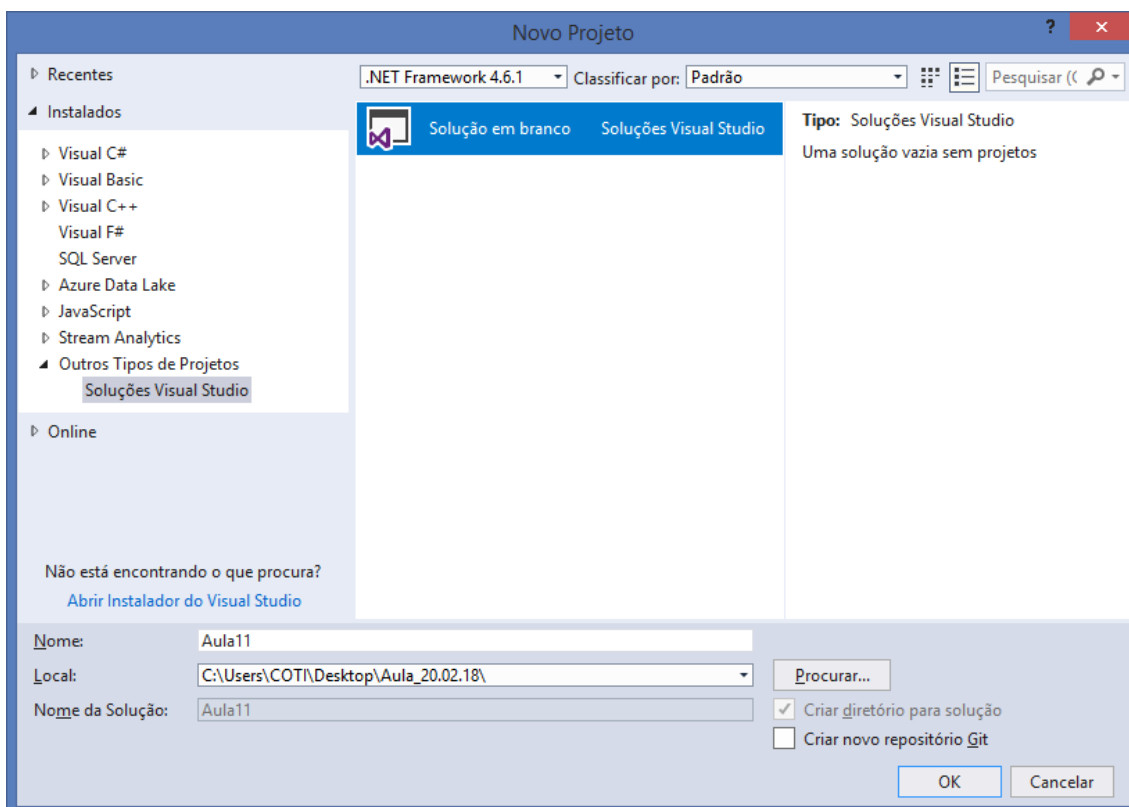
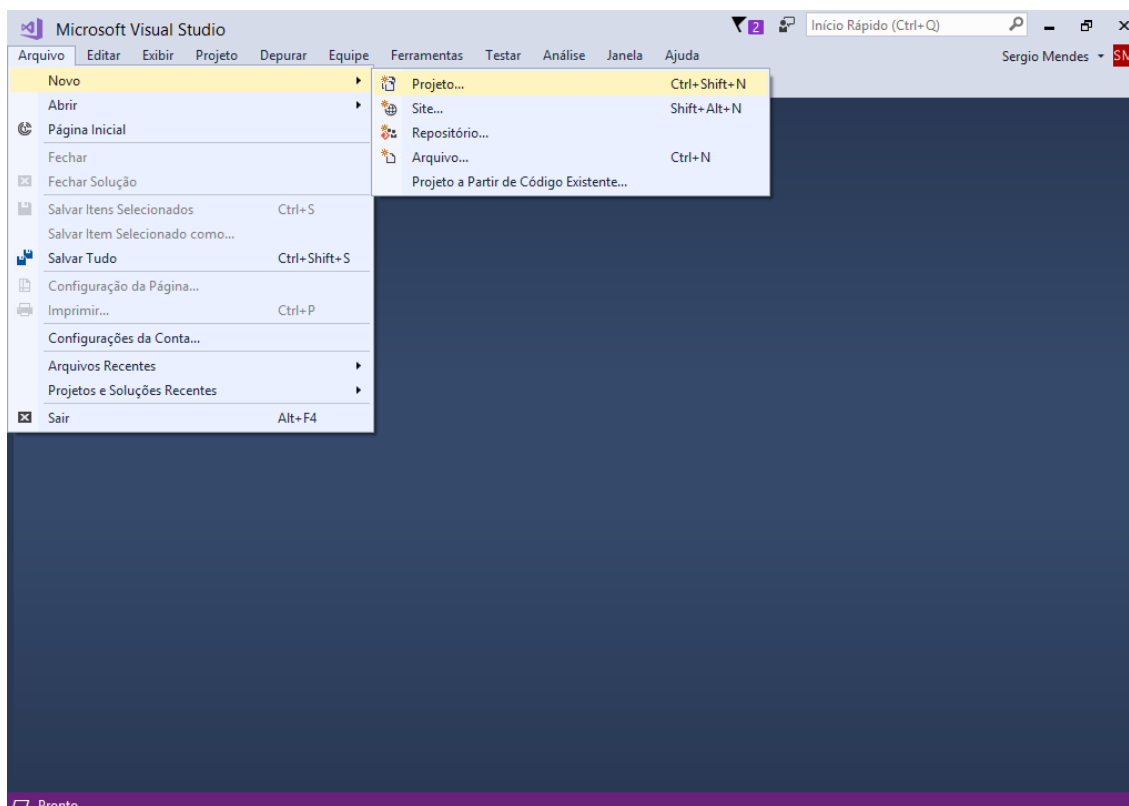
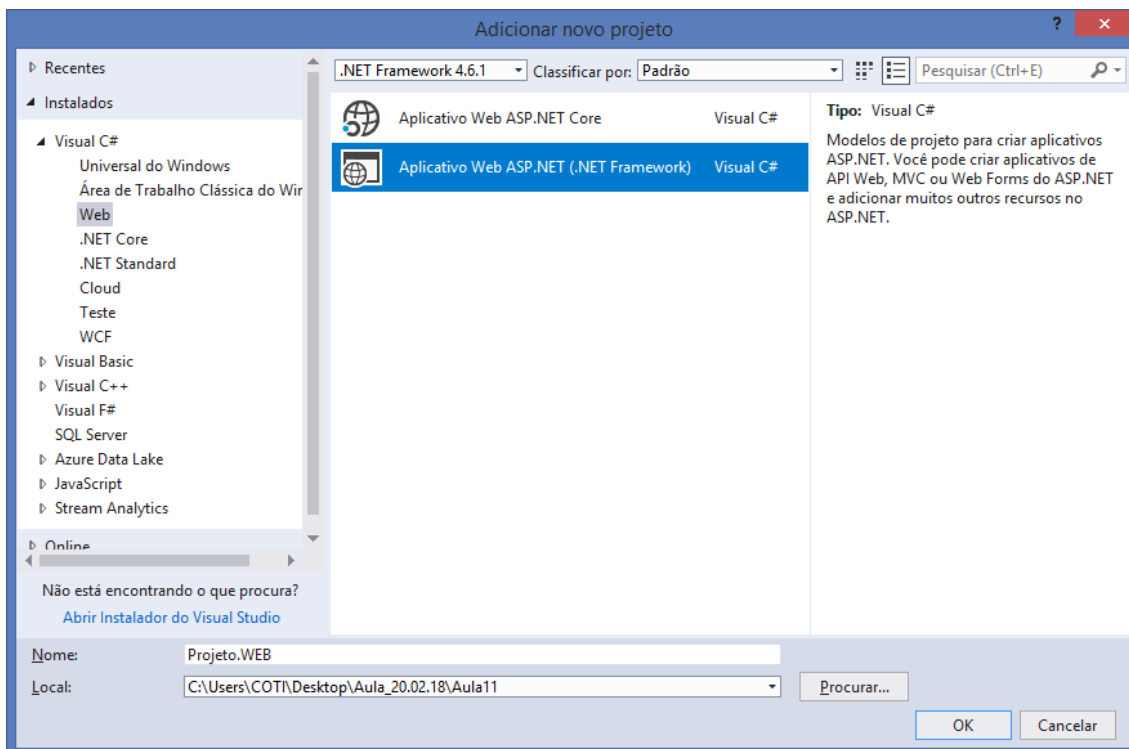
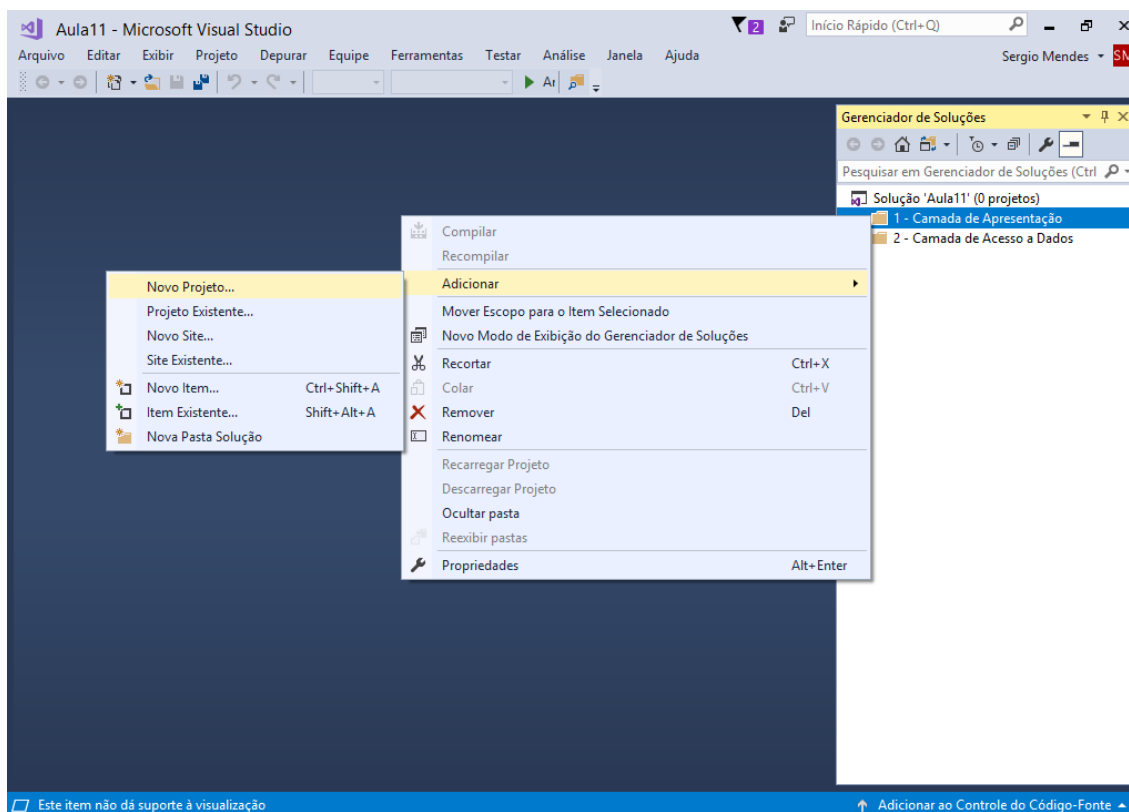


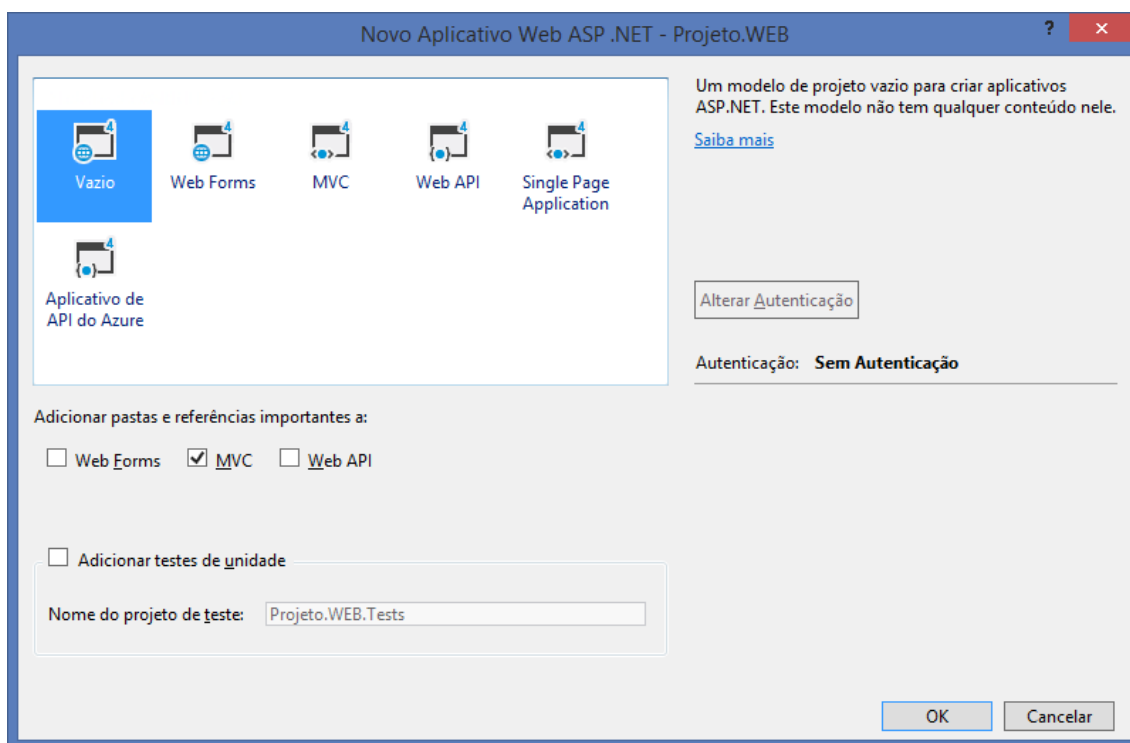
Criando uma solution:



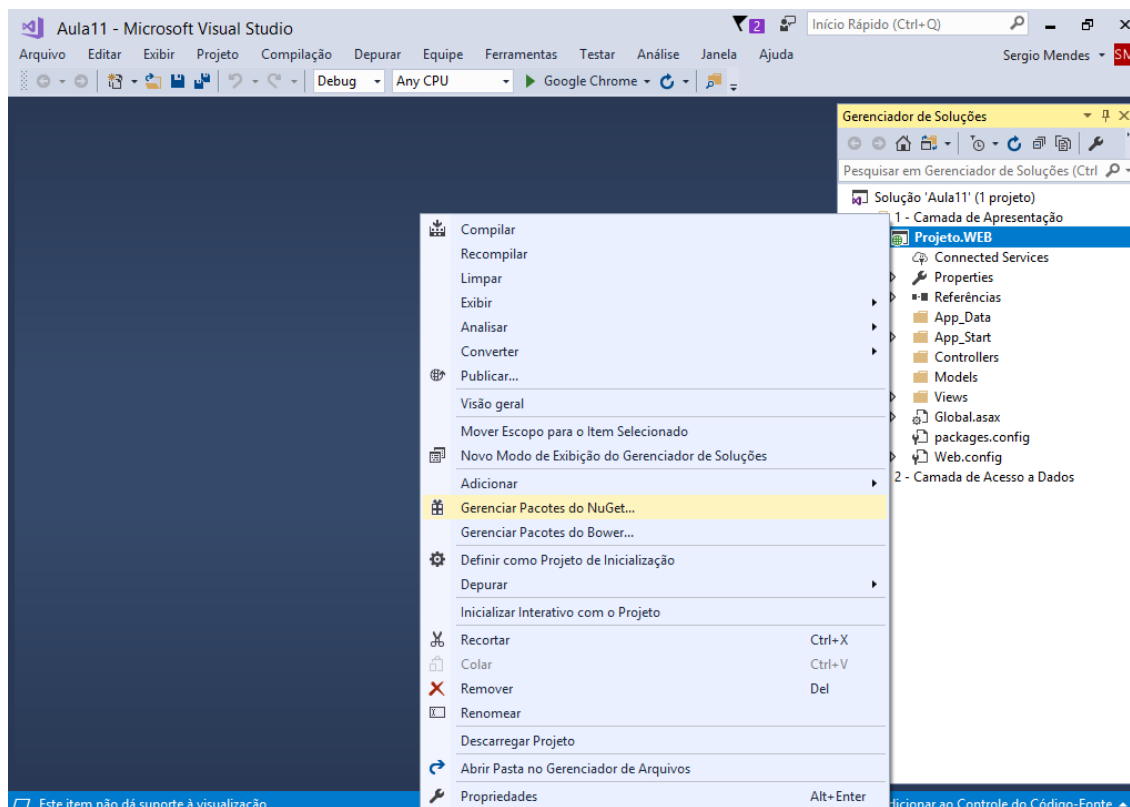
1 - Camada de Apresentação Asp.Net MVC WebApplication



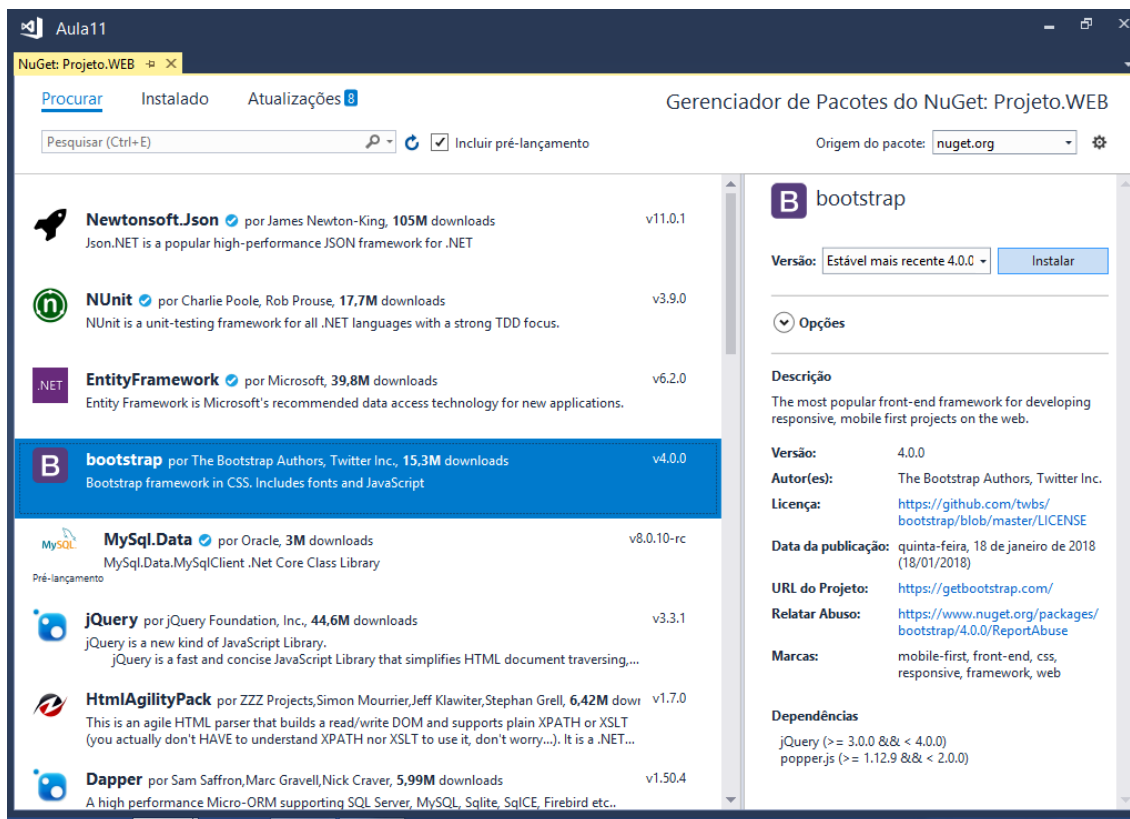
Tpo: Vazio / MVC



Instalando o bootstrap:
Gerenciador de pacotes do NuGet

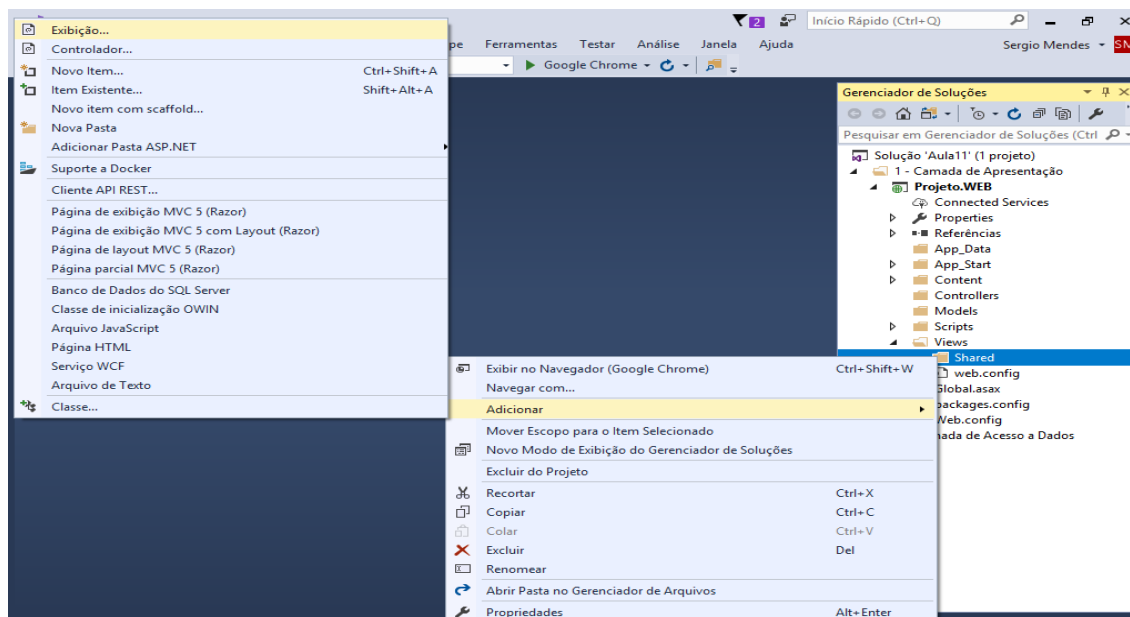


Selecione bootstrap 4.0



Master Page

Página mestra, ou seja, são páginas que contêm um layout padrão que poderá ser "herdado" por outras páginas do projeto. Por boa prática, as Master Pages são criadas em uma pasta dentro de \Views chamada de **Shared**



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

```
<!DOCTYPE html>

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

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

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

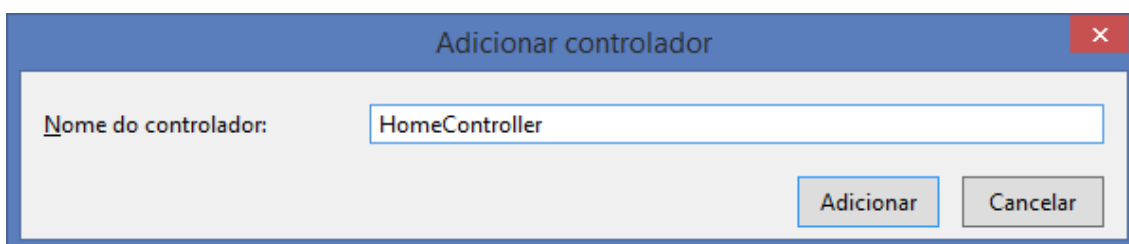
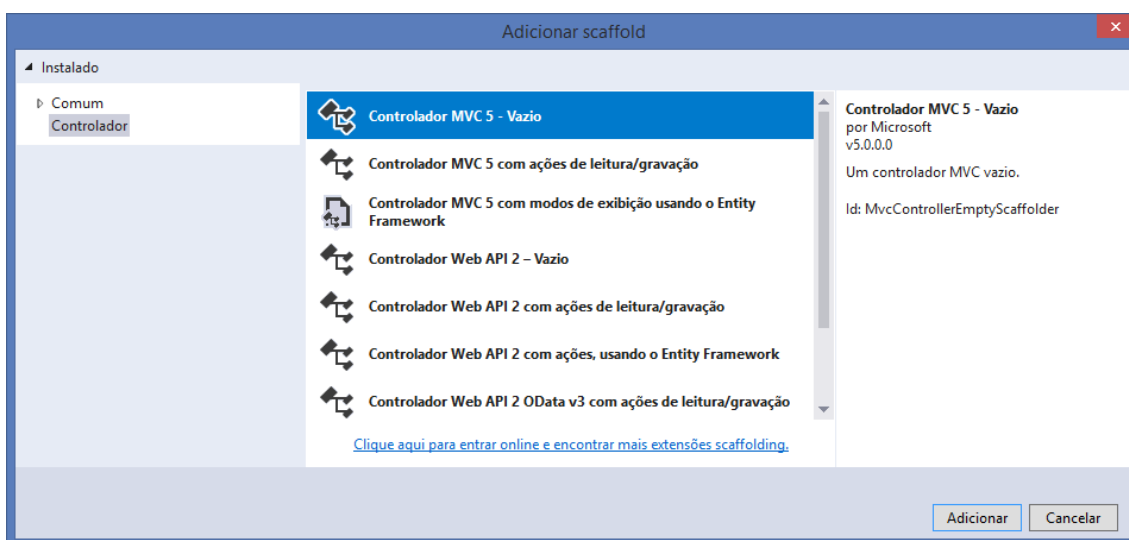
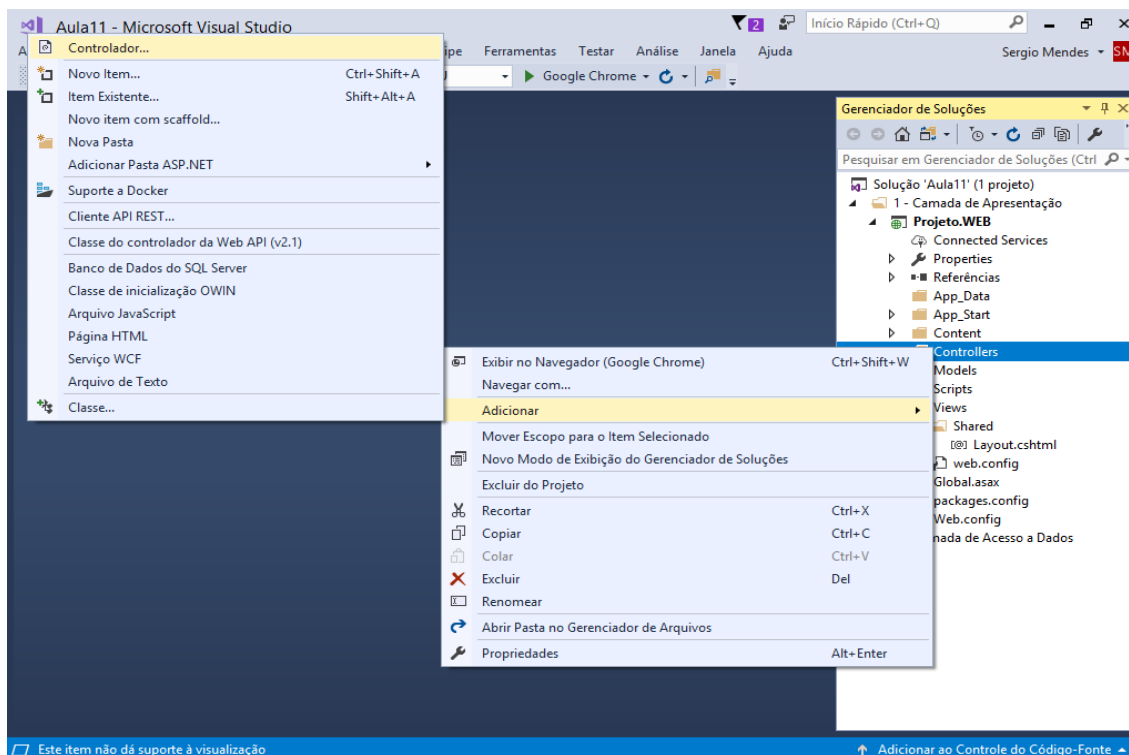
  <div class="card">
    <div class="card-body">
      <h3>Projeto de Controle de Estoques</h3>
      Aula de C# WebDeveloper - COTI Informática
    </div>
  </div>

  <!-- linha -->
  <div class="row">
    <!-- coluna tamanho 12 -->
    <div class="col-md-12">
      <!-- local para entrada do conteúdo das demais páginas.. -->
      @RenderBody()
    </div>
  </div>

</body>
</html>
```

Página inicial no projeto MVC:

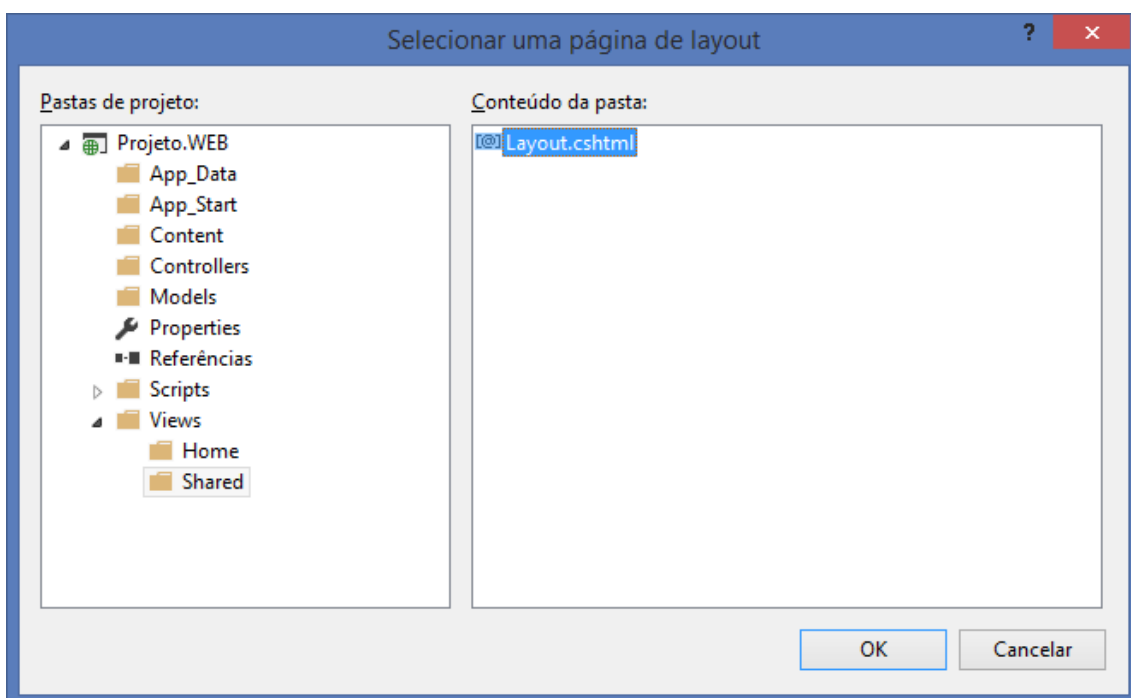
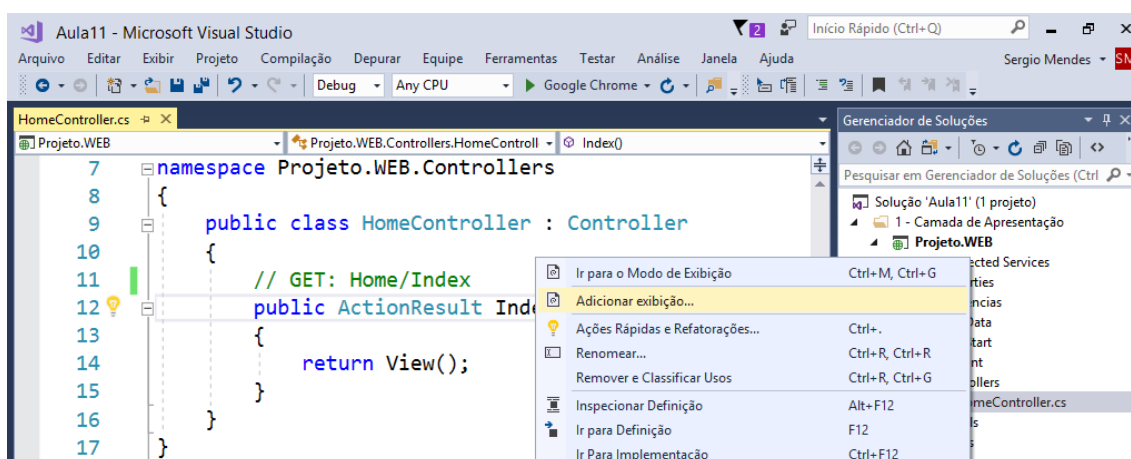
/Home/Index
[Controller] [View]



```
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();
        }
    }
}
```

Gerando a página Index:



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

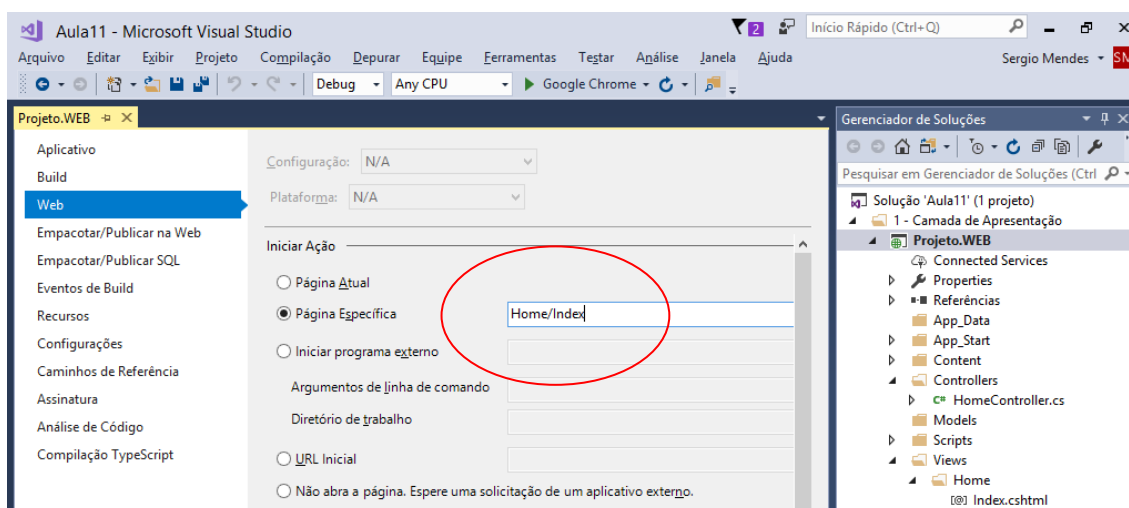
```
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/Layout.cshtml";
}
```

```
<h4>Seja bem vindo ao projeto</h4>
<hr/>
```

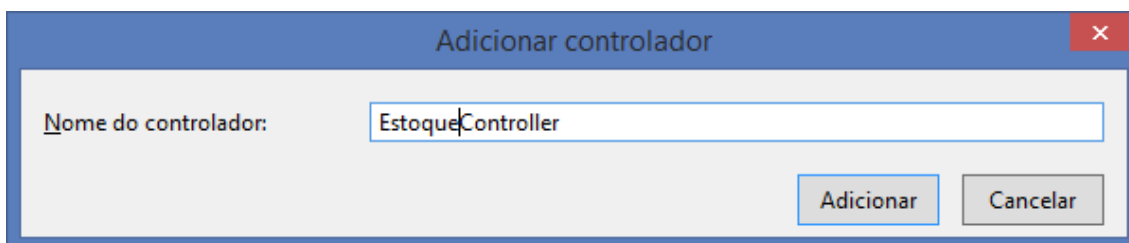
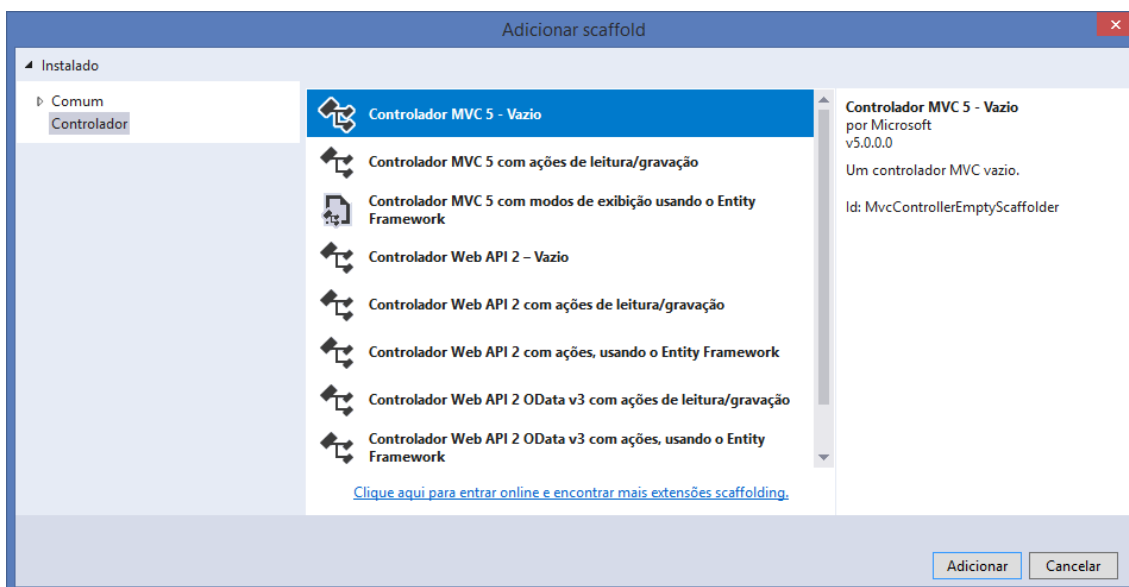
```
<div>
    Escolha a ação desejada:
</div>
```

```
<ul>
    <li> <a href="/Estoque/Cadastro">Cadastrar Estoques</a> </li>
    <li> <a href="/Estoque/Consulta">Consultar Estoques</a> </li>
</ul>
```

Definindo a página inicial do projeto:



<http://localhost:51496/Home/Index>



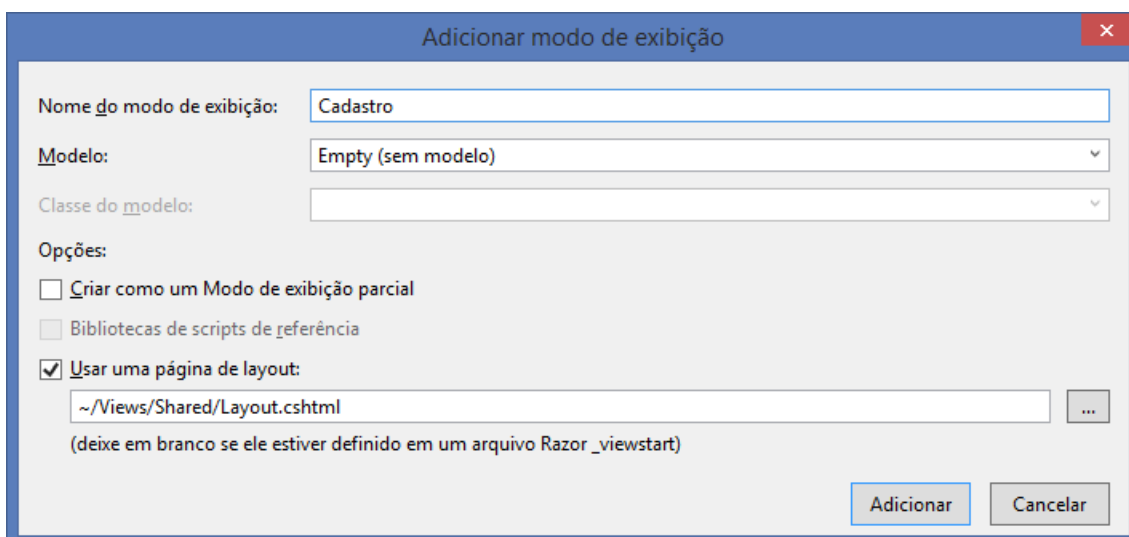
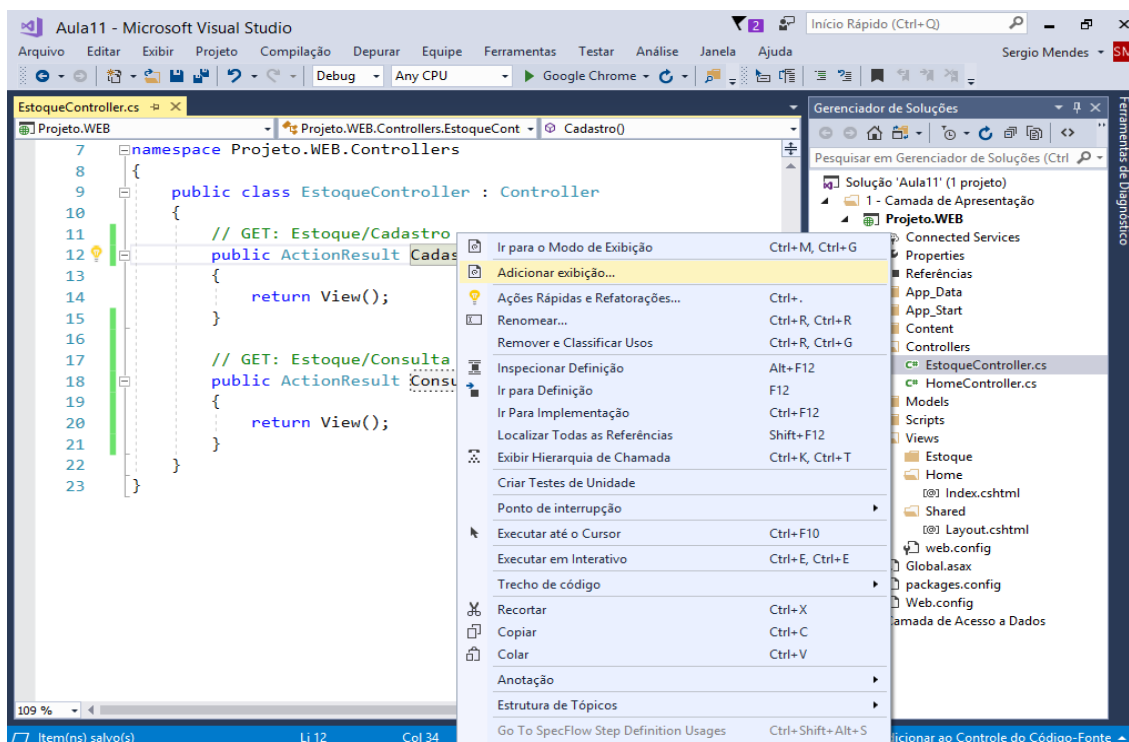
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Projeto.WEB.Controllers
{
    public class EstoqueController : Controller
    {
```

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

// GET: Estoque/Consulta
public ActionResult Consulta()
{
    return View();
}
}
```

Criando as páginas:



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

<h4>Cadastro de Estoque</h4>
<a href="/Home/Index">Página inicial</a>
<hr/>

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

        <label>Nome do Estoque:</label>
        <input type="text" id="nome" class="form-control"/>
        <br/>

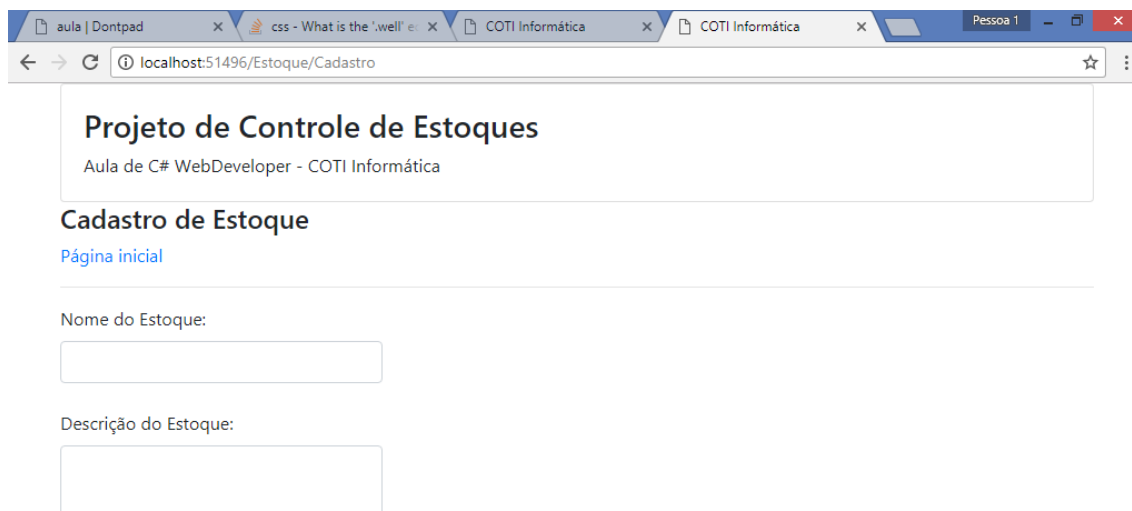
        <label>Descrição do Estoque:</label>
        <textarea id="descricao" class="form-control" rows="6"/>
        <br />

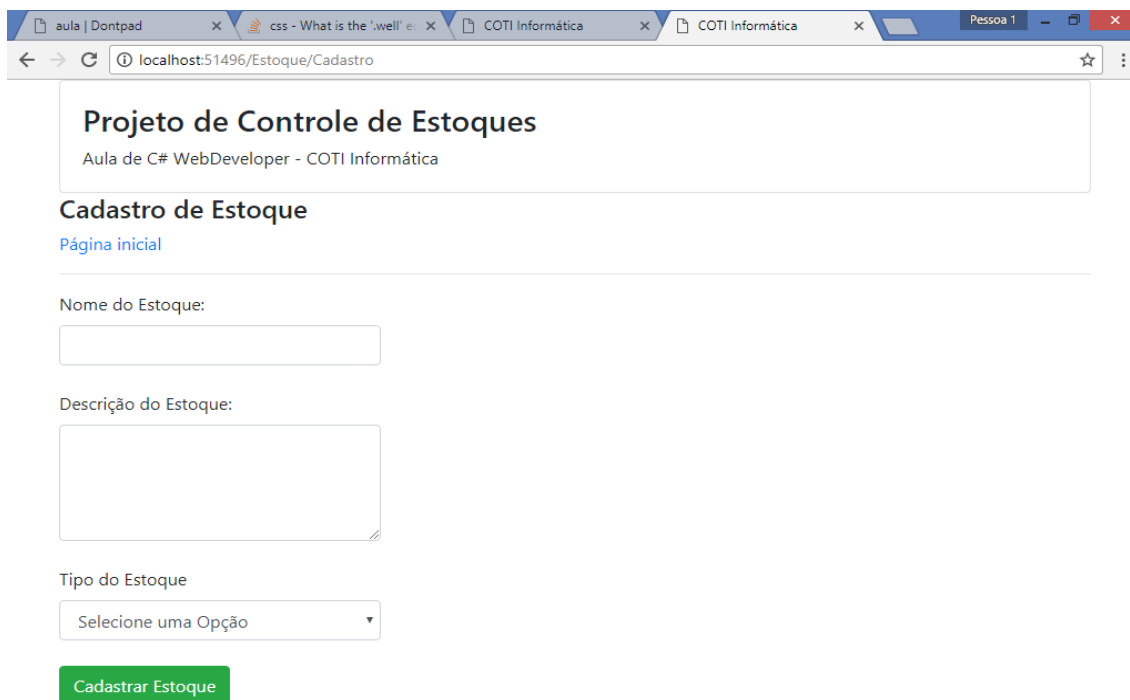
        <label>Tipo do Estoque</label>
        <select id="tipo" class="form-control">
            <option value="">Selecione uma Opção</option>
            <option value="1">Informática</option>
            <option value="2">Eletrônicos</option>
            <option value="3">Livraria</option>
            <option value="4">Games</option>
            <option value="5">Outros</option>
        </select>
        <br/>

        <button id="btncadastro" class="btn btn-success">
            Cadastrar Estoque
        </button>
        <br/>
        <br/>

        <div id="mensagem"></div>

    </div>
</div>
```





The screenshot shows a web browser window with the URL `localhost:51496/Estoque/Cadastro`. The page title is 'Projeto de Controle de Estoques' and the subtitle is 'Aula de C# WebDeveloper - COTI Informática'. Below the title is a link for 'Página inicial'. The form contains three input fields: 'Nome do Estoque:', 'Descrição do Estoque:', and 'Tipo do Estoque' (a dropdown menu with 'Selecione uma Opção'). A green button labeled 'Cadastrar Estoque' is at the bottom.

```
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>COTI Informática</title>
    <!-- folhas de estilo CSS -->
    <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
</head>
<body class="container">

    <div class="card">
        <div class="card-body">
            <h3>Projeto de Controle de Estoques</h3>
            Aula de C# WebDeveloper - COTI Informática
        </div>
    </div>

    <!-- linha -->
    <div class="row">
        <!-- coluna tamanho 12 -->
        <div class="col-md-12">
            <!-- local para entrada do conteúdo das demais páginas.. -->
            @RenderBody()
        </div>
    </div>

    <!-- Seção para a área javascript -->
    @RenderSection("scripts", required : false)
</body>
</html>
```

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

@section scripts{

    <!-- importar a biblioteca do jquery.. -->
    <script src="~/Scripts/jquery-3.0.0.min.js"></script>

    <script type="text/javascript">
        //evento executado quando a página for carregada..
        $(document).ready(function () {
            //criando um evento para quando o botão for clicado..
            $("#btncadastro").click(function () {

                //criando uma variavel JSON para resgatar
                //o conteudo de cada campo da página..
                var model = {
                    Nome: $("#nome").val(),
                    Descricao: $("#descricao").val(),
                    Tipo: $("#tipo").val()
                };

                //função AJAX para enviar os dados para o servidor..
                $.ajax({
                    type: "POST",
                    url: "/Estoque/CadastrarEstoque",
                    data: model,
                    success: function () {


                    },
                    error: function (e) {
                        //imprimir mensagem de erro..
                        $("#mensagem").html("Erro: " + e.status);
                    }
                });
            });
        });
    </script>
}

<h4>Cadastro de Estoque</h4>
<a href="/Home/Index">Página inicial</a>
<hr/>

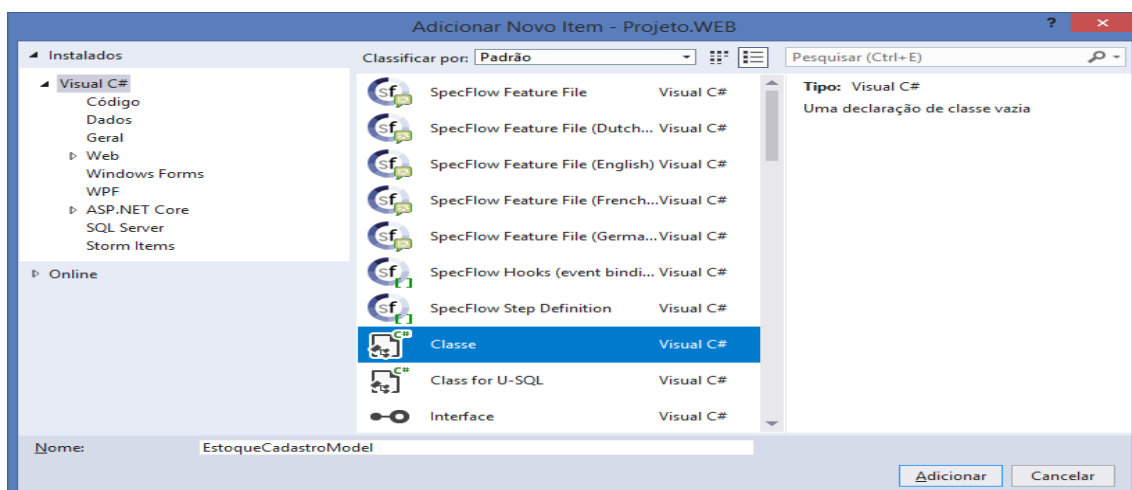
<div class="row">
    <div class="col-md-4">
        <label>Nome do Estoque:</label>
        <input type="text" id="nome" class="form-control"/>
        <br/>
        <label>Descrição do Estoque:</label>
        <textarea id="descricao" class="form-control" rows="4"></textarea>
        <br />
        <label>Tipo do Estoque</label>
        <select id="tipo" class="form-control">
            <option value="">Selecione uma Opção</option>
            <option value="1">Informática</option>
            <option value="2">Eletrônicos</option>
            <option value="3">Livraria</option>
            <option value="4">Games</option>
        </select>
    </div>
</div>
```

```
<option value="5">Outros</option>
</select>
<br/>
<button id="btncadastro" class="btn btn-success">
    Cadastrar Estoque
</button>
<br/>
<br/>
<div id="mensagem"></div>
</div>
</div>
```

Executando:



Criando uma classe de modelo para resgatar os dados enviados em formato JSON:



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

namespace Projeto.WEB.Models
{
    public class EstoqueCadastroModel
    {
        public string Nome { get; set; }
        public string Descricao { get; set; }
        public int Tipo { get; set; }
    }
}

//criando uma variavel JSON para resgatar
//o conteudo de cada campo da página..
var model = {
    Nome: $("#nome").val(),
    Descricao: $("#descricao").val(),
    Tipo: $("#tipo").val()
};
```

Na classe de controle:
/EstoqueController.cs

JsonResult

Utilizado para métodos que recebem ou retornam requisições baseadas em AJAX que manipulam dados em formato JSON.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Projeto.WEB.Models; //camada de modelo..

namespace Projeto.WEB.Controllers
{
    public class EstoqueController : Controller
    {
        public ActionResult Cadastro()
        {
            return View();
        }

        public ActionResult Consulta()
        {
            return View();
        }

        public JsonResult CadastrarEstoque(EstoqueCadastroModel model)
        {
            return Json("Estoque cadastrado com sucesso.");
        }
    }
}
```

Voltando ao jquery:

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

@section scripts{

    <!-- importar a biblioteca do jquery.. -->
    <script src="~/Scripts/jquery-3.0.0.min.js"></script>

    <script type="text/javascript">
        //evento executado quando a página for carregada..
        $(document).ready(function () {
            //criando um evento para quando o botão for clicado..
            $("#btncadastro").click(function () {

                //criando uma variavel JSON para resgatar
                //o conteudo de cada campo da página..
                var model = {
                    Nome: $("#nome").val(),
                    Descricao: $("#descricao").val(),
                    Tipo: $("#tipo").val()
                };

                //função AJAX para enviar os dados para o servidor..
                $.ajax({
                    type: "POST",
                    url: "/Estoque/CadastrarEstoque",
                    data: model,
                    success: function (msg) {

                        //imprimir a mensagem..
                        $("#mensagem").html(msg);

                        //limpar o conteudo de cada campo..
                        $("#nome").val("");
                        $("#descricao").val("");
                        $("#tipo").val("");
                    },
                    error: function (e) {
                        //imprimir mensagem de erro..
                        $("#mensagem").html("Erro: " + e.status);
                    }
                });
            });
        });
    </script>
}

<h4>Cadastro de Estoque</h4>
<a href="/Home/Index">Página inicial</a>
<hr/>

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



```
<label>Nome do Estoque:</label>
<input type="text" id="nome" class="form-control"/>
<br/>

<label>Descrição do Estoque:</label>
<textarea id="descricao" class="form-control" rows="4"></textarea>
<br />

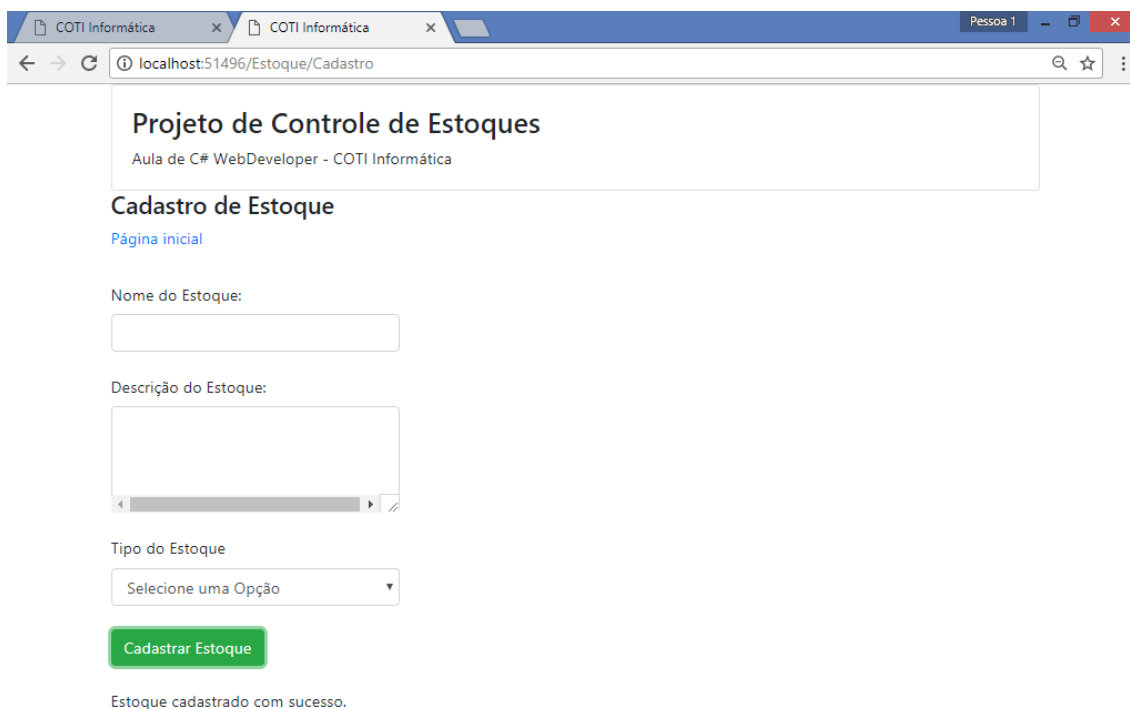
<label>Tipo do Estoque</label>
<select id="tipo" class="form-control">
  <option value="">Selecione uma Opção</option>
  <option value="1">Informática</option>
  <option value="2">Eletrônicos</option>
  <option value="3">Livraria</option>
  <option value="4">Games</option>
  <option value="5">Outros</option>
</select>
<br/>

<button id="btncadastro" class="btn btn-success">
  Cadastrar Estoque
</button>
<br/>
<br/>

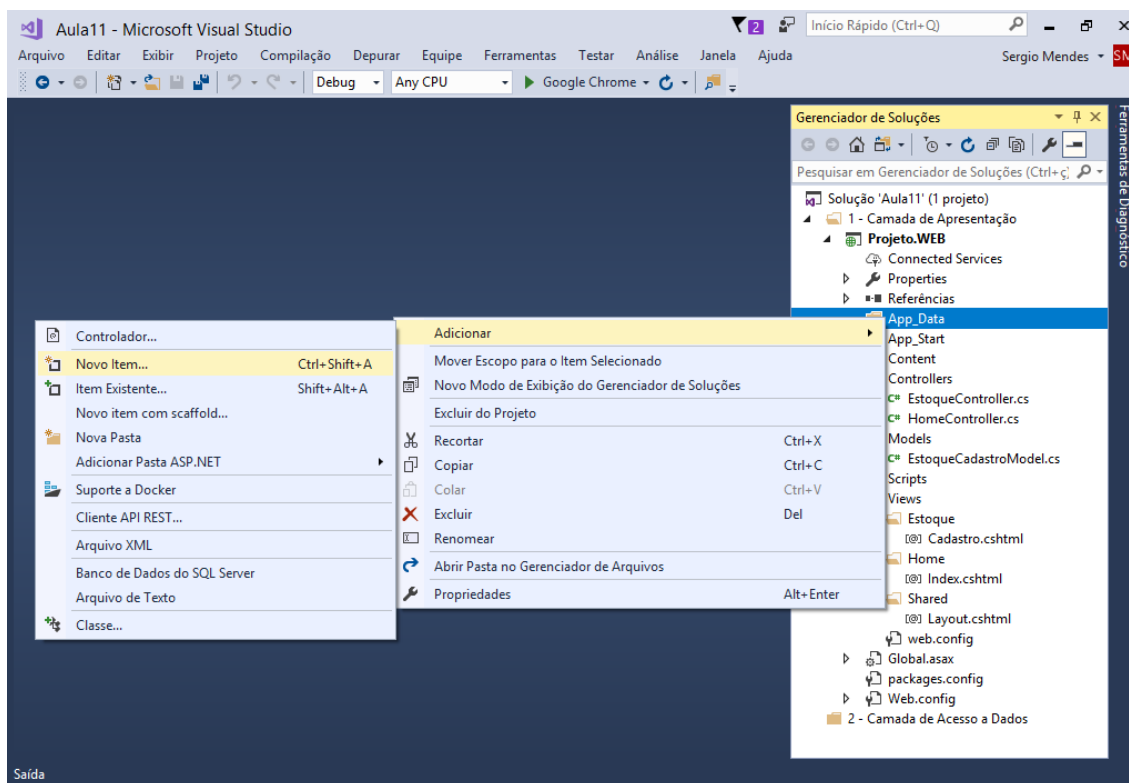
<div id="mensagem"></div>

</div>
</div>
```

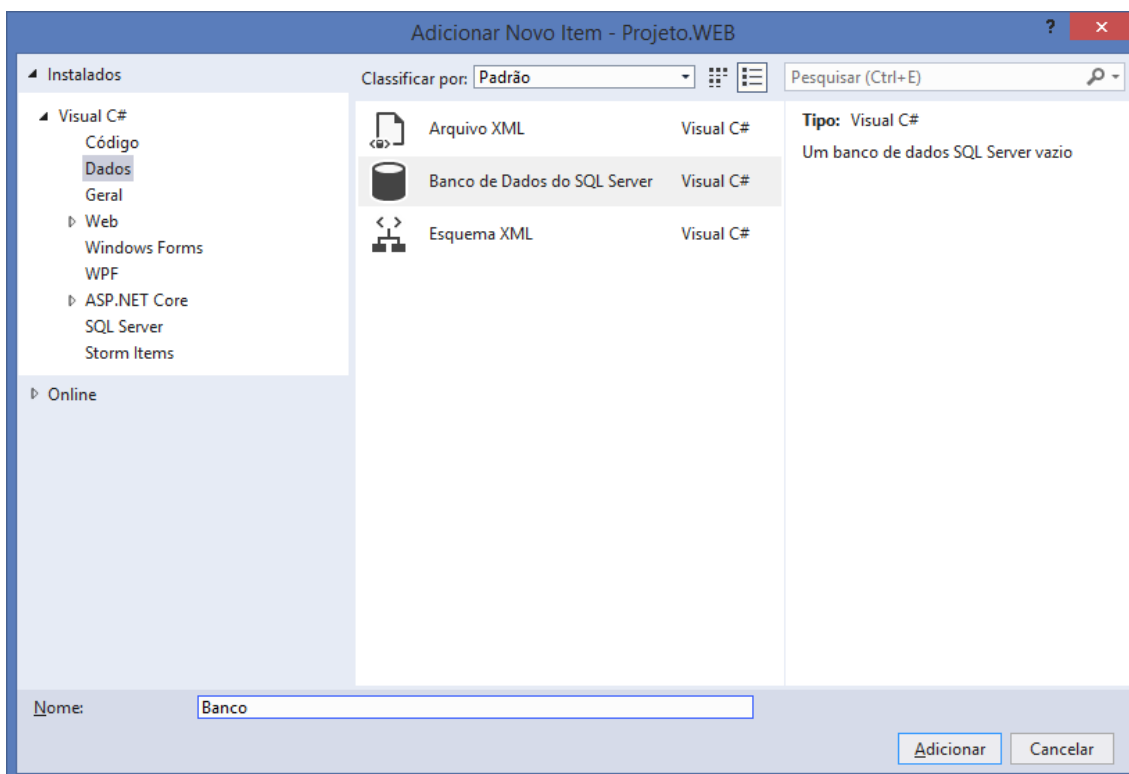
Executando:



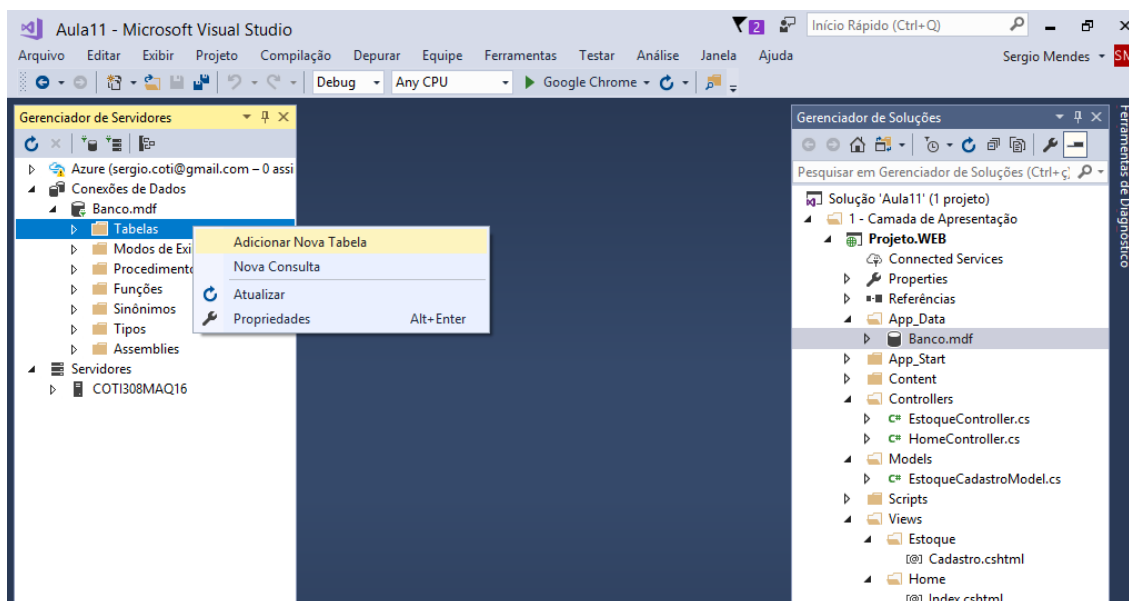
Criando a base de dados do projeto:



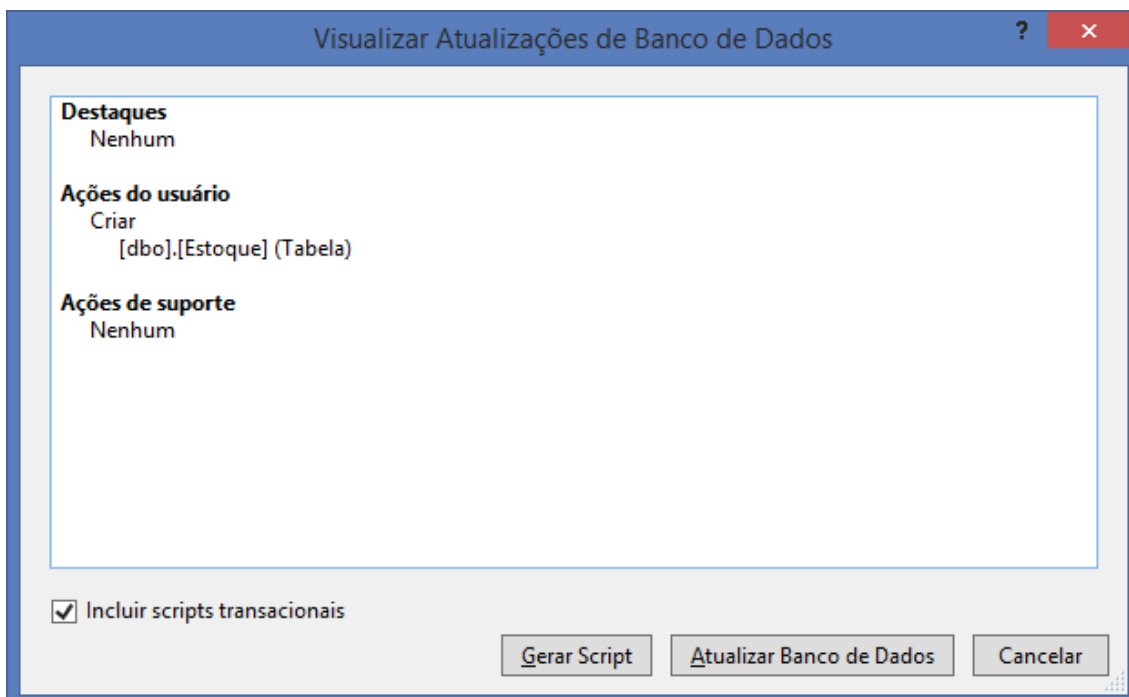
SqlServer Database (Banco de dados do SqlServer)



Criando a tabela de estoque:

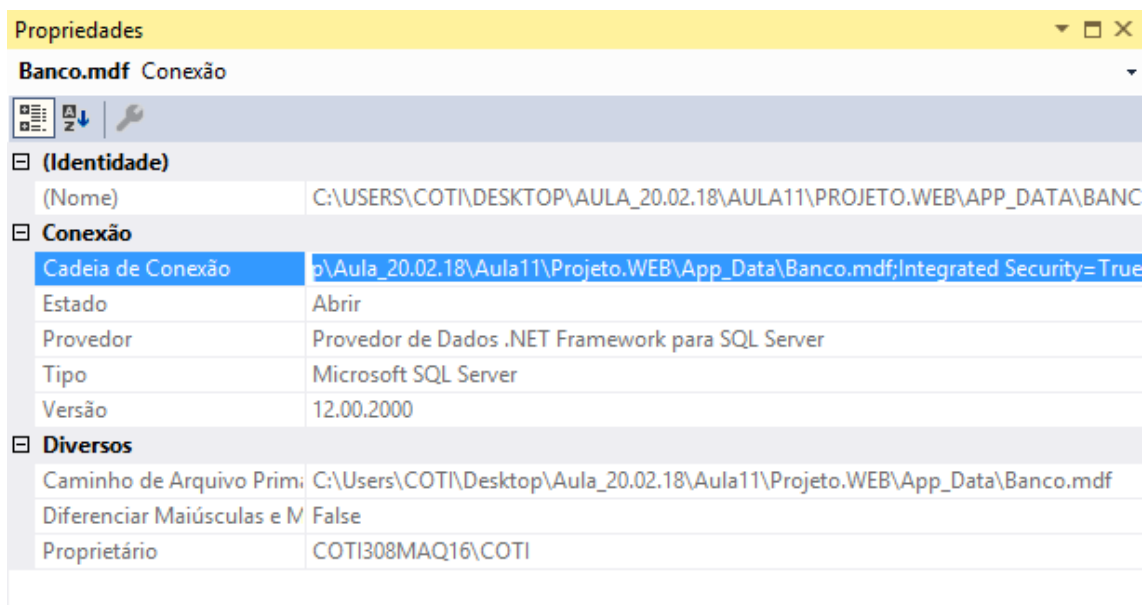


```
create table Estoque(
    IdEstoque      integer      identity,
    Nome           nvarchar(50) not null,
    Descricao     nvarchar(250) not null,
    Tipo          int          not null,
    primary key(IdEstoque))
go
```



\Web.config.xml

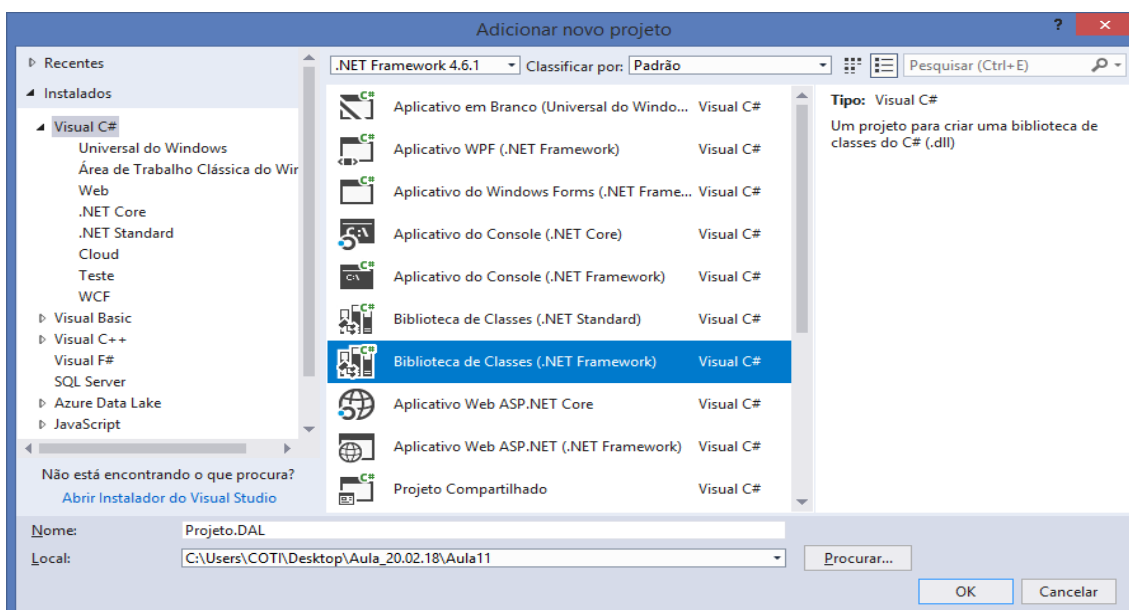
Mapeamento da connectionstring do projeto.



```
<connectionStrings>
  <add
    name="aula11"
    connectionString="Data Source=(LocalDB)\MSSQLLocalDB;
    AttachDbFilename=C:\Users\COTI\Desktop\Aula_20.02.18\Aula11\
    Projeto.WEB\App_Data\Banco.mdf; Integrated Security=True"
  />
</connectionStrings>
```

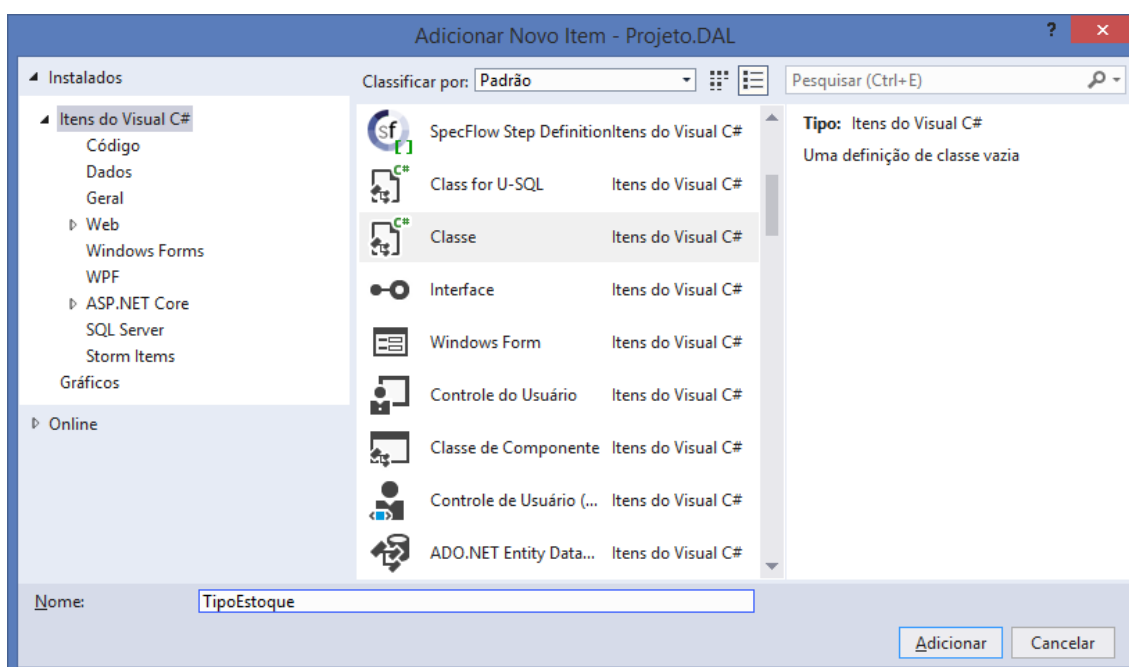
2 - Camada de Acesso a dados

DAL - Data Access Layer



Modelagem do Sistema:

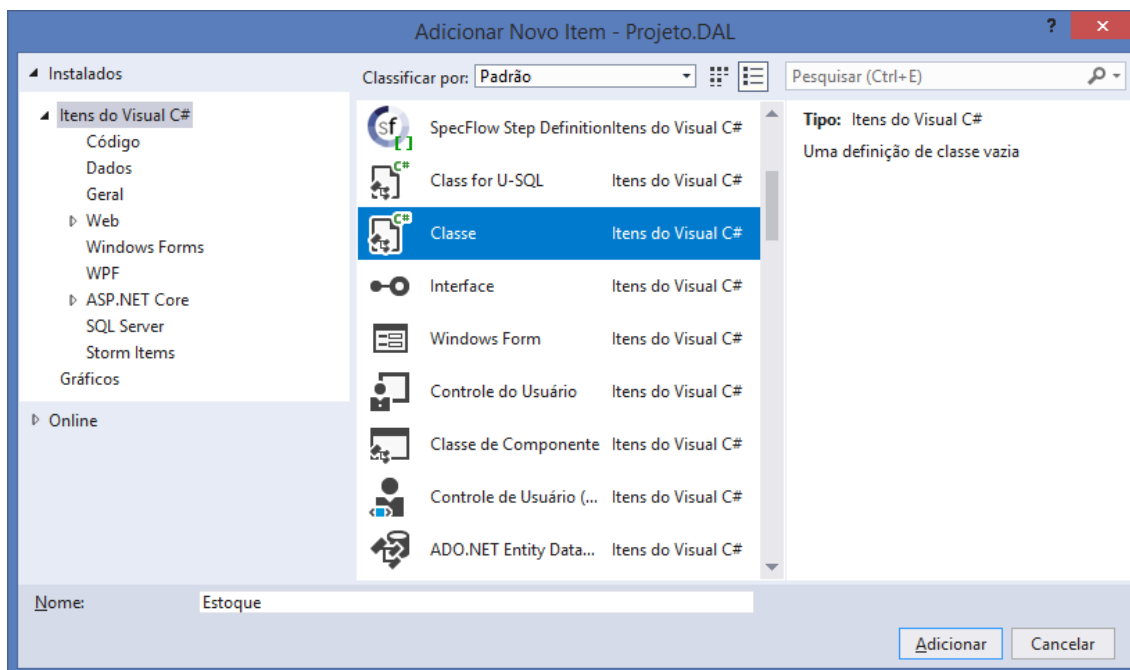
Diagrama de Classes



```

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

namespace Projeto.DAL.Entidades.Tipos
{
    public enum TipoEstoque
    {
        Informatica = 1,
        Eletronicos = 2,
        Livraria = 3,
        Games = 4,
        Outros = 5
    }
}
  
```



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Projeto.DAL.Entidades.Tipos; //importando..

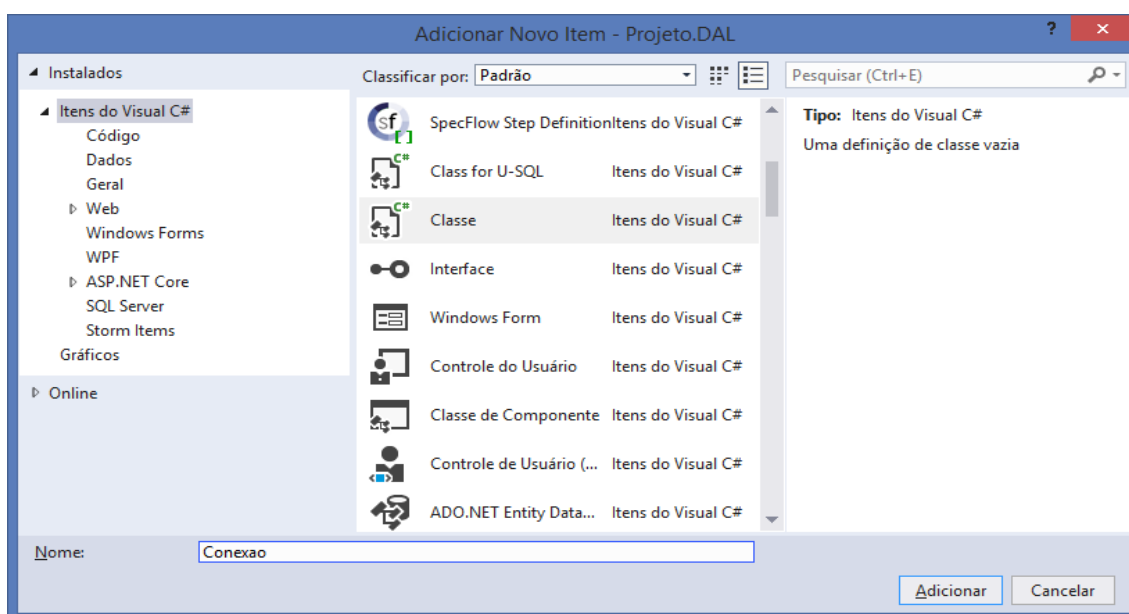
namespace Projeto.DAL.Entidades
{
    public class Estoque
    {
        //propriedades..
        public int IdEstoque { get; set; }
        public string Nome { get; set; }
        public string Descricao { get; set; }
        public TipoEstoque Tipo { get; set; }

        //construtor default..
        public Estoque()
        {
            //[ctor] + 2x[tab]
        }

        //sobrecarga de método (overloading)
        public Estoque(int idEstoque, string nome,
            string descricao, TipoEstoque tipo)
        {
            IdEstoque = idEstoque;
            Nome = nome;
            Descricao = descricao;
            Tipo = tipo;
        }
    }
}
```

```
//sobrescrita de método (override)
public override string ToString()
{
    return $"Id do Estoque: {IdEstoque}, Nome: {Nome},
        Descrição: {Descricao}, Tipo: {Tipo}";
}
}
}
```

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.Data.SqlClient; //importando..

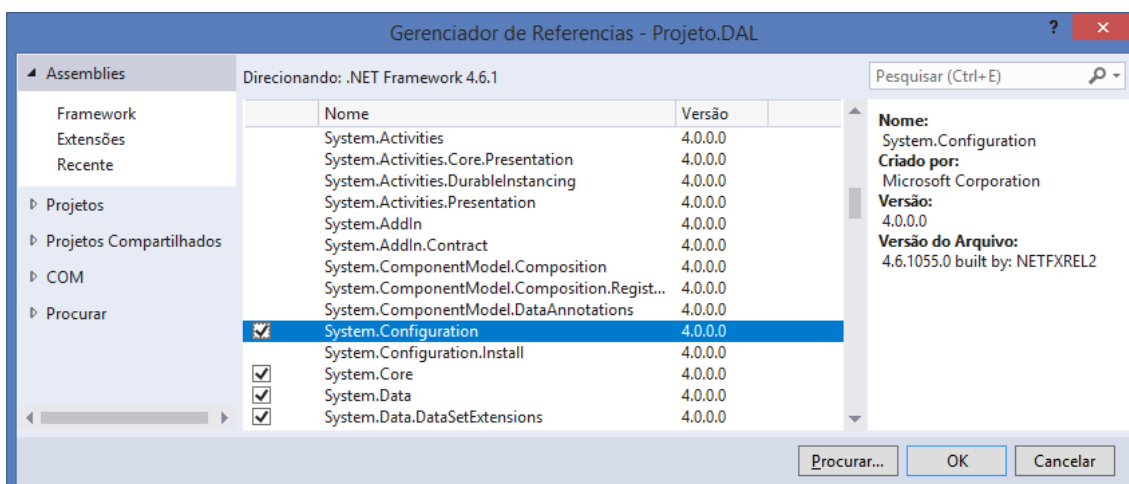
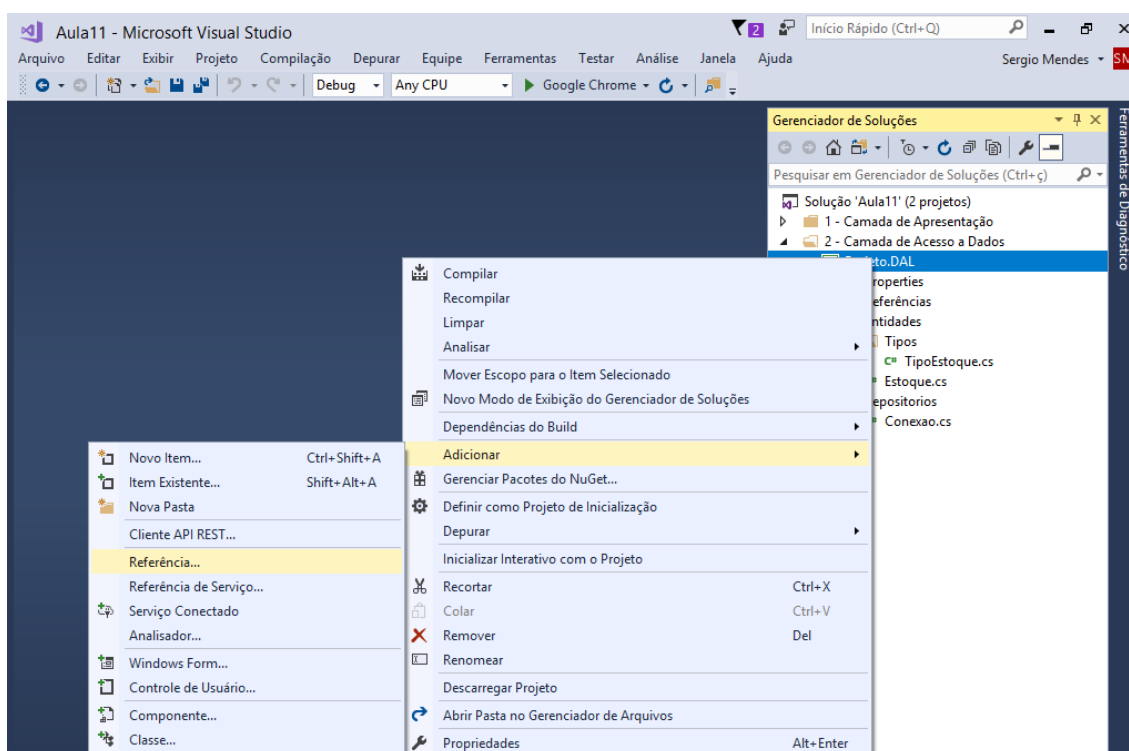
namespace Projeto.DAL.Repositorios
{
    public class Conexao
    {
        //atributos..
        protected SqlConnection con;
        protected SqlCommand cmd;
        protected SqlDataReader dr;
        protected SqlTransaction tr;

        //método para abrir conexão com o banco de dados..
        protected void OpenConnection()
        {
            con = new SqlConnection();
            con.Open(); //conectado..
        }
    }
}
```

```
//método para fechar a conexão com o banco de dados..
protected void CloseConnection()
{
    con.Close();
}
}
```

Obtendo o endereço da connectionstring:

** Incluindo no projeto uma referencia para System.Configuration



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



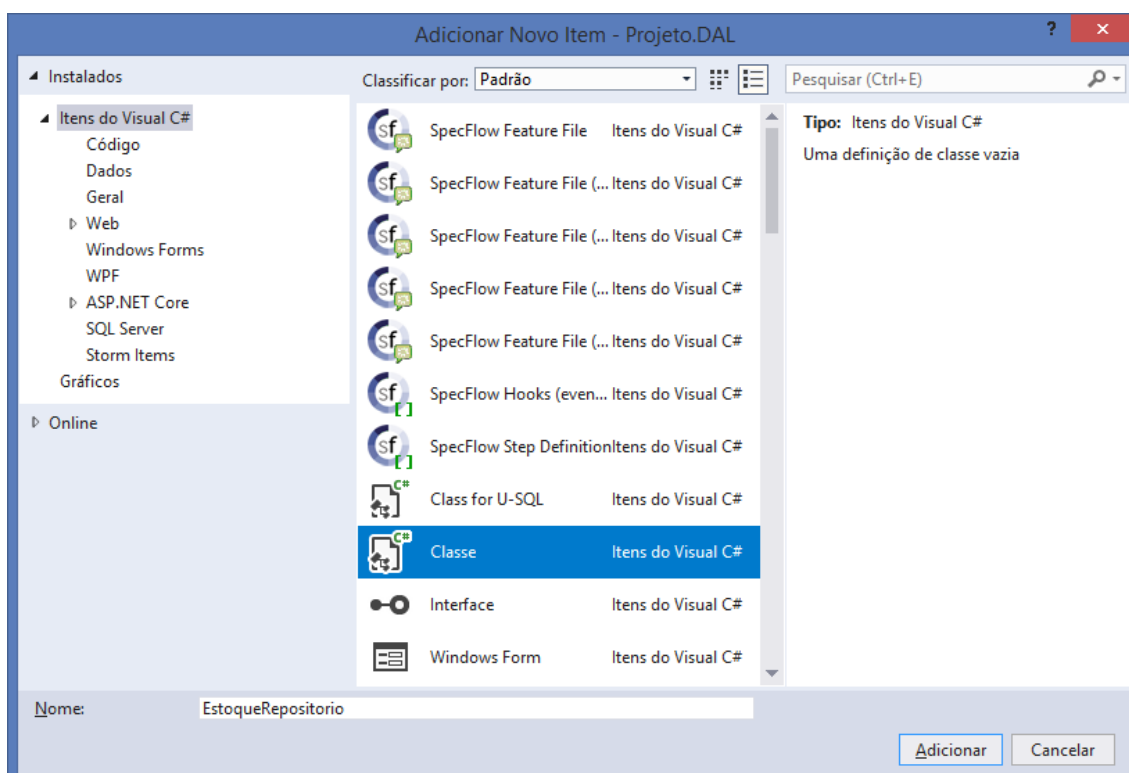
```
using System.Data.SqlClient; //importando..
using System.Configuration; //importando..

namespace Projeto.DAL.Repositorios
{
    public class Conexao
    {
        //atributos..
        protected SqlConnection con;
        protected SqlCommand cmd;
        protected SqlDataReader dr;
        protected SqlTransaction tr;

        //método para abrir conexão com o banco de dados..
        protected void OpenConnection()
        {
            con = new SqlConnection(ConfigurationManager.ConnectionStrings
                ["aula11"].ConnectionString);
            con.Open(); //conectado..
        }

        //método para fechar a conexão com o banco de dados..
        protected void CloseConnection()
        {
            con.Close();
        }
    }
}
```

Criando uma classe para persistir os dados de Estoque na base de dados (**Inserir, Alterar, Excluir e Consultar - CRUD**)



```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient; //sqlserver..
using Projeto.DAL.Entidades; //importando..
using Projeto.DAL.Entidades.Tipos; //importando..

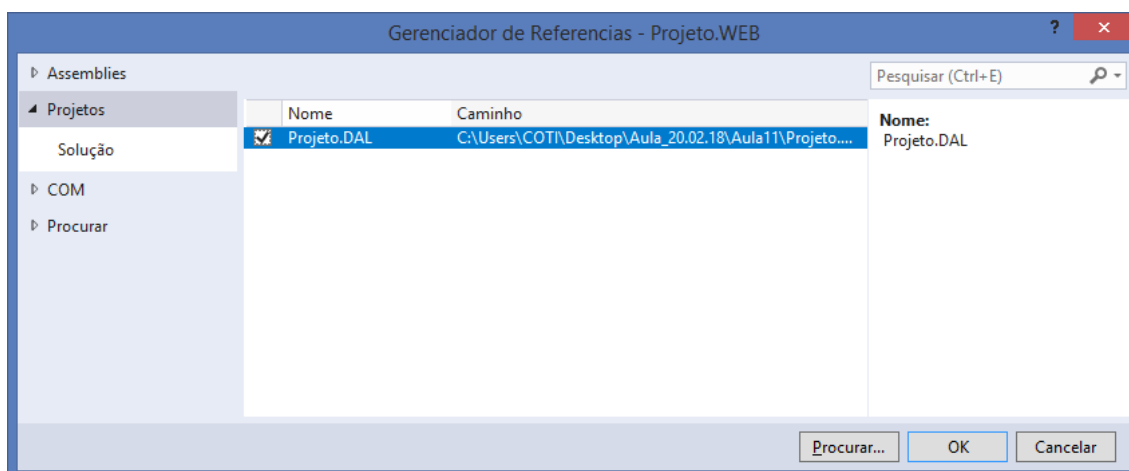
namespace Projeto.DAL.Repositorios
{
    public class EstoqueRepositorio : Conexao
    {
        //método para inserir um estoque na base de dados..
        public void Insert(Estoque e)
        {
            //abrir conexão com o banco de dados..
            OpenConnection();

            //executar uma operação (query) no banco de dados..
            string query = "insert into Estoque(Nome, Descricao, Tipo) "
                + "values(@Nome, @Descricao, @Tipo)";

            //executando no banco de dados..
            cmd = new SqlCommand(query, con);
            cmd.Parameters.AddWithValue("@Nome", e.Nome);
            cmd.Parameters.AddWithValue("@Descricao", e.Descricao);
            cmd.Parameters.AddWithValue("@Tipo", (int) e.Tipo);
            cmd.ExecuteNonQuery(); //executando..

            //fechar conexão com o banco de dados..
            CloseConnection();
        }
    }
}
```

Adicionando referencia no projeto WEB para o projeto DAL:



Voltando ao /EstoqueController:

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

```
using System.Web;
using System.Web.Mvc;
using Projeto.WEB.Models; //camada de modelo..
using Projeto.DAL.Entidades; //importando..
using Projeto.DAL.Repositorios; //importando..
using Projeto.DAL.Entidades.Tipos;

namespace Projeto.WEB.Controllers
{
    public class EstoqueController : Controller
    {
        // GET: Estoque/Cadastro
        public ActionResult Cadastro()
        {
            return View();
        }

        // GET: Estoque/Consulta
        public ActionResult Consulta()
        {
            return View();
        }

        public JsonResult CadastrarEstoque(EstoqueCadastroModel model)
        {
            try
            {
                //criando um objeto da classe Estoque..
                Estoque e = new Estoque();
                e.Nome = model.Nome;
                e.Descricao = model.Descricao;
                e.Tipo = (TipoEstoque) model.Tipo;

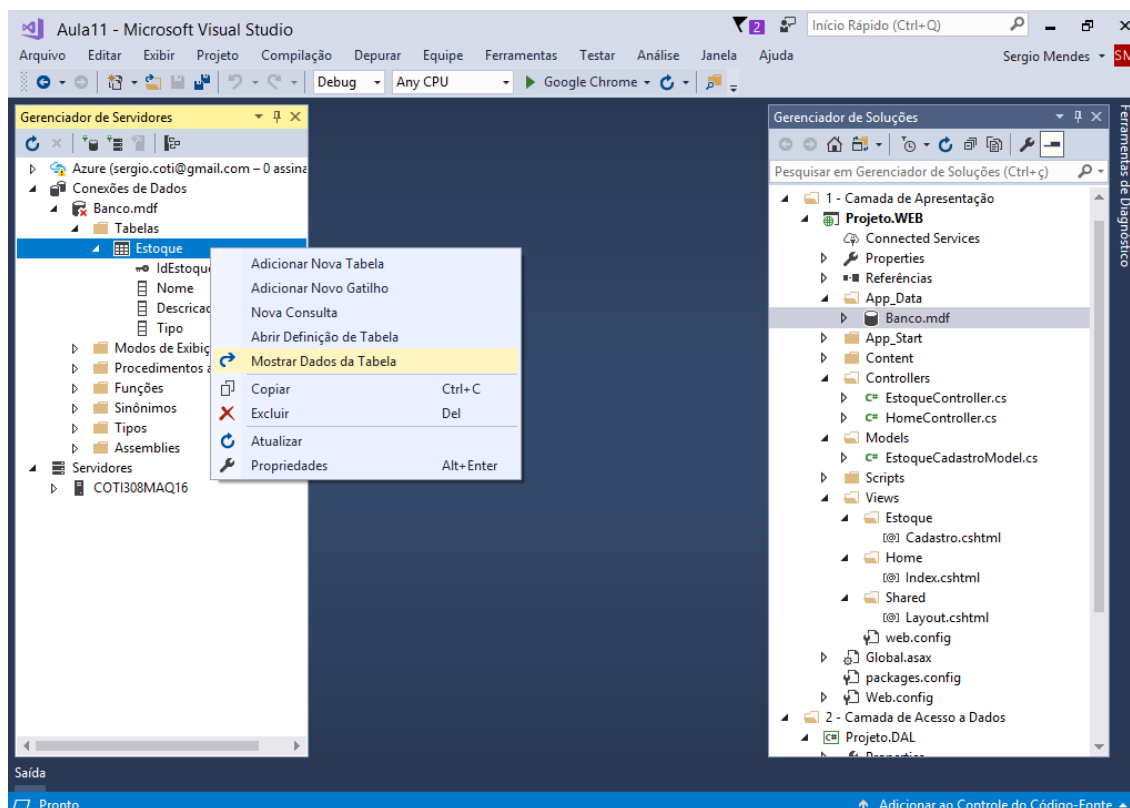
                //gravar na base de dados..
                EstoqueRepositorio rep = new EstoqueRepositorio();
                rep.Insert(e);

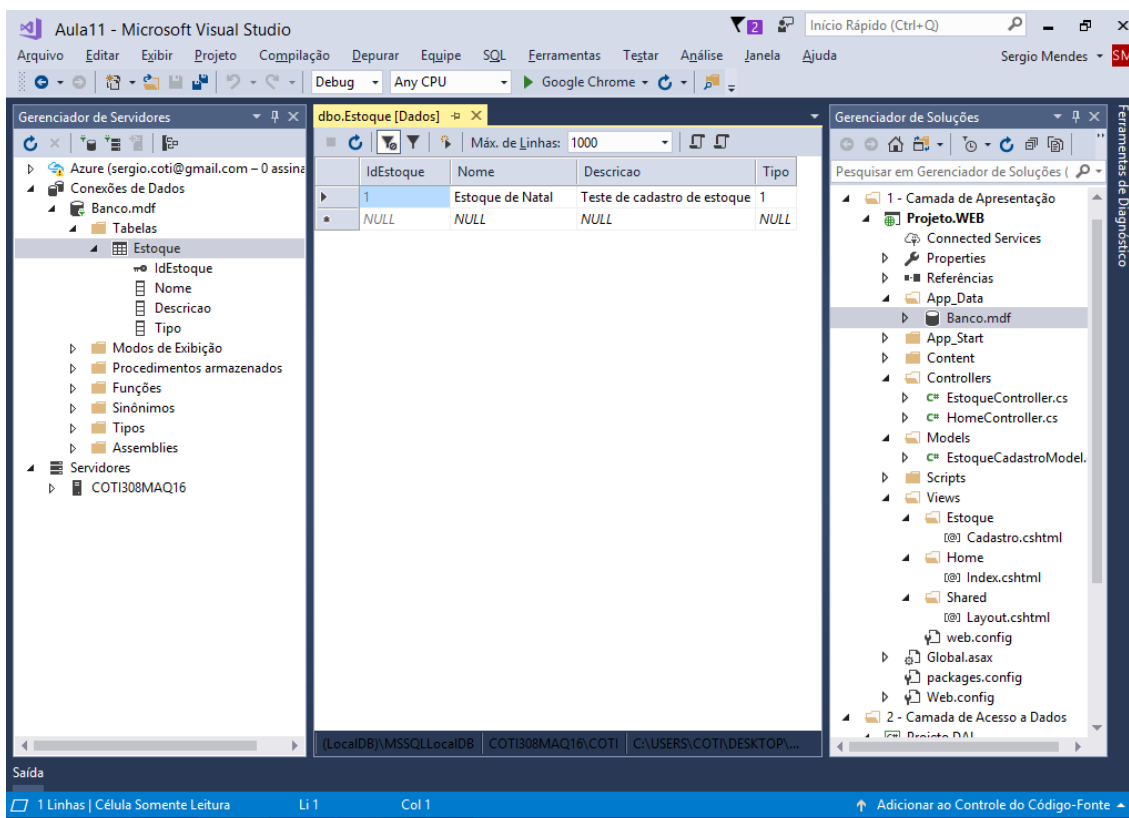
                return Json("Estoque cadastrado com sucesso.");
            }
            catch (Exception e)
            {
                return Json(e.Message);
            }
        }
    }
}
```

Executando:

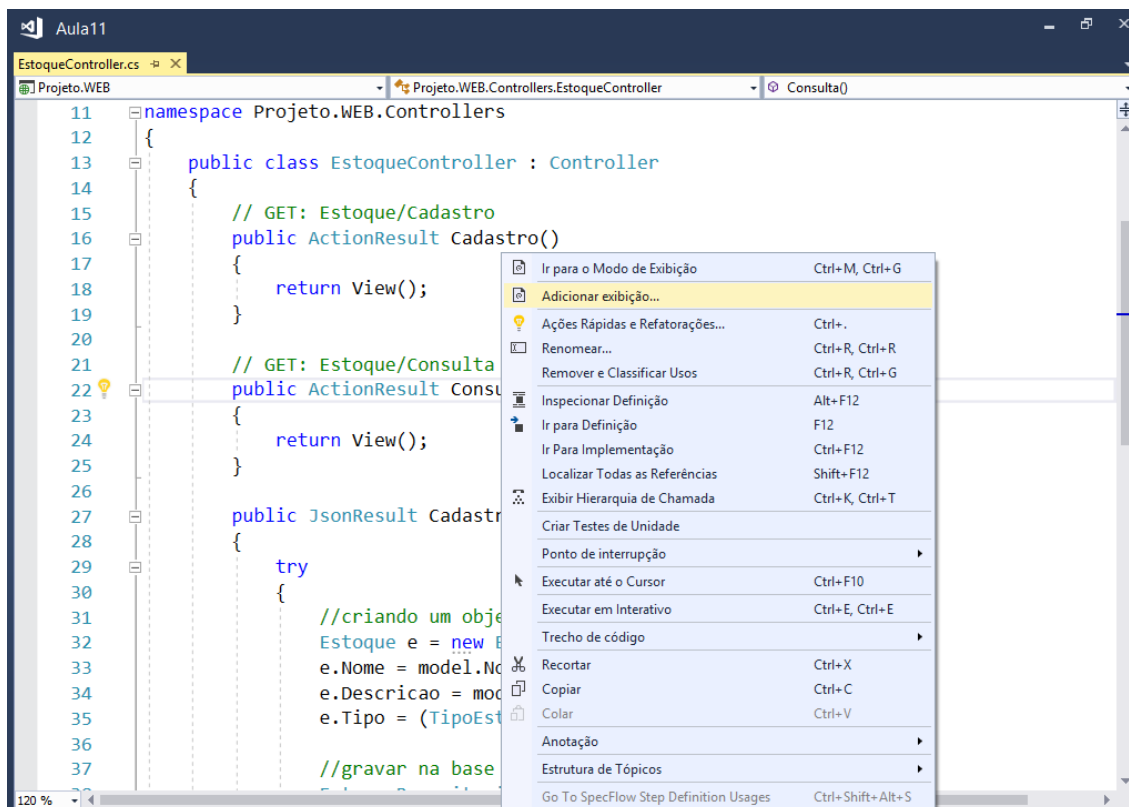


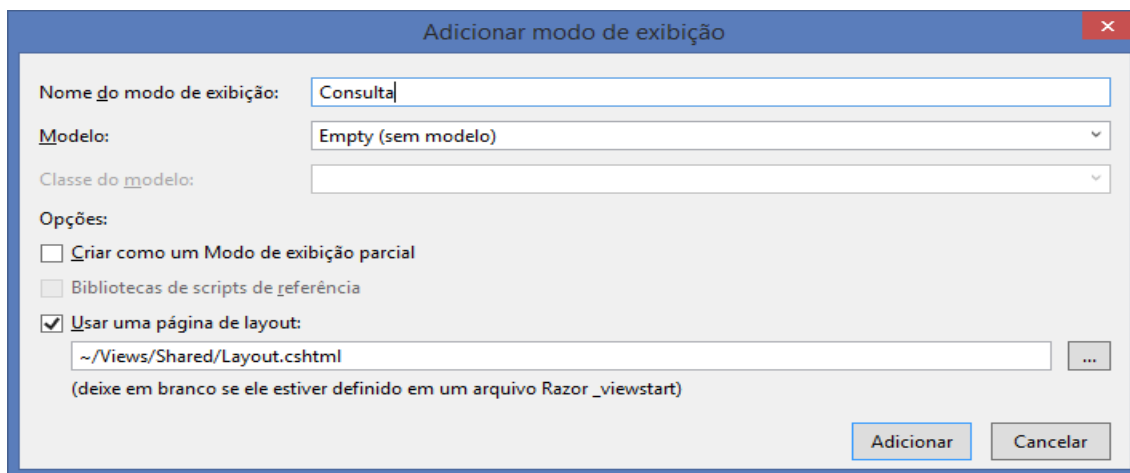
No banco de dados:





Criando a página de consulta:





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

@section scripts{

    <script src="~/Scripts/jquery-3.0.0.min.js"></script>

    <script>

        $(document).ready(function () {

        });

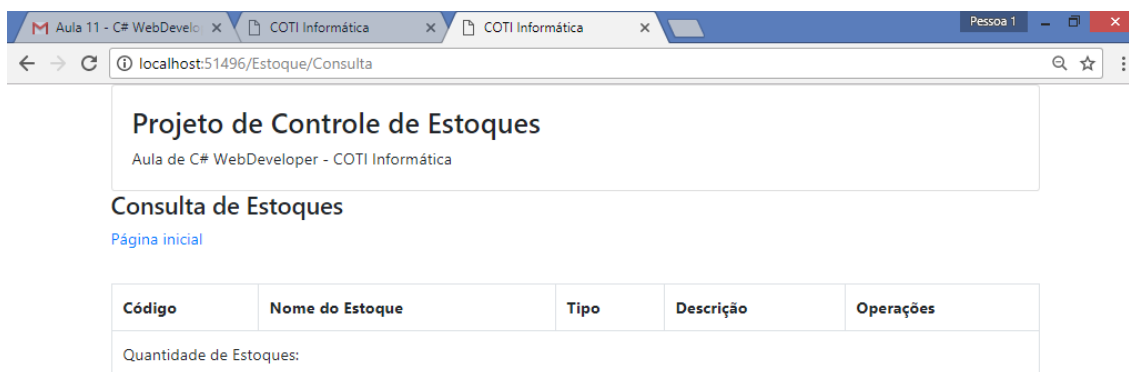
    </script>

}

<h4>Consulta de Estoques</h4>
<a href="/Home/Index">Página inicial</a>
<hr/>

<table id="tabela" class="table table-bordered">
    <thead> <!-- cabeçalho -->
        <tr> <!-- linha -->
            <th>Código</th>
            <th>Nome do Estoque</th>
            <th>Tipo</th>
            <th>Descrição</th>
            <th>Operações</th>
        </tr>
    </thead>
    <tbody> <!-- corpo -->

    </tbody>
    <tfoot> <!-- rodapé -->
        <tr> <!-- linha -->
            <td colspan="5">
                Quantidade de Estoques: <span id="quantidade"></span>
            </td>
        </tr>
    </tfoot>
</table>
```



Criando uma função AJAX para executar a consulta de estoques na classe de controle:

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

@section scripts{

    <script src="~/Scripts/jquery-3.0.0.min.js"></script>

    <script>

        //quando a página esta carregada..
        $(document).ready(function () { //faça..

            //função ajax para executar a consulta de estoques..
            $.ajax({
                type: 'GET', //tipo de requisição HTTP GET
                url: '/Estoque/ConsultarEstoques', //método
                data: {}, //vazio..
                success: function () { //promisse..

                },
                error: function (e) { //promisse..
                    $("#mensagem").html("Erro: " + e.status);
                }
            });

        });

    </script>

}

<h4>Consulta de Estoques</h4>
<a href="/Home/Index">Página inicial</a>
<hr/>

<div id="mensagem"></div>
```

```
<table id="tabela" class="table table-bordered">
  <thead> <!-- cabeçalho -->
    <tr> <!-- linha -->
      <th>Código</th>
      <th>Nome do Estoque</th>
      <th>Tipo</th>
      <th>Descrição</th>
      <th>Operações</th>
    </tr>
  </thead>
  <tbody> <!-- corpo -->

</tbody>
<tfoot> <!-- rodapé -->
  <tr> <!-- linha -->
    <td colspan="5">
      Quantidade de Estoque: <span id="quantidade"></span>
    </td>
  </tr>
</tfoot>
</table>
```

Executando:



Programando as demais operações em banco de dados na classe **EstoqueRepositorio.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient; //sqlserver..
using Projeto.DAL.Entidades; //importando..
using Projeto.DAL.Entidades.Tipos; //importando..

namespace Projeto.DAL.Repositorios
{
```



```
public class EstoqueRepositorio : Conexao
{
    //método para inserir um estoque na base de dados..
    public void Insert(Estoque e)
    {
        //abrir conexão com o banco de dados..
        OpenConnection();

        //executar uma operação (query) no banco de dados..
        string query = "insert into Estoque(Nome, Descricao, Tipo) "
            + "values(@Nome, @Descricao, @Tipo)";

        //executando no banco de dados..
        cmd = new SqlCommand(query, con);
        cmd.Parameters.AddWithValue("@Nome", e.Nome);
        cmd.Parameters.AddWithValue("@Descricao", e.Descricao);
        cmd.Parameters.AddWithValue("@Tipo", (int) e.Tipo);
        cmd.ExecuteNonQuery(); //executando..

        //fechar conexão com o banco de dados..
        CloseConnection();
    }

    //método para atualizar um estoque na base de dados..
    public void Update(Estoque e)
    {
        OpenConnection(); //abrir conexão..

        string query = "update Estoque set Nome = @Nome,
            Descricao = @Descricao, "
            + "Tipo = @Tipo where IdEstoque = @IdEstoque";

        //executando..
        cmd = new SqlCommand(query, con);
        cmd.Parameters.AddWithValue("@IdEstoque", e.IdEstoque);
        cmd.Parameters.AddWithValue("@Nome", e.Nome);
        cmd.Parameters.AddWithValue("@Descricao", e.Descricao);
        cmd.Parameters.AddWithValue("@Tipo", (int) e.Tipo);
        cmd.ExecuteNonQuery(); //executando..

        CloseConnection(); //fechar conexão..
    }

    //método para excluir um estoque na base de dados..
    public void Delete(int idEstoque)
    {
        OpenConnection();

        string query = "delete from Estoque where IdEstoque = @IdEstoque";

        cmd = new SqlCommand(query, con);
        cmd.Parameters.AddWithValue("@IdEstoque", idEstoque);
        cmd.ExecuteNonQuery();

        CloseConnection();
    }

    //método para listar todos os estoques do banco de dados..
    public List<Estoque> FindAll()
    {

```

```

OpenConnection();

string query = "select * from Estoque";

cmd = new SqlCommand(query, con);
dr = cmd.ExecuteReader();

//declarando uma lista de Estoque..
List<Estoque> lista = new List<Estoque>();

//percorrer o objeto SqlDataReader..
while(dr.Read())
{
    Estoque e = new Estoque(); ///instanciando..
    e.IdEstoque = Convert.ToInt32(dr["IdEstoque"]);
    e.Nome = Convert.ToString(dr["Nome"]);
    e.Descricao = Convert.ToString(dr["Descricao"]);
    e.Tipo = (TipoEstoque) Convert.ToInt32(dr["Tipo"]);

    //adicionar o estoque na lista..
    lista.Add(e);
}

CloseConnection();
return lista; //retornando a lista
}

//método para obter 1 Estoque pelo id..
public Estoque FindById(int idEstoque)
{
    OpenConnection();

    string query = "select * from Estoque where IdEstoque = @IdEstoque";

    cmd = new SqlCommand(query, con);
    cmd.Parameters.AddWithValue("@IdEstoque", idEstoque);
    dr = cmd.ExecuteReader(); //executa e retorna os registros..

    Estoque e = null; //sem espaço de memória..

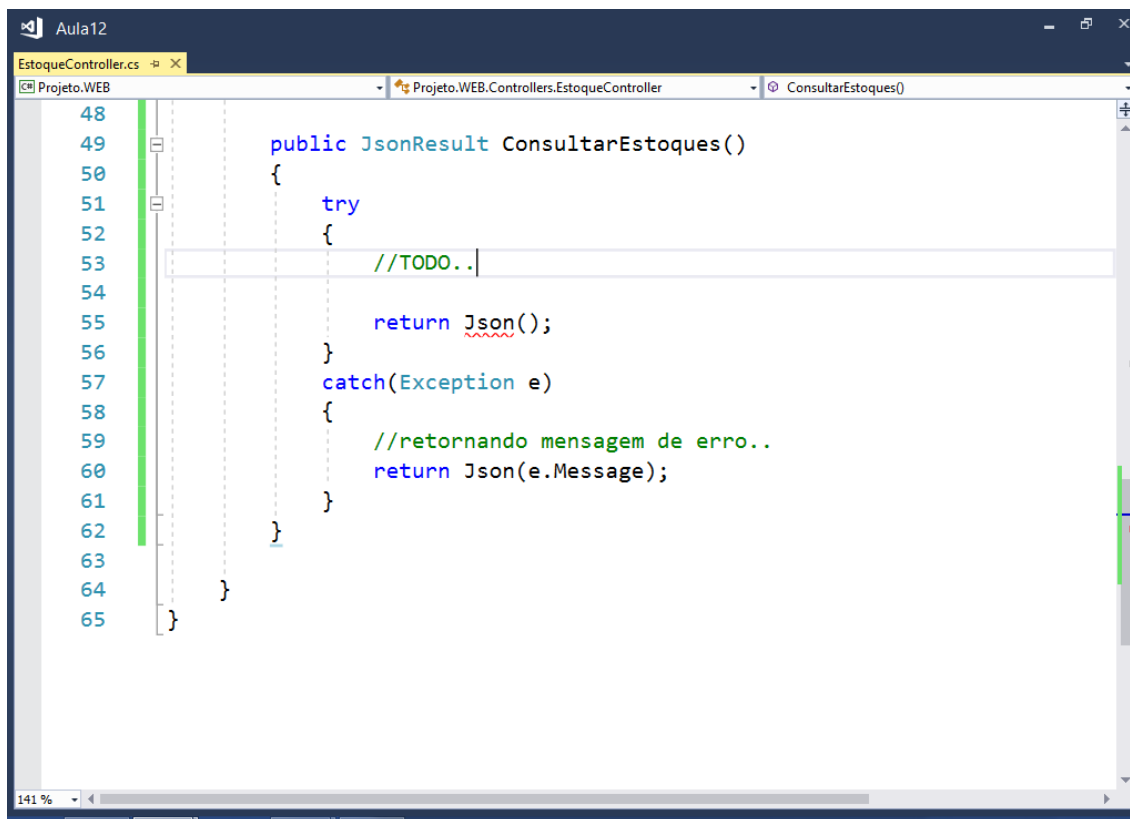
    if(dr.Read()) //se algum registro foi encontrado..
    {
        e = new Estoque(); //abrindo espaço de memória..

        e.IdEstoque = Convert.ToInt32(dr["IdEstoque"]);
        e.Nome = Convert.ToString(dr["Nome"]);
        e.Descricao = Convert.ToString(dr["Descricao"]);
        e.Tipo = (TipoEstoque)Convert.ToInt32(dr["Tipo"]);
    }

    CloseConnection();
    return e; //retornando o objeto Estoque..
}
}
}

```

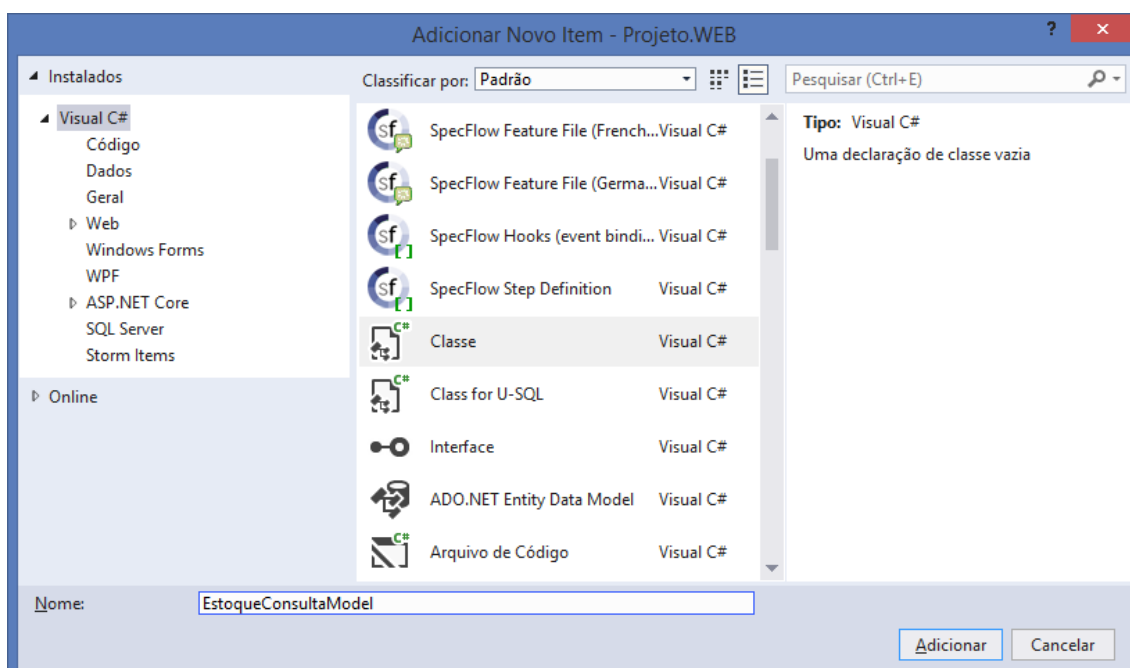
Voltando para a classe **EstoqueController.cs**
(Camada de controle do projeto MVC)



```

48
49     public JsonResult ConsultarEstoque()
50     {
51         try
52         {
53             //TODO..
54
55             return Json();
56         }
57         catch (Exception e)
58         {
59             //retornando mensagem de erro..
60             return Json(e.Message);
61         }
62     }
63
64 }
65
  
```

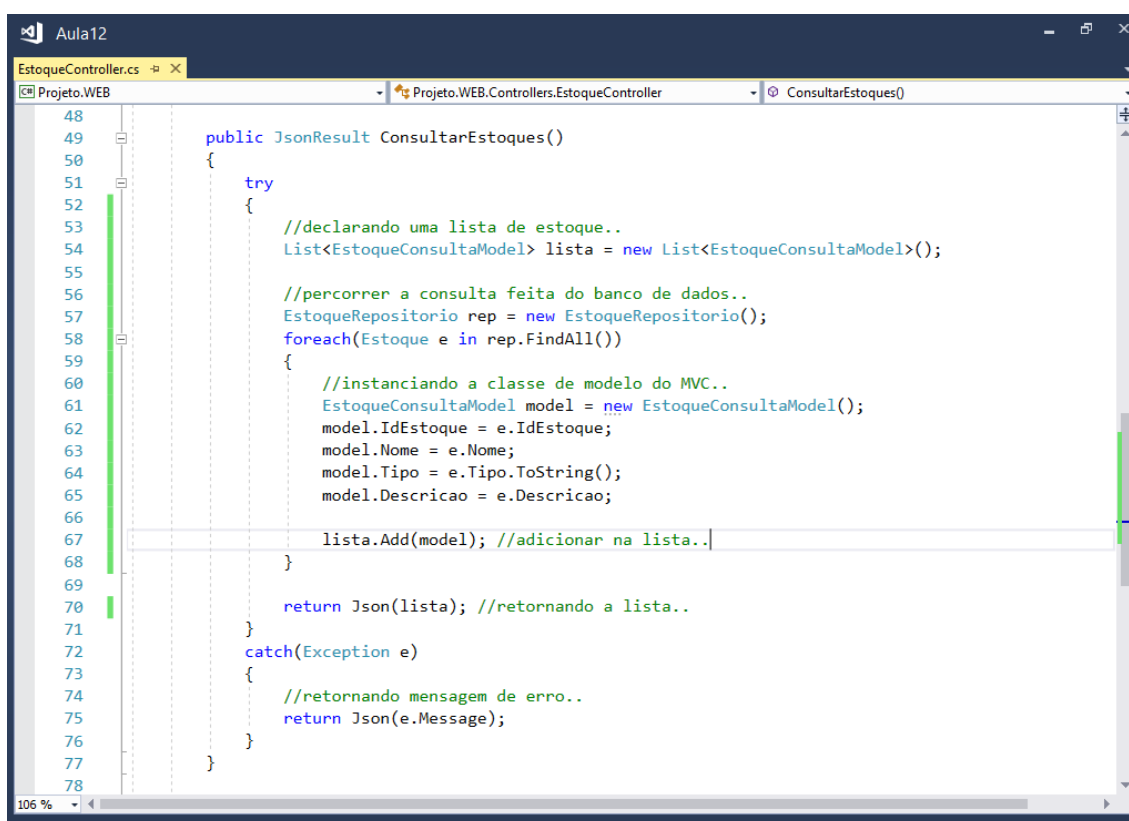
Criando uma classe de modelo para retornar os dados referentes a consulta de Estoque:



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

namespace Projeto.WEB.Models
{
    public class EstoqueConsultaModel
    {
        public int IdEstoque { get; set; }
        public string Nome { get; set; }
        public string Descricao { get; set; }
        public string Tipo { get; set; }
    }
}
```

Finalizando o método de consulta de Estoques:



```
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78

public JsonResult ConsultarEstoque()
{
    try
    {
        //declarando uma lista de estoque..
        List<EstoqueConsultaModel> lista = new List<EstoqueConsultaModel>();

        //percorrer a consulta feita do banco de dados..
        EstoqueRepositorio rep = new EstoqueRepositorio();
        foreach(Estoque e in rep.FindAll())
        {
            //instanciando a classe de modelo do MVC..
            EstoqueConsultaModel model = new EstoqueConsultaModel();
            model.IdEstoque = e.IdEstoque;
            model.Nome = e.Nome;
            model.Tipo = e.Tipo.ToString();
            model.Descricao = e.Descricao;

            lista.Add(model); //adicionar na lista..
        }

        return Json(lista); //retornando a lista..
    }
    catch(Exception e)
    {
        //retornando mensagem de erro..
        return Json(e.Message);
    }
}
```

Se a chamada AJAX para um método JsonResult seja do tipo HTTP GET é necessário especificar que o método JsonResult possui suporte a chamadas do tipo GET, caso contrario o método sequer ouviria a requisição.

JsonRequestBehavior.AllowGet

Faz com que o método JsonResult aceite requisições do tipo HTTP e GET e também retorna respostas.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Projeto.WEB.Models; //camada de modelo..
using Projeto.DAL.Entidades; //importando..
using Projeto.DAL.Repositorios; //importando..
using Projeto.DAL.Entidades.Tipos;

namespace Projeto.WEB.Controllers
{
    public class EstoqueController : Controller
    {
        // GET: Estoque/Cadastro
        public ActionResult Cadastro()
        {
            return View();
        }

        // GET: Estoque/Consulta
        public ActionResult Consulta()
        {
            return View();
        }

        public JsonResult CadastrarEstoque(EstoqueCadastroModel model)
        {
            try
            {
                //criando um objeto da classe Estoque..
                Estoque e = new Estoque();
                e.Nome = model.Nome;
                e.Descricao = model.Descricao;
                e.Tipo = (TipoEstoque) model.Tipo;

                //gravar na base de dados..
                EstoqueRepositorio rep = new EstoqueRepositorio();
                rep.Insert(e);

                return Json("Estoque cadastrado com sucesso.");
            }
            catch (Exception e)
            {
                return Json(e.Message);
            }
        }

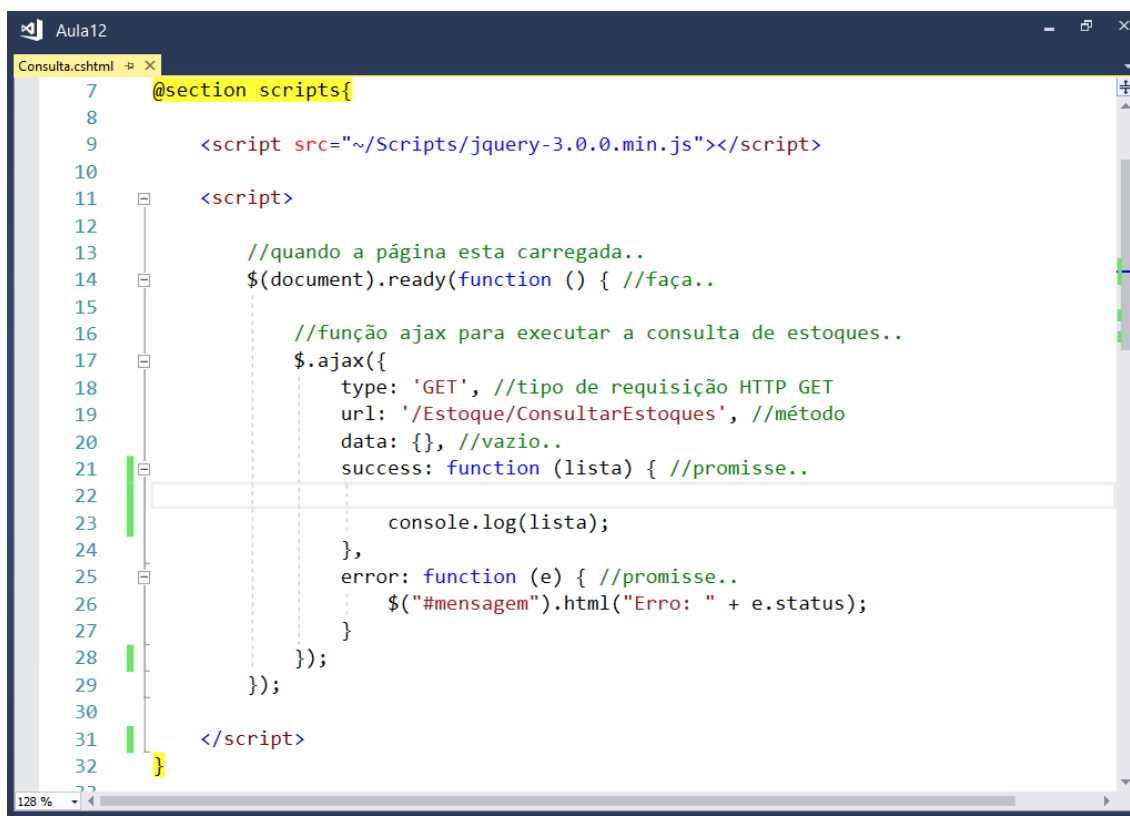
        public JsonResult ConsultarEstoque()
        {
            try
            {
                //declarando uma lista de estoque..
                List<EstoqueConsultaModel> lista
                    = new List<EstoqueConsultaModel>();
            }
        }
    }
}
```

```
//percorrer a consulta feita do banco de dados..
EstoqueRepositorio rep = new EstoqueRepositorio();
foreach(Estoque e in rep.FindAll())
{
    //instanciando a classe de modelo do MVC..
    EstoqueConsultaModel model = new EstoqueConsultaModel();
    model.IdEstoque = e.IdEstoque;
    model.Nome = e.Nome;
    model.Tipo = e.Tipo.ToString();
    model.Descricao = e.Descricao;

    lista.Add(model); //adicionar na lista..
}

return Json(lista, JsonRequestBehavior.AllowGet);
//retornando a lista..
}
catch(Exception e)
{
    //retornando mensagem de erro..
    return Json(e.Message, JsonRequestBehavior.AllowGet);
}
}
}
```

Voltando na página:



```
Aula12
ConsultarEstoque.cshtml
7  @section scripts{
8
9      <script src="~/Scripts/jquery-3.0.0.min.js"></script>
10
11     <script>
12
13         //quando a página esta carregada..
14         $(document).ready(function () { //faça..
15
16             //função ajax para executar a consulta de estoques..
17             $.ajax({
18                 type: 'GET', //tipo de requisição HTTP GET
19                 url: '/Estoque/ConsultarEstoque', //método
20                 data: {}, //vazio..
21                 success: function (lista) { //promisse..
22                     console.log(lista);
23                 },
24                 error: function (e) { //promisse..
25                     $("#mensagem").html("Erro: " + e.status);
26                 }
27             });
28         });
29     </script>
30
31 }
```

```
@section scripts{
```

```
<script src="~/Scripts/jquery-3.0.0.min.js"></script>
```

```
<script>
```

```
//quando a página esta carregada..
```

```
$(document).ready(function () { //faça..
```

```
//função ajax para executar a consulta de estoques..
```

```
$.ajax({
```

```
type: 'GET', //tipo de requisição HTTP GET
```

```
url: '/Estoque/ConsultarEstoques', //método
```

```
data: {}, //vazio..
```

```
success: function (lista) { //promisse..
```

```
console.log(lista);
```

```
},
```

```
error: function (e) { //promisse..
```

```
$("#mensagem").html("Erro: " + e.status);
```

```
}
```

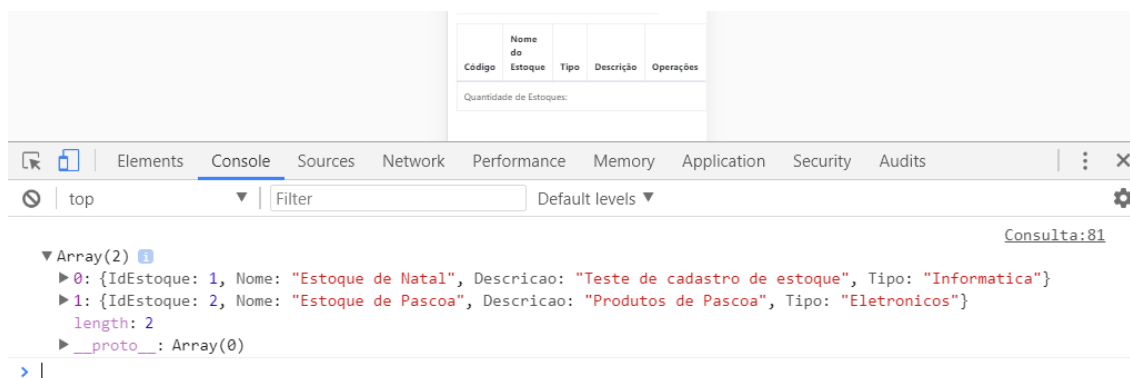
```
});
```

```
});
```

```
</script>
```

```
}
```

Testando:



The screenshot shows a web application interface with a table of inventory items. The table has columns: Nome do Estoque, Tipo, Descrição, and Operações. Below the table is a form labeled 'Quantidade de Estoques:'. The Chrome DevTools Console is open, showing the response of an AJAX call. The response is an array of two objects, each representing an inventory item.

Nome do Estoque	Tipo	Descrição	Operações
Estoque de Natal	Informatica	Teste de cadastro de estoque	
Estoque de Pascoa	Eletronicos	Produtos de Pascoa	

```

Array(2)
  0: {IdEstoque: 1, Nome: "Estoque de Natal", Descricao: "Teste de cadastro de estoque", Tipo: "Informatica"}
  1: {IdEstoque: 2, Nome: "Estoque de Pascoa", Descricao: "Produtos de Pascoa", Tipo: "Eletronicos"}
  length: 2
  __proto__: Array(0)
  
```

Exibindo a listagem de estoques:

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

@section scripts{

    <script src="~/Scripts/jquery-3.0.0.min.js"></script>

    <script>

        //quando a página esta carregada..
        $(document).ready(function () { //faça..

            //função ajax para executar a consulta de estoques..
            $.ajax({
                type: 'GET', //tipo de requisição HTTP GET
                url: '/Estoque/ConsultarEstoques', //método
                data: {}, //vazio..
                success: function (lista) { //promisse..

                    //declarando uma variavel..
                    var conteudo = "";
                    //percorrer o array de estoques..
                    $.each(lista, function (key, estoque) {

                        conteudo += "<tr>";
                        conteudo += "<td>" + estoque.IdEstoque + "</td>";
                        conteudo += "<td>" + estoque.Nome + "</td>";
                        conteudo += "<td>" + estoque.Descricao + "</td>";
                        conteudo += "<td>" + estoque.Tipo + "</td>";
                        conteudo += "<td>";

                        conteudo += "<button class='btn btn-danger btn-sm'>Excluir</button>";
                        conteudo += "&nbsp;"; //espaço..
                        conteudo += "<button class='btn btn-primary btn-sm'>Atualizar</button>";

                        conteudo += "</td>";
                        conteudo += "</tr>";
                    });

                    //imprimindo a variavel 'conteudo' dentro da tabela..
                    $("#tabela tbody").html(conteudo);

                    //imprimindo a quantidade de produtos..
                    //length -> retorna a quantidade de elementos de um item
                    $("#quantidade").html(lista.length);
                },
                error: function (e) { //promisse..
                    $("#mensagem").html("Erro: " + e.status);
                }
            });
        });
    </script>
}
```



```
<h4>Consulta de Estoques</h4>
<a href="/Home/Index">Página inicial</a>
<hr/>

<div id="mensagem"></div>

<table id="tabela" class="table table-bordered">

    <thead> <!-- cabeçalho -->
        <tr> <!-- linha -->
            <th>Código</th>
            <th>Nome do Estoque</th>
            <th>Tipo</th>
            <th>Descrição</th>
            <th>Operações</th>
        </tr>
    </thead>

    <tbody> <!-- corpo -->

    </tbody>

    <tfoot> <!-- rodapé -->
        <tr> <!-- linha -->
            <td colspan="5">
                Quantidade de Estoques: <span id="quantidade"></span>
            </td>
        </tr>
    </tfoot>

</table>
```



Código	Nome do Estoque	Tipo	Descrição	Operações
1	Estoque de Natal	Teste de cadastro de estoque	Informatica	<button>Excluir</button> <button>Atualizar</button>
2	Estoque de Pascoa	Produtos de Pascoa	Eletronicos	<button>Excluir</button> <button>Atualizar</button>
3	Estoque de Feriado	Estoque Teste	Games	<button>Excluir</button> <button>Atualizar</button>
4	Estoque de Livros	Estoque Especial de Livros	Livraria	<button>Excluir</button> <button>Atualizar</button>
5	Estoque de Notebooks	Notebooks em Promoção	Informatica	<button>Excluir</button> <button>Atualizar</button>

Quantidade de Estoques: 5

Criando janelas modais no bootstrap:

```
@{
```

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

```
}
```

```
@section scripts{
```

```
<script src="~/Scripts/jquery-3.0.0.min.js"></script>  
<script src="~/Scripts/bootstrap.min.js"></script>
```

```
<script>
```

```
//quando a página esta carregada..  
$(document).ready(function () { //faça..
```

```
//função ajax para executar a consulta de estoques..
```

```
$.ajax({  
    type: 'GET', //tipo de requisição HTTP GET  
    url: '/Estoque/ConsultarEstoques', //método  
    data: {}, //vazio..  
    success: function (lista) { //promisse..
```

```
        //declarando uma variavel..  
        var conteudo = "";
```

```
        //percorrer o array de estoques..  
        $.each(lista, function (key, estoque) {
```

```
            conteudo += "<tr>";  
            conteudo += "<td>" + estoque.IdEstoque + "</td>";  
            conteudo += "<td>" + estoque.Nome + "</td>";  
            conteudo += "<td>" + estoque.Descricao + "</td>";  
            conteudo += "<td>" + estoque.Tipo + "</td>";  
            conteudo += "<td>";
```

```
                conteudo += "<button  
                    data-target='#janelaexclusao'  
                    data-toggle='modal' class='btn btn-danger  
                        btn-sm'>Excluir</button>";
```

```
                conteudo += "&nbsp;"; //espaço..  
                conteudo += "<button data-target='#janelaedicao'  
                    data-toggle='modal' class='btn btn-primary  
                        btn-sm'>Atualizar</button>";
```

```
            conteudo += "</td>";
```

```
        conteudo += "</tr>";
```

```
    });
```

```

        //imprimindo a variavel 'conteudo' dentro da tabela..
        $("#tabela tbody").html(conteudo);

        //imprimindo a quantidade de produtos..
        //length -> retorna a quantidade de elementos de um item
        $("#quantidade").html(lista.length);
    },
    error: function (e) { //promisse..
        $("#mensagem").html("Erro: " + e.status);
    }
});
});

</script>
}

<h4>Consulta de Estoques</h4>
<a href="/Home/Index">Página inicial</a>
<hr/>

<div id="mensagem"></div>

<table id="tabela" class="table table-bordered">
    <thead> <!-- cabeçalho -->
        <tr> <!-- linha -->
            <th>Código</th>
            <th>Nome do Estoque</th>
            <th>Tipo</th>
            <th>Descrição</th>
            <th>Operações</th>
        </tr>
    </thead>
    <tbody> <!-- corpo -->

    </tbody>
    <tfoot> <!-- rodapé -->
        <tr> <!-- linha -->
            <td colspan="5">
                Quantidade de Estoques: <span id="quantidade"></span>
            </td>
        </tr>
    </tfoot>
</table>

<!-- janela para excluir o produto -->
<div id="janelaexclusao" class="modal fade">
    <div class="modal-dialog">
        <div class="modal-content">

            <!-- cabeçalho -->
            <div class="modal-header bg-danger text-white">
                <h4 class="modal-title">Exclusão de Estoque</h4>

```

```

        <button type="button" class="close text-white"
            data-dismiss="modal">&times;</button>
    </div>

    <!-- corpo -->
    <div class="modal-body">

        <h4>Deseja excluir o estoque selecionado?</h4>
        <hr/>

        Id do Estoque: <span id="idestoque_exclusao"></span>
        <br/>

        Nome: <span id="nome_exclusao"></span>
        <br />

        Tipo: <span id="tipo_exclusao"></span>
        <br />

        Descrição: <span id="descricao_exclusao"></span>
        <br />

    </div>

    <!-- rodape -->
    <div class="modal-footer">
        <button class="btn btn-danger">
            Confirmar Exclusão
        </button>
    </div>

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

<!-- janela para atualizar o produto -->
<div id="janelaedicao" class="modal fade">
    <div class="modal-dialog">
        <div class="modal-content">

            <!-- cabeçalho -->
            <div class="modal-header bg-primary text-white">
                <h4 class="modal-title">Edição de Estoque</h4>
                <button type="button" class="close text-white"
                    data-dismiss="modal">&times;</button>
            </div>

            <!-- corpo -->
            <div class="modal-body">

                <label>Id do Estoque:</label>
                <input type="text" id="idestoque" class="form-control"

```

```

        readonly="readonly"/>
    <br/>

    <label>Nome do Estoque:</label>
    <input type="text" id="nome" class="form-control" />
    <br />

    <label>Descrição do Estoque:</label>
    <textarea id="descricao" class="form-control"
        rows="4"></textarea>
    <br />

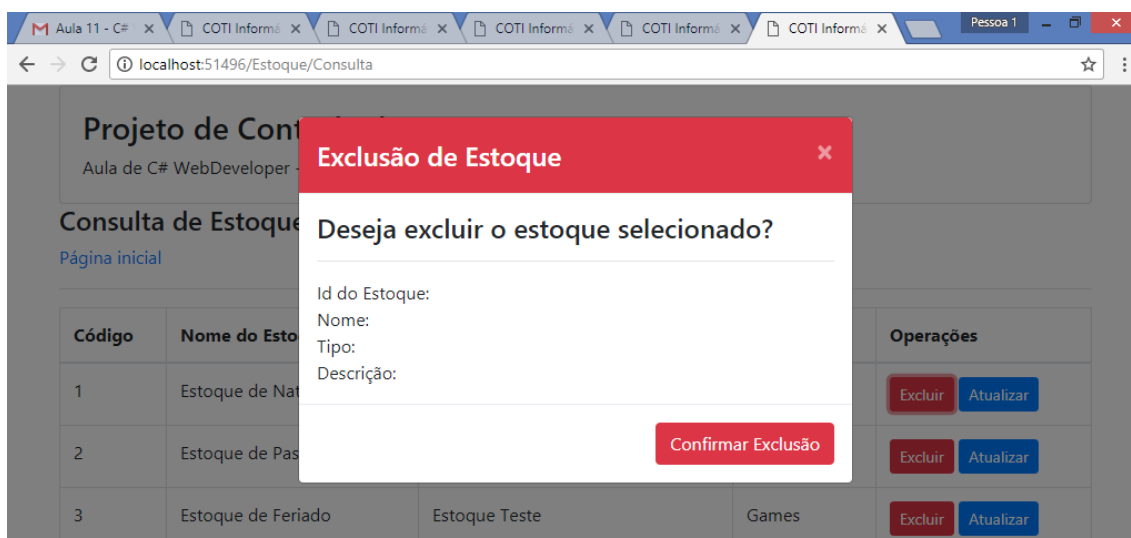
    <label>Tipo do Estoque</label>
    <select id="tipo" class="form-control">
        <option value="">Selecione uma Opção</option>
        <option value="1">Informática</option>
        <option value="2">Eletrônicos</option>
        <option value="3">Livraria</option>
        <option value="4">Games</option>
        <option value="5">Outros</option>
    </select>
    <br />

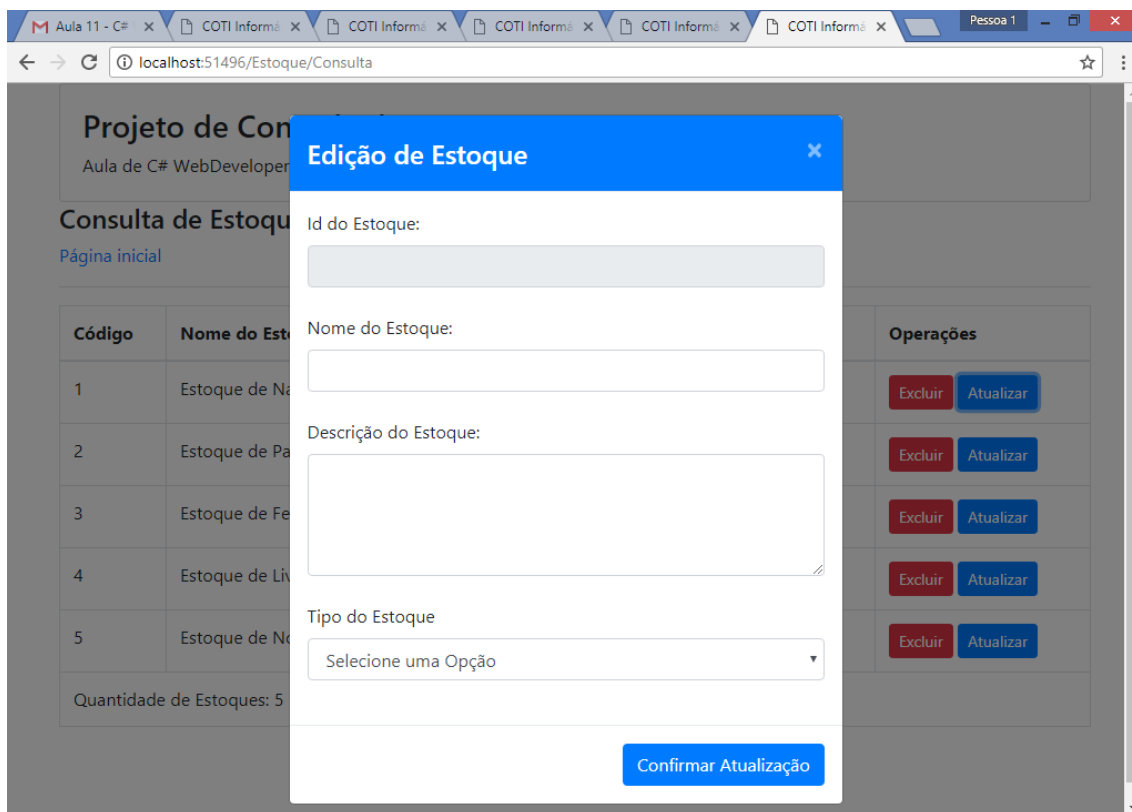
</div>

<!-- rodape -->
<div class="modal-footer">
    <button class="btn btn-primary">
        Confirmar Atualização
    </button>
</div>

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

```





The screenshot shows a web browser window with the URL `localhost:51496/Estoque/Consulta`. The page displays a table of inventory items and a modal form for editing an item.

Projeto de Consulta
Aula de C# WebDeveloper

Consulta de Estoque
[Página inicial](#)

Código	Nome do Estoque
1	Estoque de N...
2	Estoque de Pa...
3	Estoque de Fe...
4	Estoque de Li...
5	Estoque de N...

Quantidade de Estoque: 5

Edição de Estoque

Id do Estoque:

Nome do Estoque:

Descrição do Estoque:

Tipo do Estoque:

Operações

<input type="button" value="Excluir"/>	<input type="button" value="Atualizar"/>
<input type="button" value="Excluir"/>	<input type="button" value="Atualizar"/>
<input type="button" value="Excluir"/>	<input type="button" value="Atualizar"/>
<input type="button" value="Excluir"/>	<input type="button" value="Atualizar"/>
<input type="button" value="Excluir"/>	<input type="button" value="Atualizar"/>

Continua...