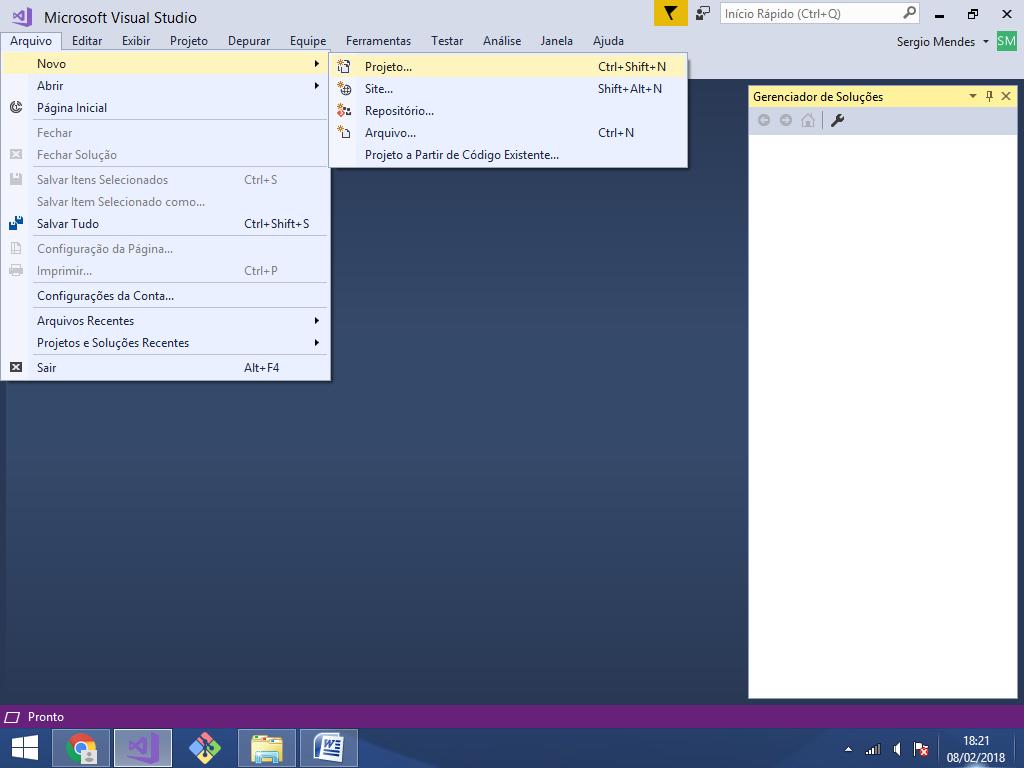
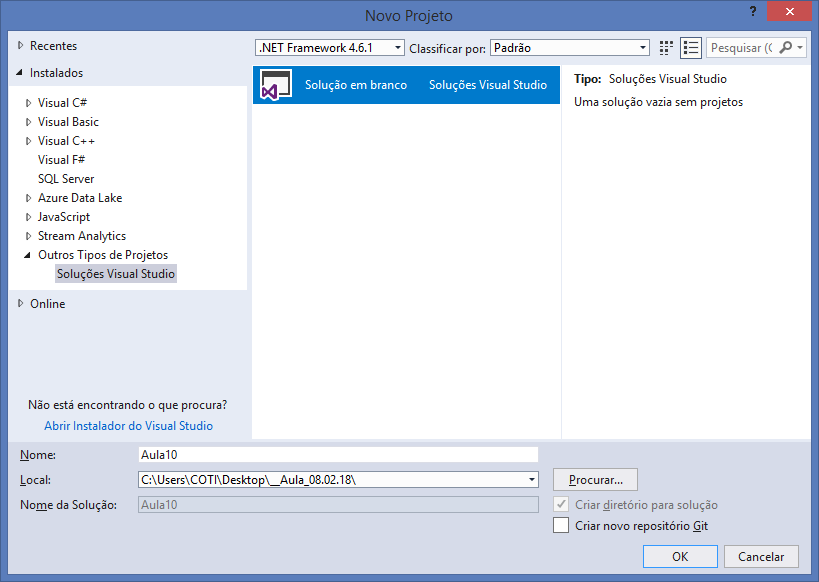
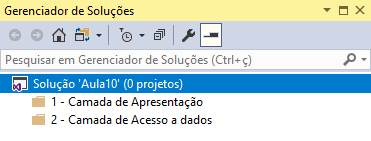
Criando uma nova solution em branco:





**Separando a solution em pastas:**

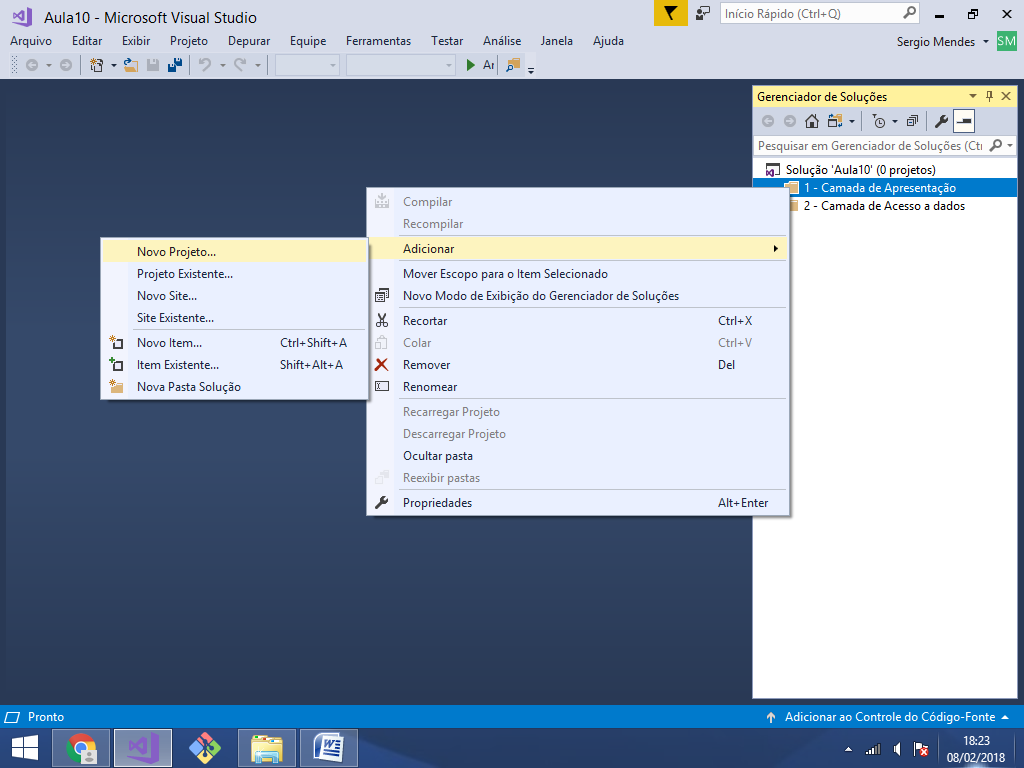
Solution Folders

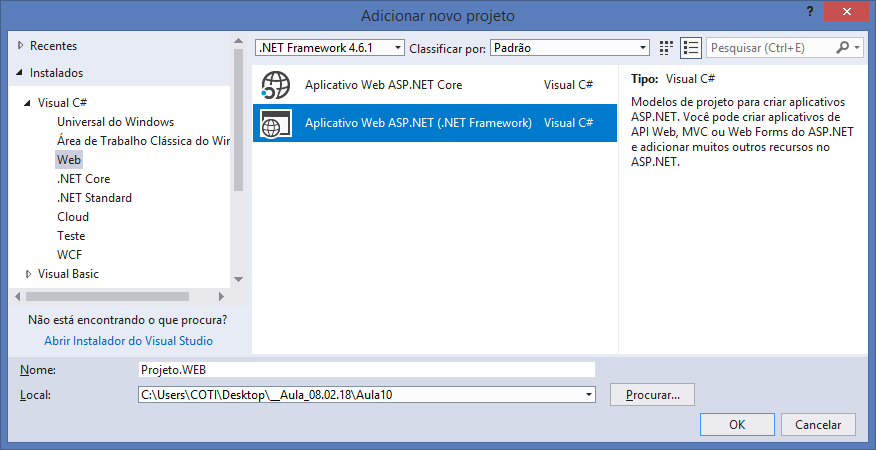


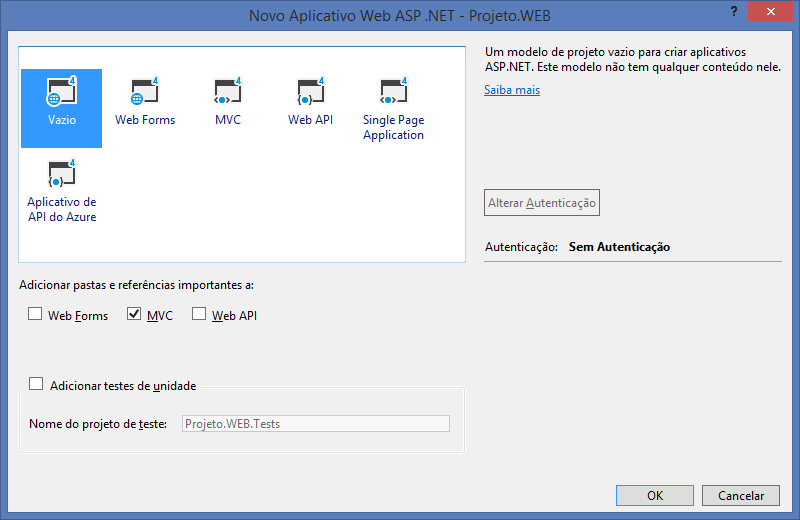
--------------------------------

1 - Camada de Apresentação

Criando o projeto Asp.Net MVC



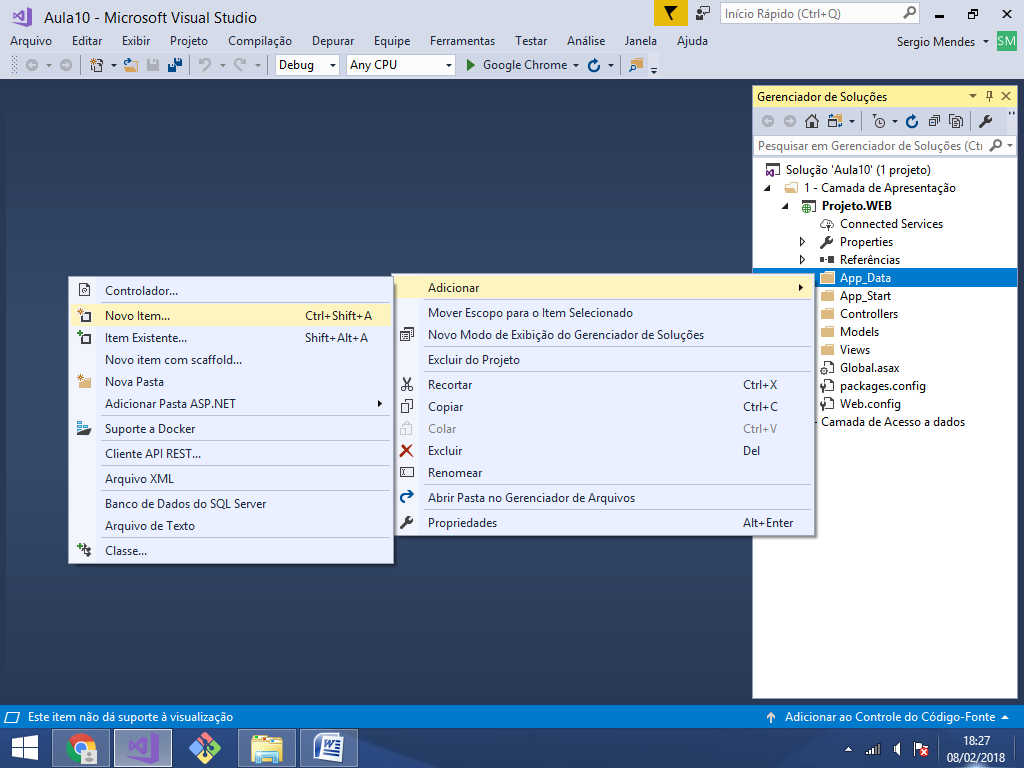


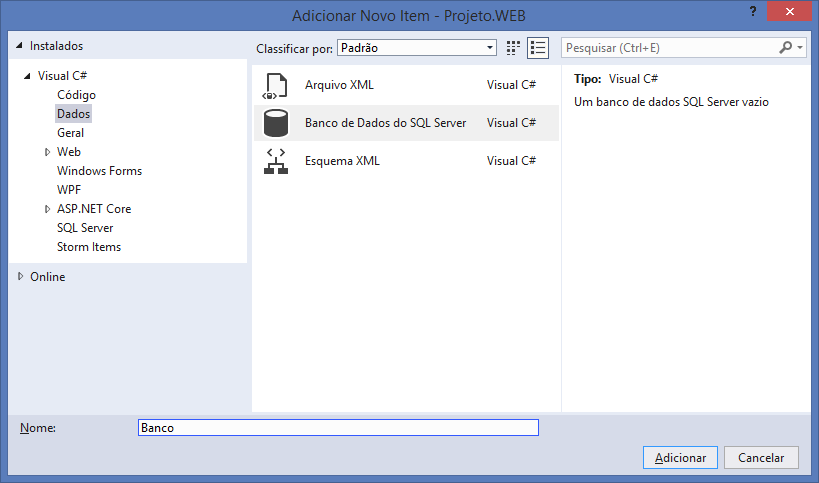


**Criando a base de dados**

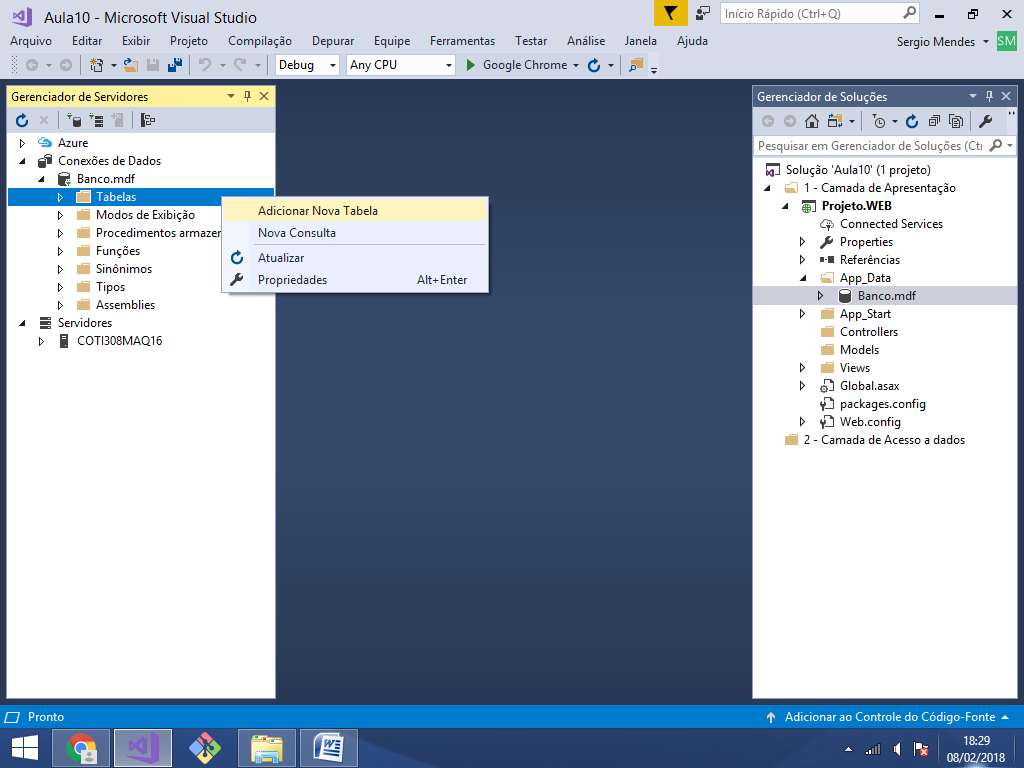
MDF - Master Database File

(Arquivo de banco de dados)





**Criando uma tabela de produtos na base de dados:**



**create table Produto(**

**IdProduto integer identity,**

**Nome nvarchar(50) not null,**

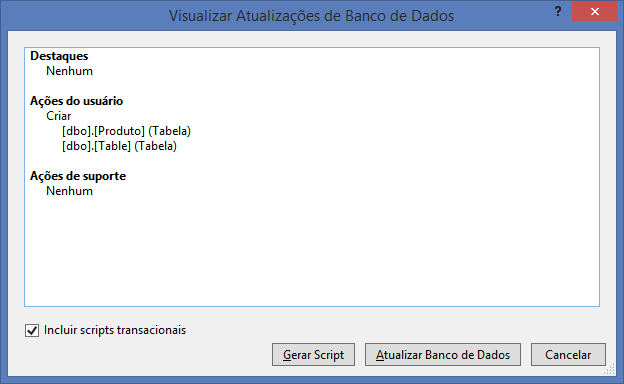
**Preco decimal(18,2) not null,**

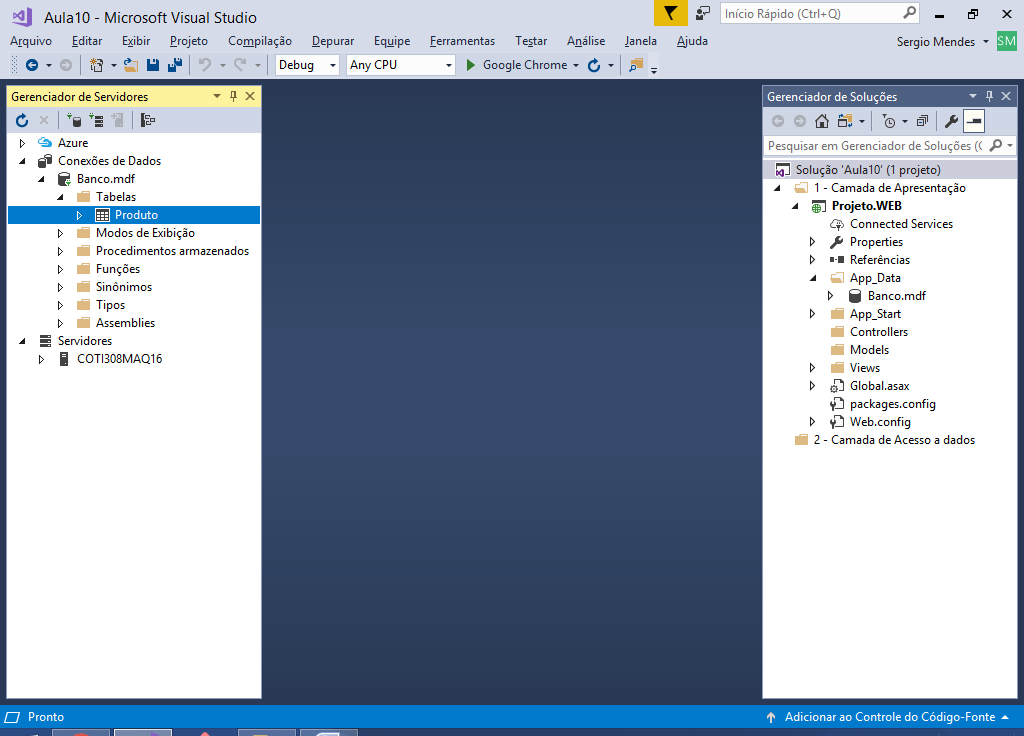
**Quantidade integer not null,**

**primary key(IdProduto))**

**go**

**Executando:**





\Web.config.xml

Mapeamento da connectionstring

<connectionStrings>

<add

name="aula10"

connectionString="Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=C:\Users\COTI\Desktop\\_\_Aula\_08.02.18\

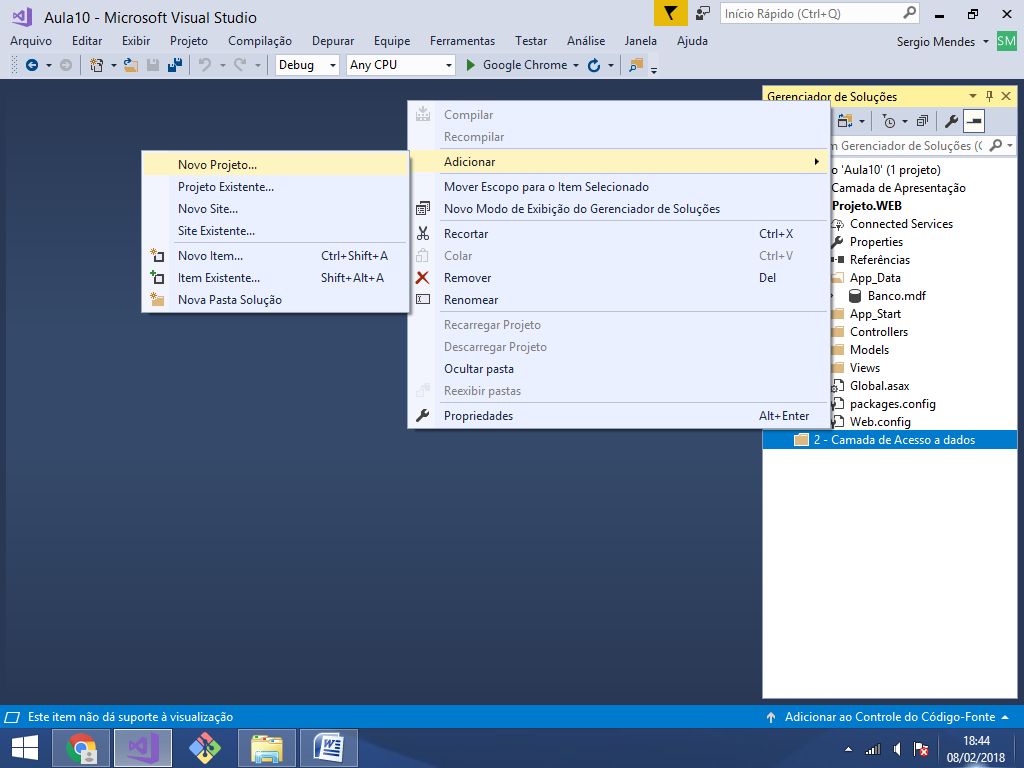
Aula10\Projeto.WEB\App\_Data\Banco.mdf;Integrated Security=True"

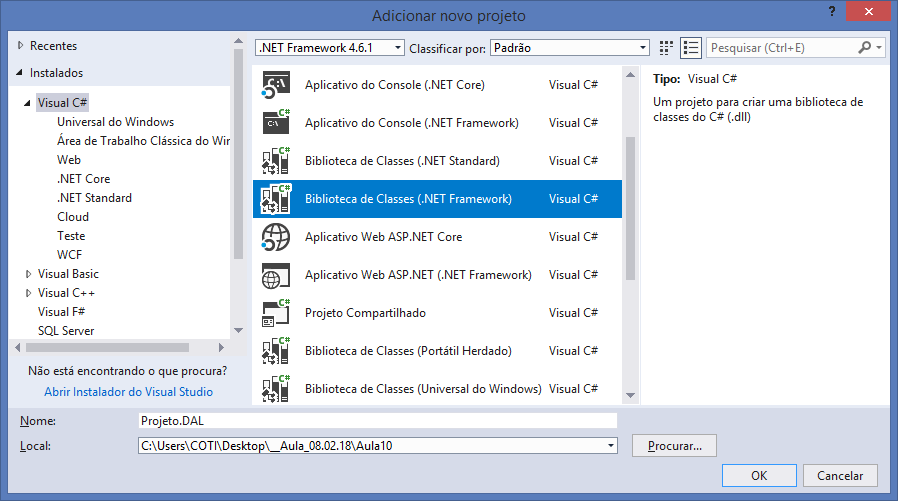
/>

</connectionStrings>

2 - Camada de Acesso a dados

Biblioteca de Classes (Class Library)





Classe de entidade: **Produto**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Projeto.DAL.Entidades

{

public class Produto

{

//propriedades..

public int IdProduto { get; set; }

public string Nome { get; set; }

public decimal Preco { get; set; }

public int Quantidade { get; set; }

//construtor default..

public Produto()

{

//[ctor] + 2x[tab]

}

//construtor com entrada de argumentos (sobrecarga de construtores)

public Produto(int idProduto, string nome, decimal preco, int quantidade)

{

IdProduto = idProduto;

Nome = nome;

Preco = preco;

Quantidade = quantidade;

}

//sobrescrever o método ToString()..

public override string ToString()

{

return $"Id do Produto: {IdProduto}, Nome: {Nome},

Preço: {Preco}, Quantidade: {Quantidade}";

}

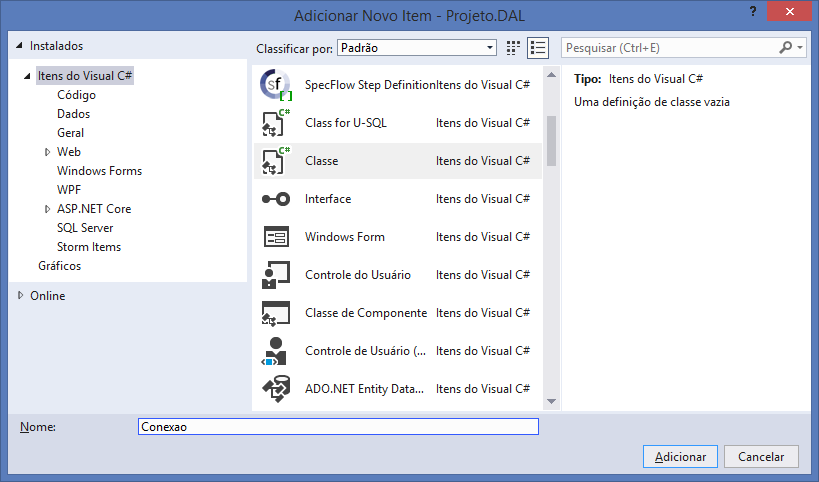
}

}

------------------------------------

**Criando a classe para conexão com o banco de dados:**

/Repositorios/Conexao.cs



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient; //biblioteca para acesso ao SqlServer..

namespace Projeto.DAL.Repositorios

{

public class Conexao

{

//atributos..

protected SqlConnection con; //conexão com o banco de dados..

protected SqlCommand cmd; //executar comandos SQL..

protected SqlDataReader dr; //ler dados das consultas..

protected SqlTransaction tr; //transações (commit, rollback)

//método para abrir conexão com o banco de dados..

protected void OpenConnection()

{

con = new SqlConnection();

con.Open(); //conexão aberta..

}

//método para fechar conexão com o banco de dados..

protected void CloseConnection()

{

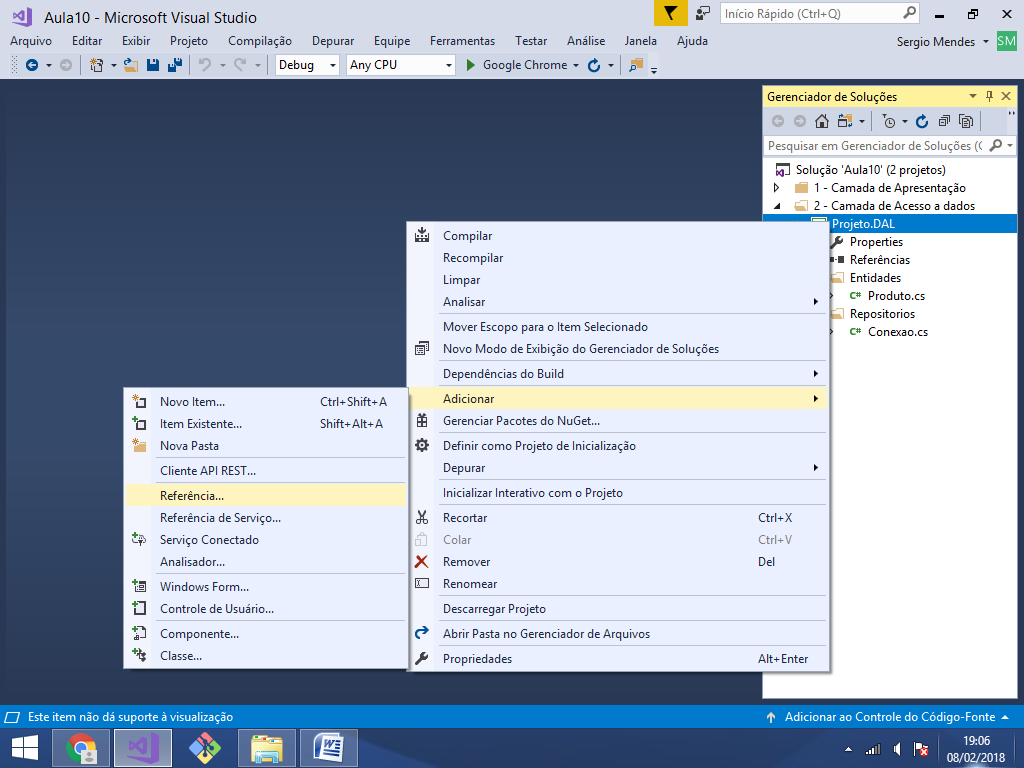
con.Close(); //conexão fechada..

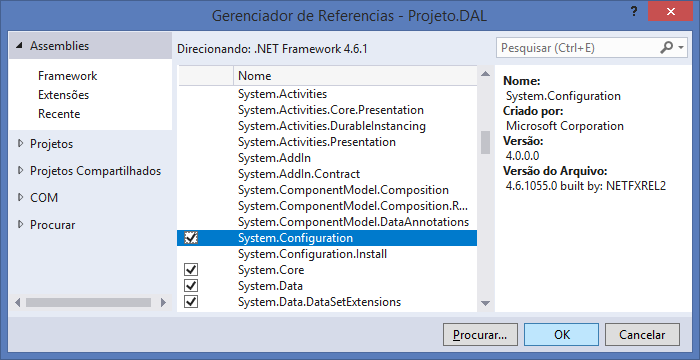
}

}

}

É necessário adicionar no projeto DAL uma referencia para a biblioteca **System.Configuration** de modo que seja possivel referenciar o endereço da connectionstring do banco de dados.





using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient; //biblioteca para acesso ao SqlServer..

using System.Configuration; //connectionstring

namespace Projeto.DAL.Repositorios

{

public class Conexao

{

//atributos..

protected SqlConnection con; //conexão com o banco de dados..

protected SqlCommand cmd; //executar comandos SQL..

protected SqlDataReader dr; //ler dados das consultas..

protected SqlTransaction tr; //transações (commit, rollback)

//método para abrir conexão com o banco de dados..

protected void OpenConnection()

{

con = new SqlConnection(ConfigurationManager.ConnectionStrings

["aula10"].ConnectionString);

con.Open(); //conexão aberta..

}

//método para fechar conexão com o banco de dados..

protected void CloseConnection()

{

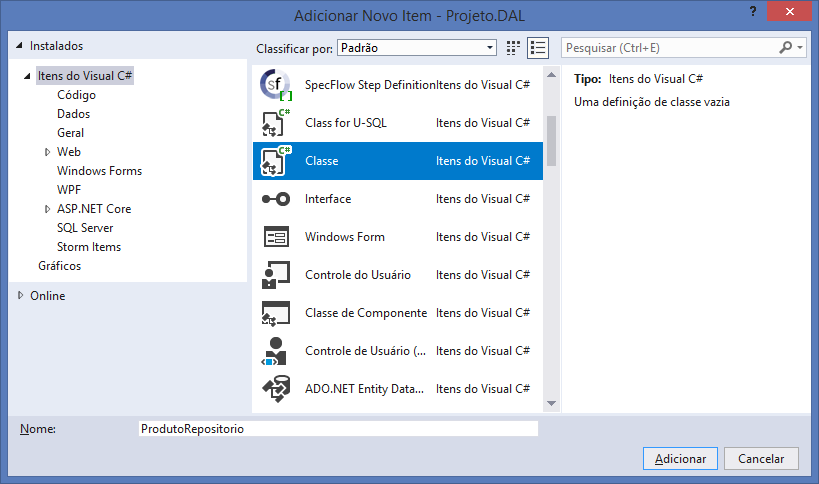
con.Close(); //conexão fechada..

}

}

}

Classe: **/Repositorios/ProdutoRepositorio.cs**



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient; //acesso ao sqlserver..

using Projeto.DAL.Entidades; //classes de entidade..

namespace Projeto.DAL.Repositorios

{

//REGRA: Cada classe de 'Repositorio' HERDA a classe 'Conexao'

public class ProdutoRepositorio : Conexao

{

//método para gravar um produto na base de dados..

public void Insert(Produto p)

{

OpenConnection(); //abrir conexão..

//escrever o comando SQL (query)

string query = "insert into Produto(Nome, Preco, Quantidade) "

+ "values(@Nome, @Preco, @Quantidade)";

//executando no banco de dados..

cmd = new SqlCommand(query, con);

cmd.Parameters.AddWithValue("@Nome", p.Nome);

cmd.Parameters.AddWithValue("@Preco", p.Preco);

cmd.Parameters.AddWithValue("@Quantidade", p.Quantidade);

cmd.ExecuteNonQuery(); //executando..

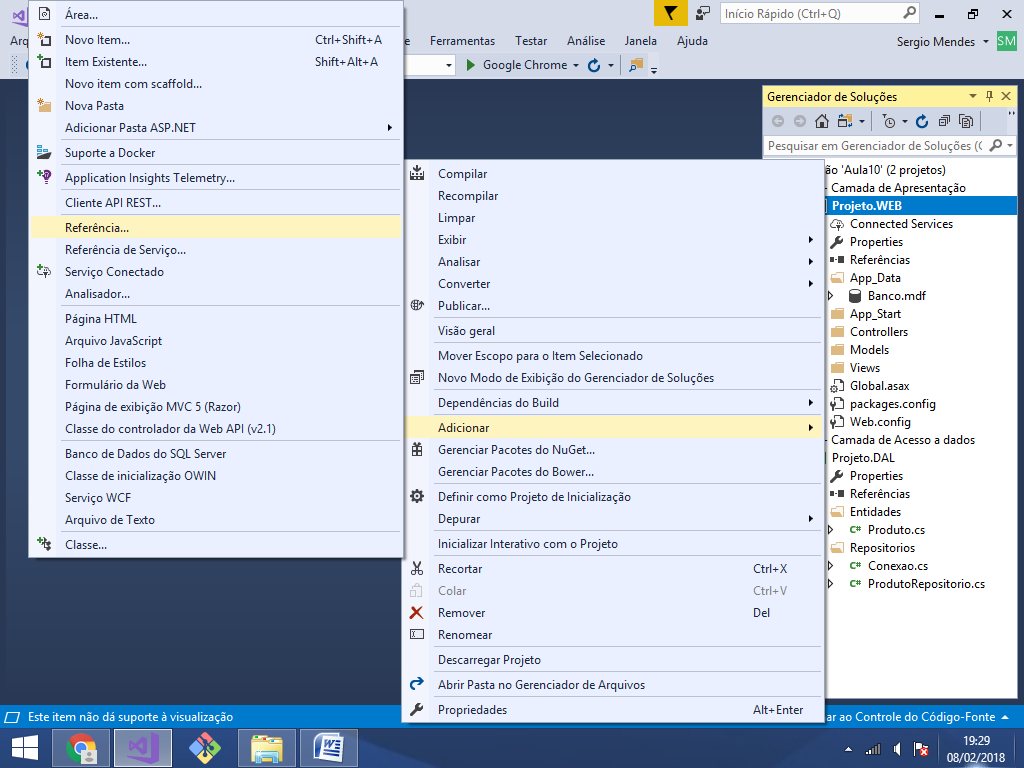
CloseConnection(); //fechar conexão..

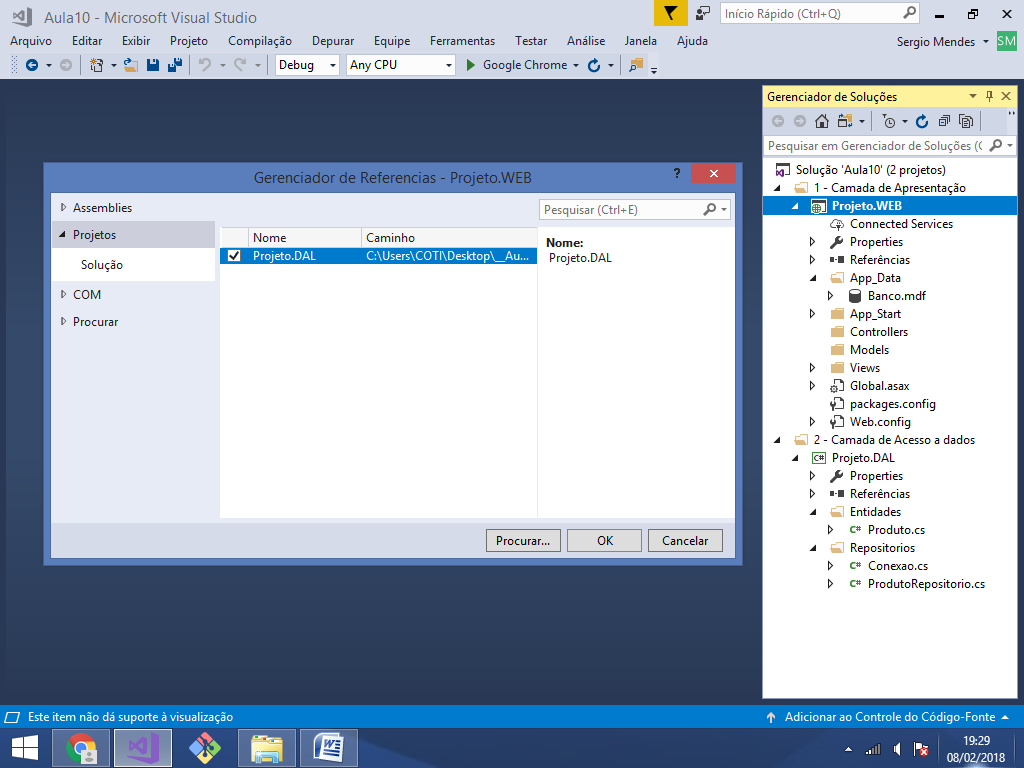
}

}

}

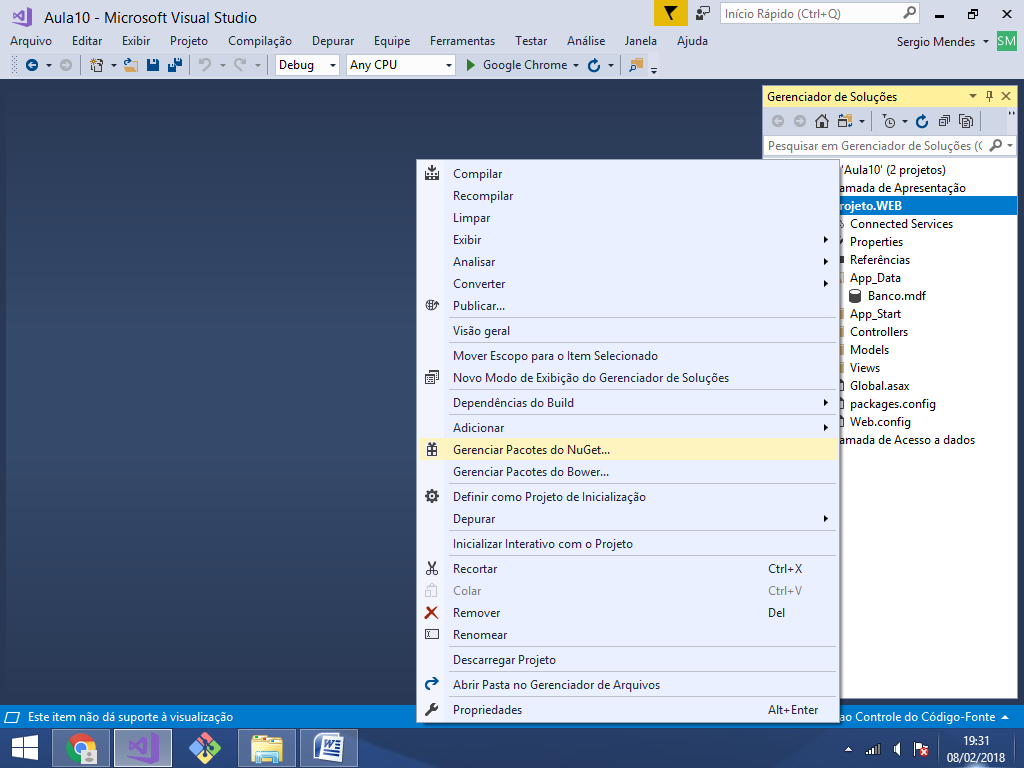
Adicionando adicionar referencia no projeto WEB para o projeto DAL

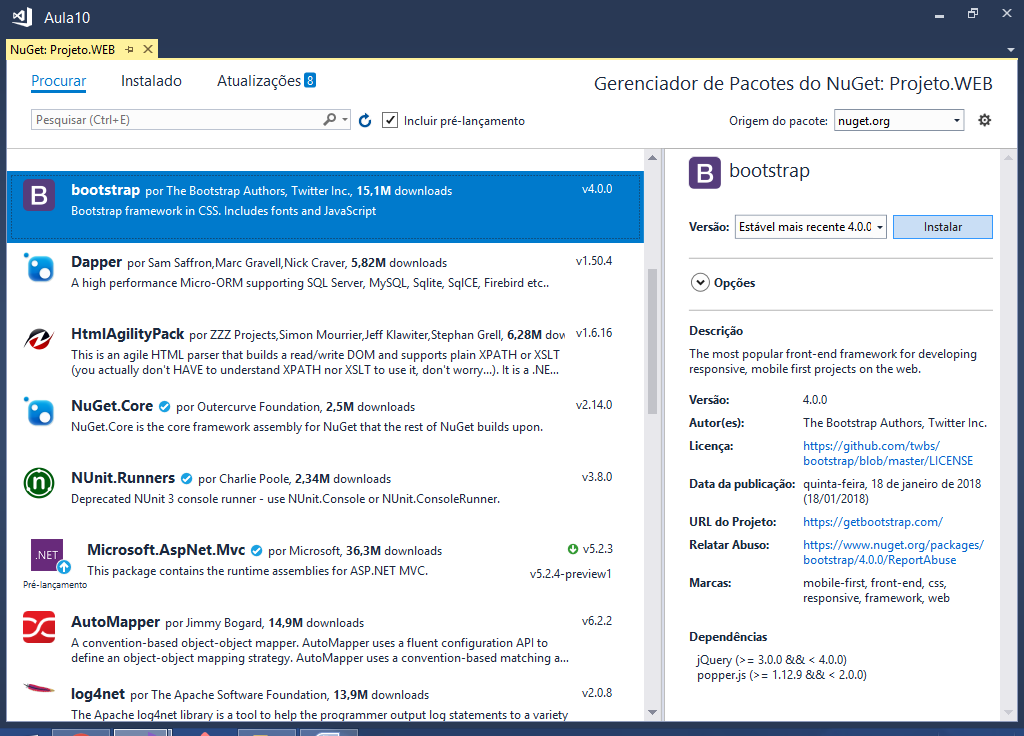




**Instalando o bootstrap no projeto MVC:**

Gerenciador de pacotes do NuGet



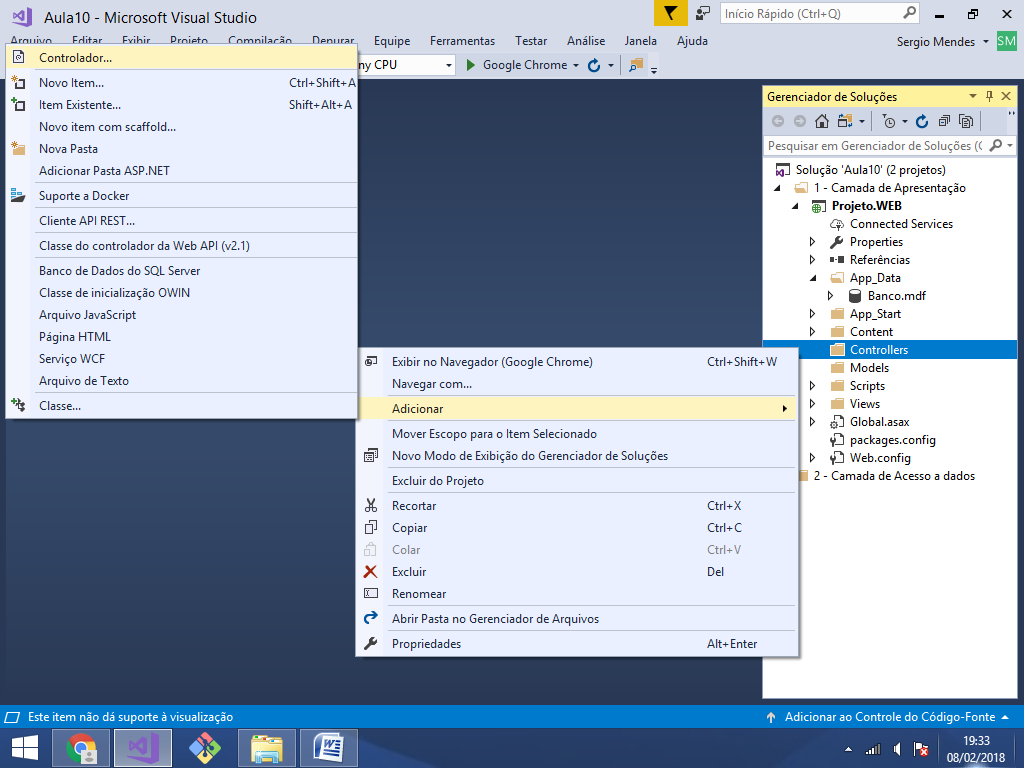


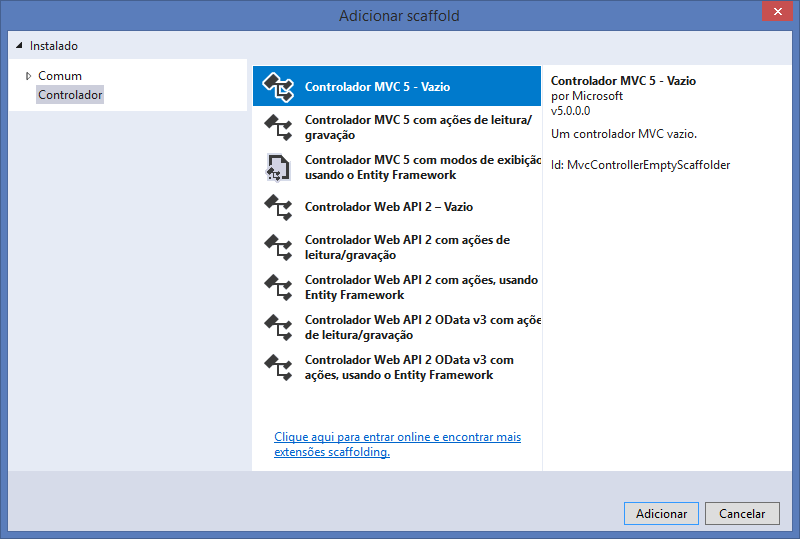
Criando a página inicial do projeto

/Home/Index

[Controller] [View]

**Passo 1) Criando o controller:**





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()

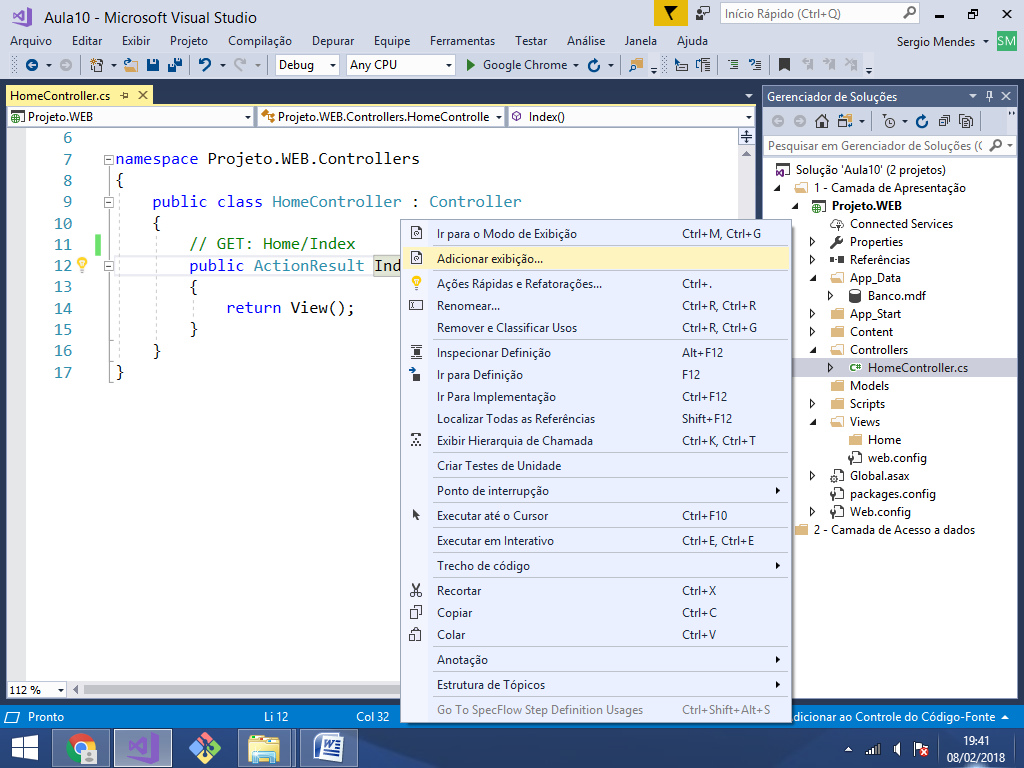
{

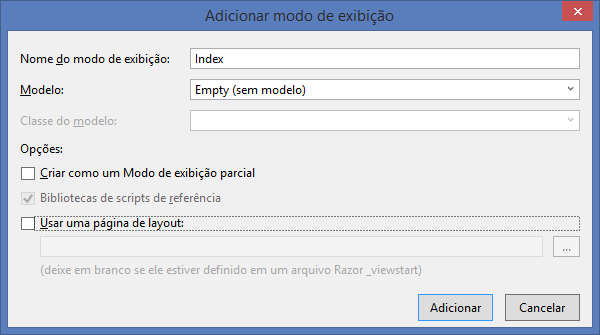
return View();

}

}

}





<!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>

<h2>Projeto Controle de Produtos</h2>

Asp.Net MVC com JQuery (AJAX)

<hr/>

<a href="/Produto/Cadastro">Cadastrar Produtos</a>

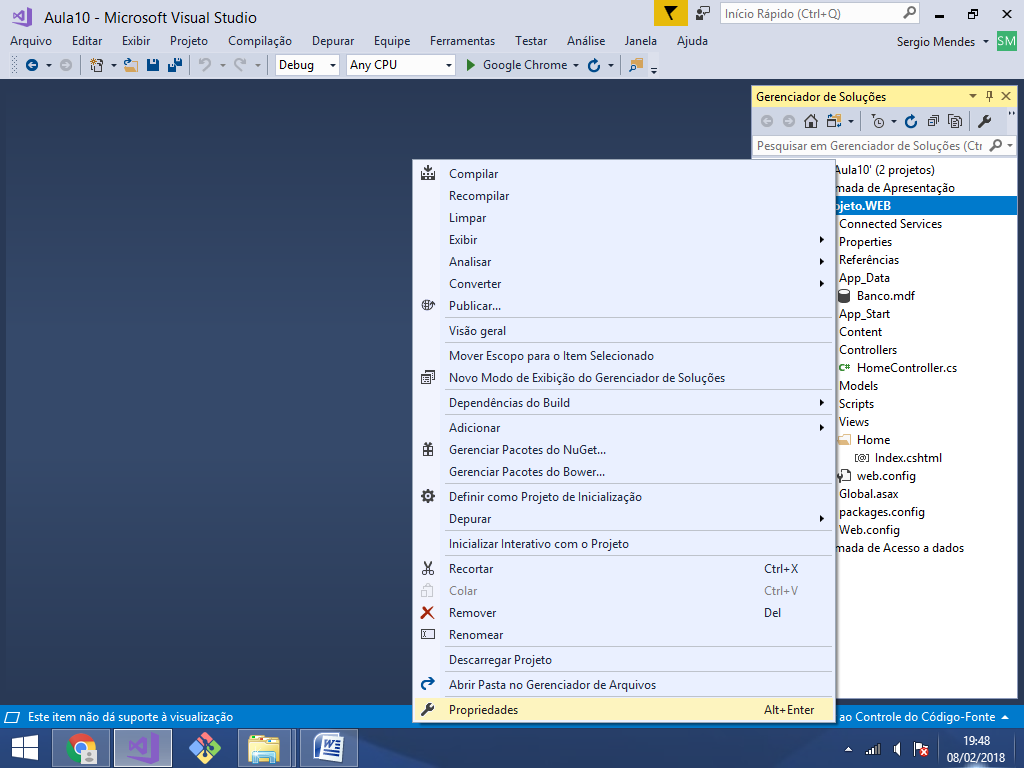
</div>

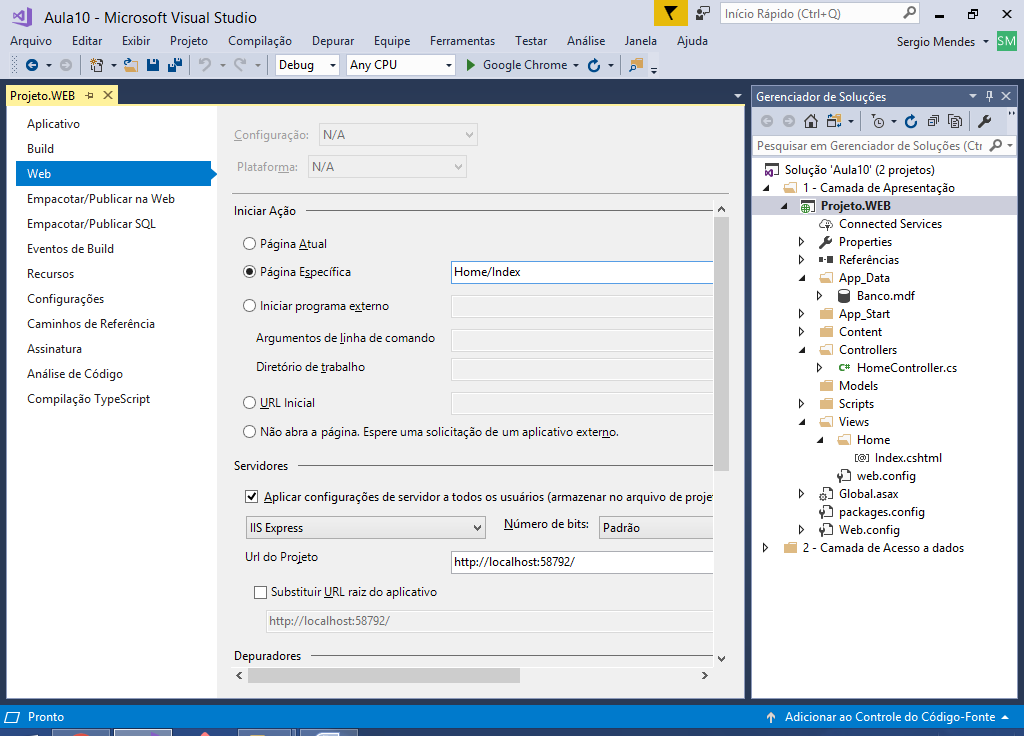
</body>

</html>

---------------------------------------------

**Definindo a página inicial do projeto:**





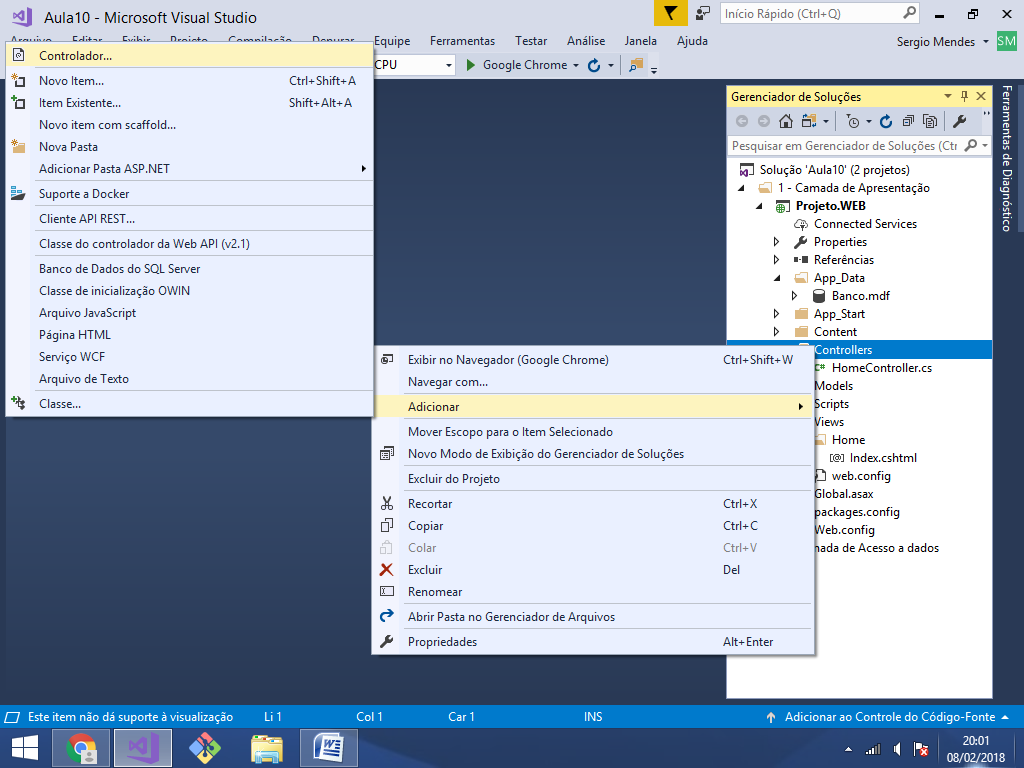
Executando:

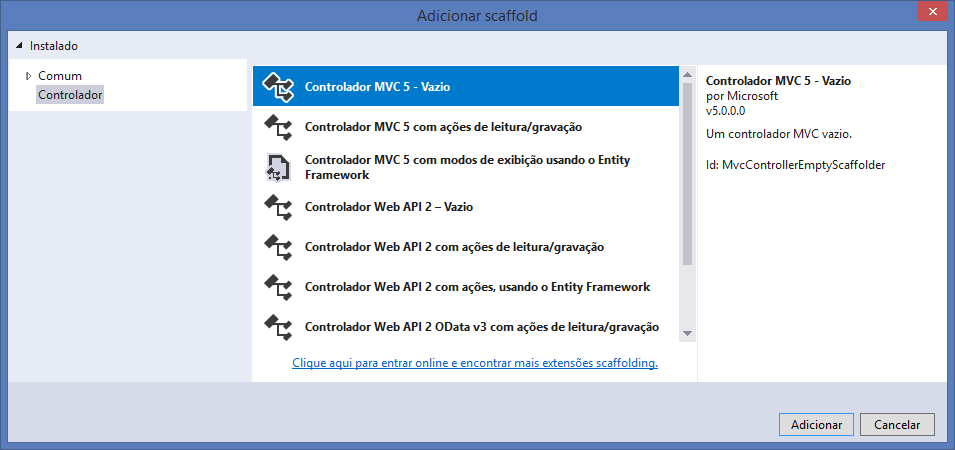
http://localhost:58792/Home/Index

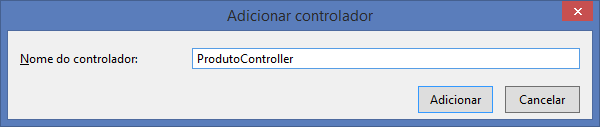


Criando um controle de produtos:

/ProdutoController.cs







using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Projeto.WEB.Controllers

{

public class ProdutoController : Controller

{

// GET: Produto/Cadastro

public ActionResult Cadastro()

{

return View();

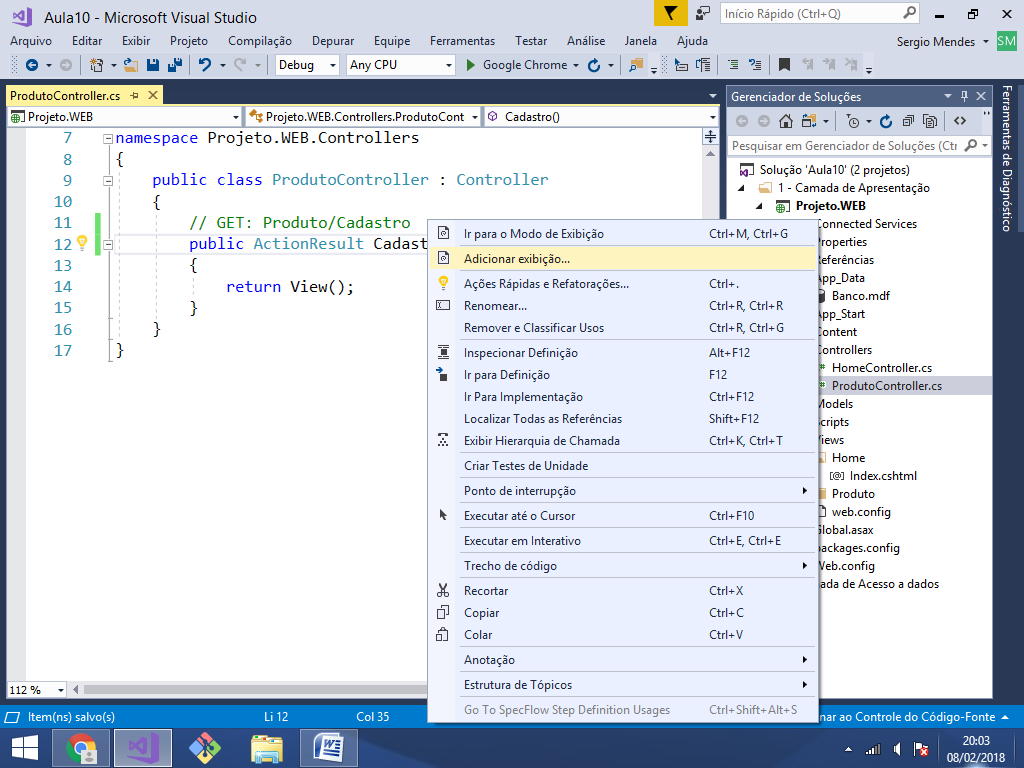
}

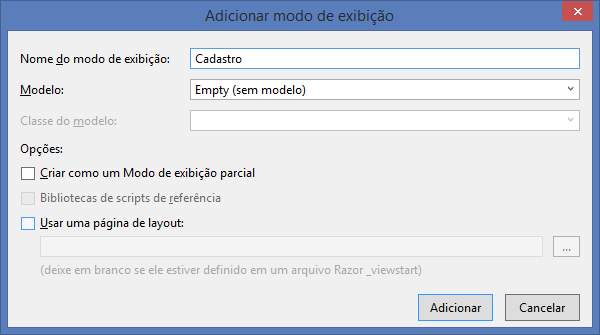
}

}

--------------------------------------------------

**Gerando a página de cadastro:**

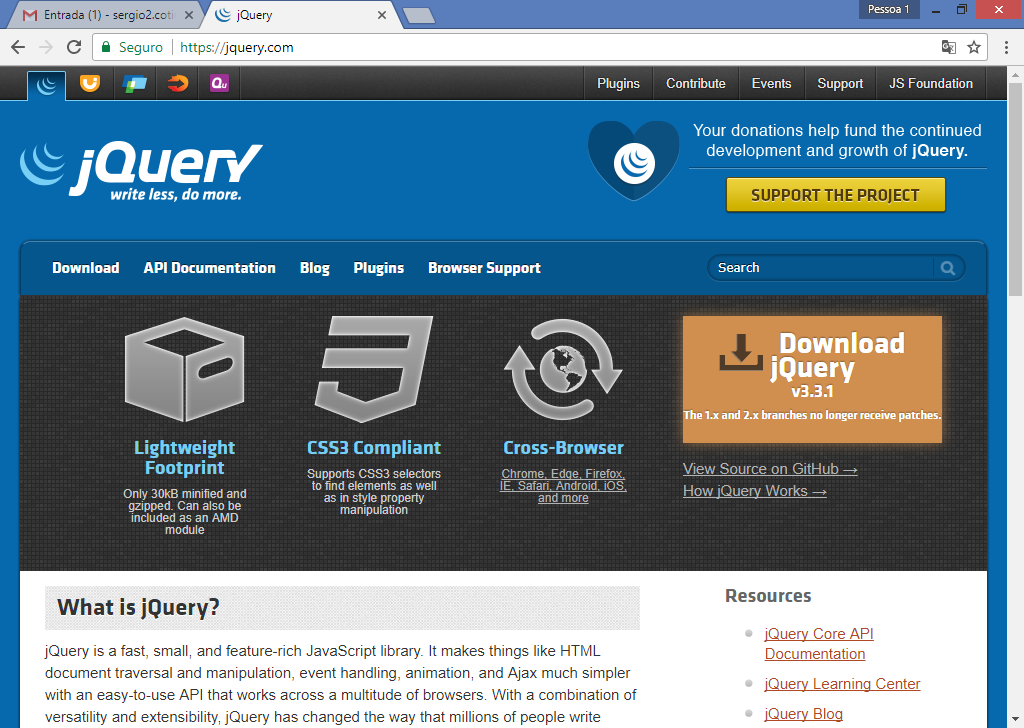




JQuery (https://jquery.com/)

Considerado a principal e mais utilizada biblioteca javascript do mercado.

Possui unimeras funções que facilitam e simplificam a escrita de código javascript.



<!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>

<h2>Formulário de Cadastro de Produto</h2>

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

<hr/>

<span id="texto"></span>

<br />

<br />

<div class="row">

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

<label>Nome do Produto:</label>

<input type="text" id="nome" class="form-control"

placeholder="Nome do Produto"/>

<br/>

<label>Preço:</label>

<input type="text" id="preco" class="form-control"

placeholder="Preço"/>

<br/>

<label>Quantidade:</label>

<input type="text" id="quantidade" class="form-control"

placeholder="Quantidade"/>

<br/>

<button class="btn btn-success">

Cadastrar Produto

</button>

</div>

</div>

</div>

**<!-- incluindo arquivos javascript.. -->**

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

**<!-- conteudo javascript.. -->**

**<script>**

**//função inicial (executada quando a página é carregada..)**

**$(document).ready(function () {**

**$("#texto").html("Por favor, preencha o formulario abaixo:");**

**});**

**</script>**

</body>

</html>



**Capturando os valores dos campos   
e armazenando em uma variavel JSON:**

<!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>

<h2>Formulário de Cadastro de Produto</h2>

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

<hr/>

<span id="texto"></span>

<br />

<br />

<div class="row">

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

<label>Nome do Produto:</label>

<input type="text" id="nome" class="form-control"

placeholder="Nome do Produto"/>

<br/>

<label>Preço:</label>

<input type="text" id="preco" class="form-control"

placeholder="Preço"/>

<br/>

<label>Quantidade:</label>

<input type="text" id="quantidade" class="form-control"

placeholder="Quantidade"/>

<br/>

<button id="btncadastro" class="btn btn-success">

Cadastrar Produto

</button>

<br />

<br />

<strong>

<span id="mensagem"></span>

</strong>

</div>

</div>

</div>

<!-- incluindo arquivos javascript.. -->

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

<!-- conteudo javascript.. -->

<script>

//função inicial (executada quando a página é carregada..)

$(document).ready(function () {

//imprimindo mensagem..

$("#texto").html("Por favor, preencha o formulario abaixo:");

//evento quando o botão for clicado..

$("#btncadastro").click(function () {

//ler e armazenar os dados digitados nos campos..

//JSON -> JavaScript Object Notation

var model = {

Nome: $("#nome").val(),

Preco: $("#preco").val(),

Quantidade: $("#quantidade").val()

};

//JSON.stringify

$("#mensagem").html(JSON.stringify(model));

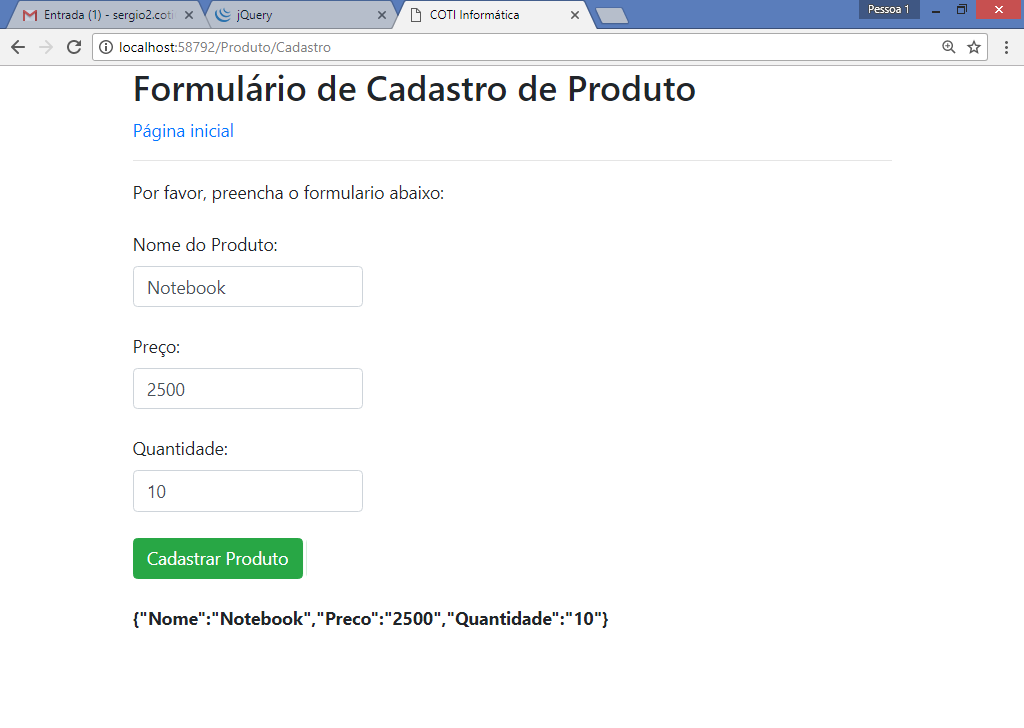
});

});

</script>

</body>

</html>



AJAX (Assyncronous JavaScript and XML)

Uso de Javascript para realizar requisições a um servidor web e obter respostas. Quando a comunicação entre uma página HTML e seu servidor é feita por meio de javascript, a resposta que o servidor devolve para a página apos a requisição não faz com que todo o conteudo da pagina seja recarregado.

Hoje, o uso do formato JSON para envio e recebimento de dados em AJAX esta sendo muito mais difundido do que originalmente o XML

<!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>

<h2>Formulário de Cadastro de Produto</h2>

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

<hr/>

<span id="texto"></span>

<br />

<br />

<div class="row">

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

<label>Nome do Produto:</label>

<input type="text" id="nome" class="form-control"

placeholder="Nome do Produto"/>

<br/>

<label>Preço:</label>

<input type="text" id="preco" class="form-control"

placeholder="Preço"/>

<br/>

<label>Quantidade:</label>

<input type="text" id="quantidade" class="form-control"

placeholder="Quantidade"/>

<br/>

<button id="btncadastro" class="btn btn-success">

Cadastrar Produto

</button>

<br />

<br />

<strong>

<span id="mensagem"></span>

</strong>

</div>

</div>

</div>

<!-- incluindo arquivos javascript.. -->

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

<!-- conteudo javascript.. -->

<script>

//função inicial (executada quando a página é carregada..)

$(document).ready(function () {

//imprimindo mensagem..

$("#texto").html("Por favor, preencha o formulario abaixo:");

//evento quando o botão for clicado..

$("#btncadastro").click(function () {

//ler e armazenar os dados digitados nos campos..

//JSON -> JavaScript Object Notation

var model = {

Nome: $("#nome").val(),

Preco: $("#preco").val(),

Quantidade: $("#quantidade").val()

};

//JSON.stringify

//$("#mensagem").html(JSON.stringify(model));

$.ajax({

type: "POST", //tipo da requisição..

url: "/Produto/CadastrarProduto", //método no controller..

data: model, //enviando o JSON..

success: function () {

},

error: function (e) {

//imprimir mensagem de erro..

$("#mensagem").html("Ocorreu um erro: " + e.status);

}

});

});

});

</script>

</body>

</html>

Testando:

**Erro 404 (Página ou recurso não encontrado)**



----------------------------

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

\*\* Esta classe model deverá ter as suas propriedades   
com o mesmo nome dos atributos do JSON.

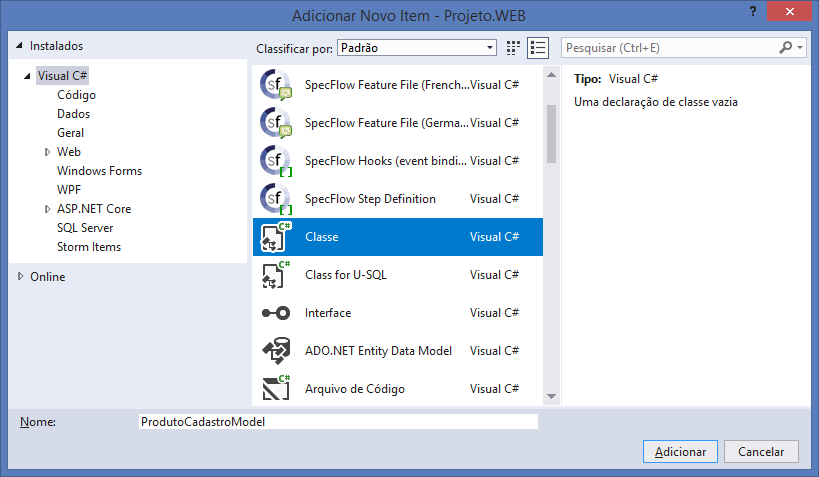
**var model = {**

**Nome: $("#nome").val(),**

**Preco: $("#preco").val(),**

**Quantidade: $("#quantidade").val()**

**};**



using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace Projeto.WEB.Models

{

public class ProdutoCadastroModel

{

public string Nome { get; set; }

public decimal Preco { get; set; }

public int Quantidade { get; set; }

}

}

----------------

var model = {

Nome: $("#nome").val(),

Preco: $("#preco").val(),

Quantidade: $("#quantidade").val()

};

**JsonResult**

Utilizado na classe de controle do MVC para declarar métodos que recebem e retornam dados em formato JSON. Ou seja, um método JsonResult é indicado para quando o controle esta recebendo uma requisição do tipo AJAX (Javasscript)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using Projeto.DAL.Entidades; //importando..

using Projeto.DAL.Repositorios; //importando..

using Projeto.WEB.Models; //classes de modelo..

namespace Projeto.WEB.Controllers

{

public class ProdutoController : Controller

{

// GET: Produto/Cadastro

public ActionResult Cadastro()

{

return View();

}

**//método executado pela requisição AJAX do jquery..**

**public JsonResult CadastrarProduto(ProdutoCadastroModel model)**

**{**

**try**

**{**

**Produto p = new Produto(); //classe de entidade..**

**p.Nome = model.Nome;**

**p.Preco = model.Preco;**

**p.Quantidade = model.Quantidade;**

**//gravar na base de dados..**

**ProdutoRepositorio rep = new ProdutoRepositorio();**

**rep.Insert(p); //executando..**

**//retornando mensagem de sucesso..**

**return Json($"Produto {p.Nome}, cadastrado com sucesso.");**

**}**

**catch(Exception e)**

**{**

**//retornando mensagem de erro..**

**return Json(e.Message);**

**}**

**}**

}

}

**Na página:**

<!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>

<h2>Formulário de Cadastro de Produto</h2>

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

<hr/>

<span id="texto"></span>

<br />

<br />

<div class="row">

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

<label>Nome do Produto:</label>

<input type="text" id="nome" class="form-control"

placeholder="Nome do Produto"/>

<br/>

<label>Preço:</label>

<input type="text" id="preco" class="form-control"

placeholder="Preço"/>

<br/>

<label>Quantidade:</label>

<input type="text" id="quantidade" class="form-control"

placeholder="Quantidade"/>

<br/>

<button id="btncadastro" class="btn btn-success">

Cadastrar Produto

</button>

<br />

<br />

<strong>

<span id="mensagem"></span>

</strong>

</div>

</div>

</div>

**<!-- incluindo arquivos javascript.. -->**

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

**<!-- conteudo javascript.. -->**

**<script>**

**//função inicial (executada quando a página é carregada..)**

**$(document).ready(function () {**

**//imprimindo mensagem..**

**$("#texto").html("Por favor, preencha o formulario abaixo:");**

**//evento quando o botão for clicado..**

**$("#btncadastro").click(function () {**

**//ler e armazenar os dados digitados nos campos..**

**//JSON -> JavaScript Object Notation**

**var model = {**

**Nome: $("#nome").val(),**

**Preco: $("#preco").val(),**

**Quantidade: $("#quantidade").val()**

**};**

**//JSON.stringify**

**//$("#mensagem").html(JSON.stringify(model));**

**$.ajax({**

**type: "POST", //tipo da requisição..**

**url: "/Produto/CadastrarProduto", //método no controller..**

**data: model, //enviando o JSON..**

**success: function (msg) {**

**//imprimir mensagem..**

**$("#mensagem").html(msg);**

**//limpar o conteudo de cada campo..**

**$("#nome").val("");**

**$("#preco").val("");**

**$("#quantidade").val("");**

**},**

**error: function (e) {**

**//imprimir mensagem de erro..**

**$("#mensagem").html("Ocorreu um erro: " + e.status);**

**}**

**});**

**});**

**});**

**</script>**

</body>

</html>

Executando:



No banco de dados:

