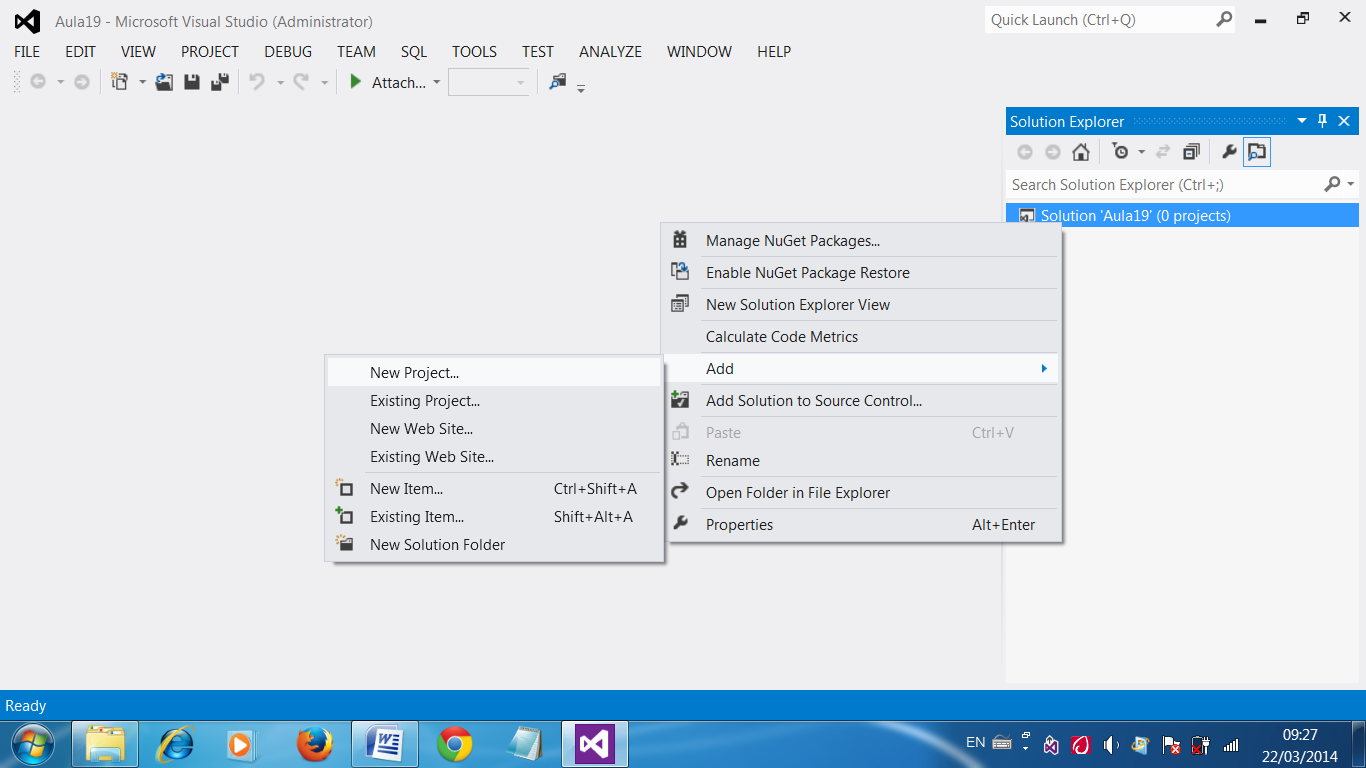
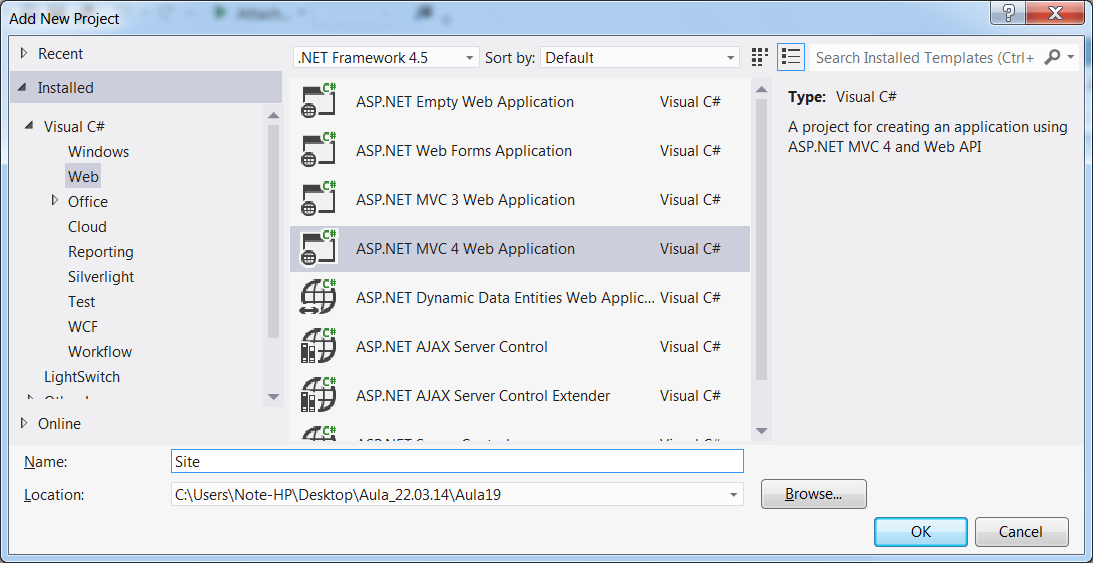


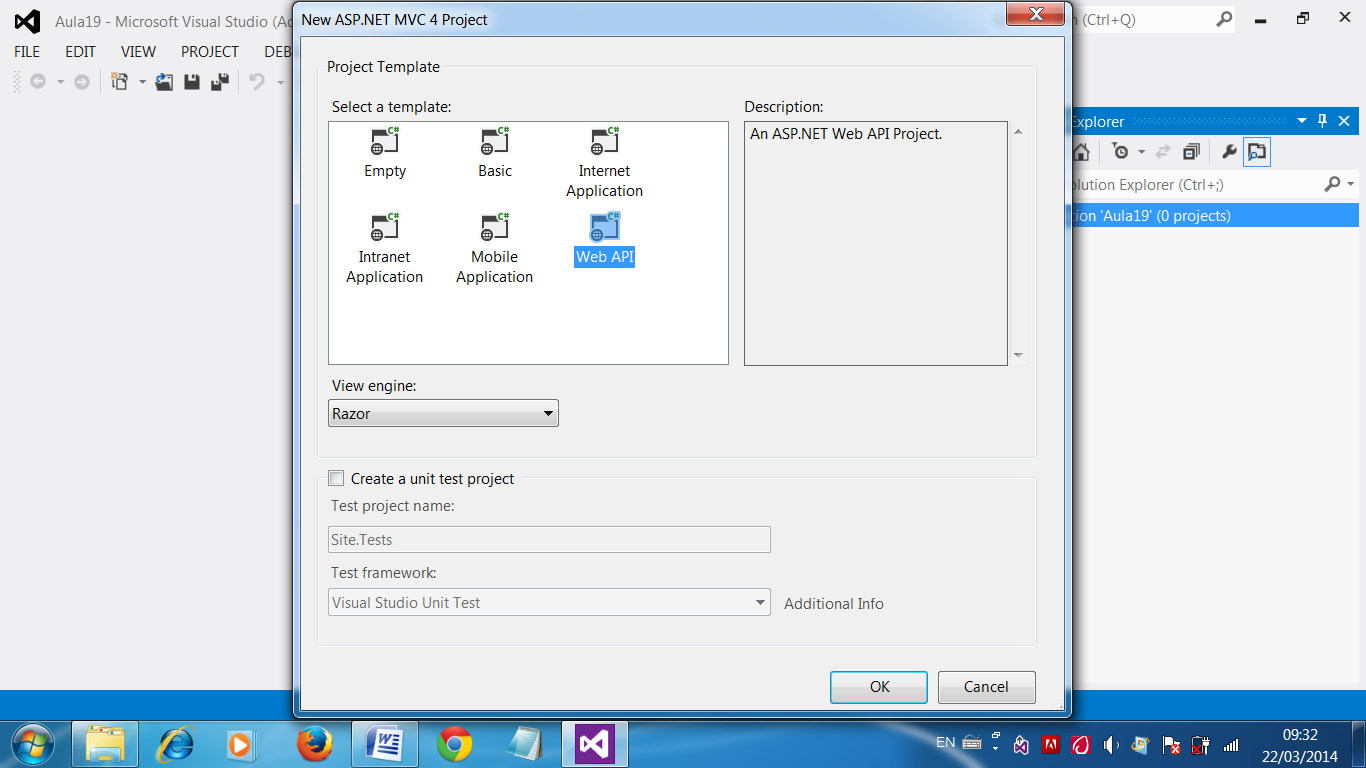
Criando o Projeto MVC4



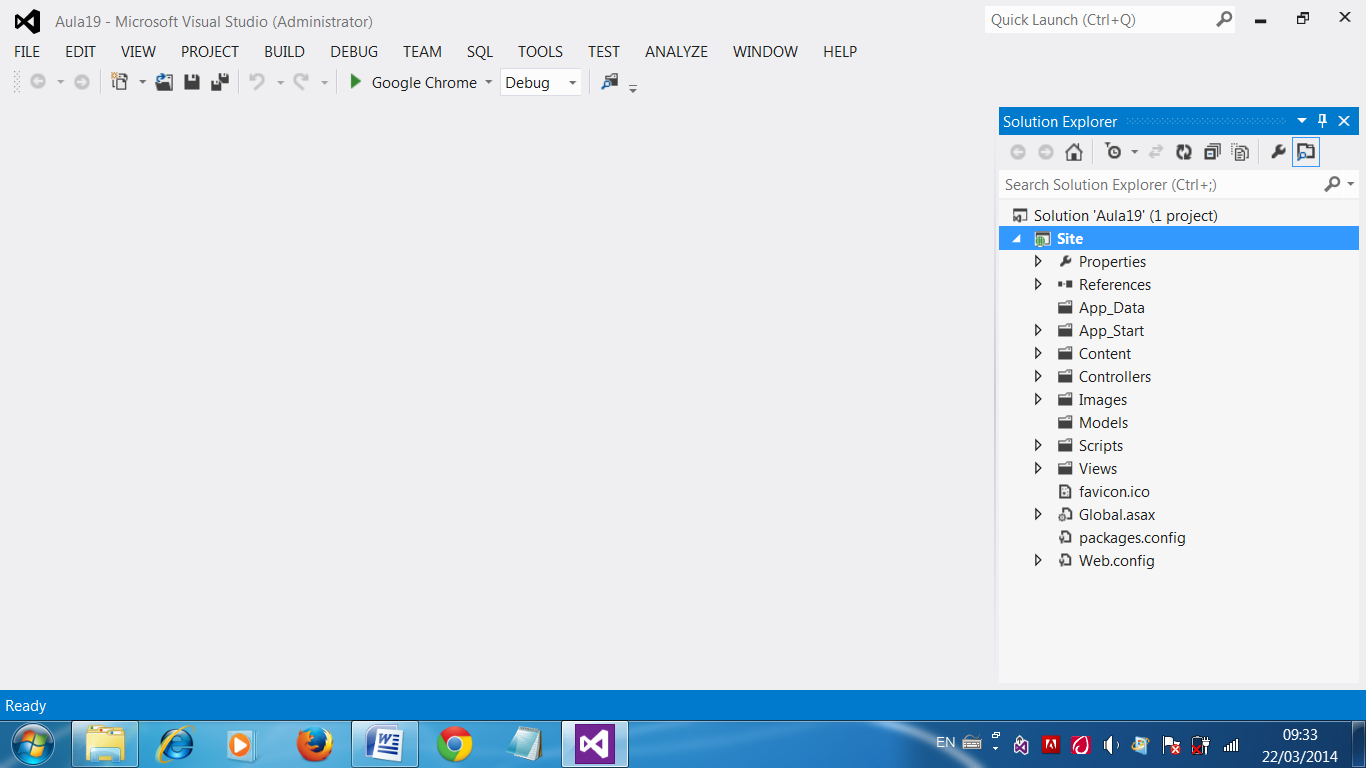


Web API

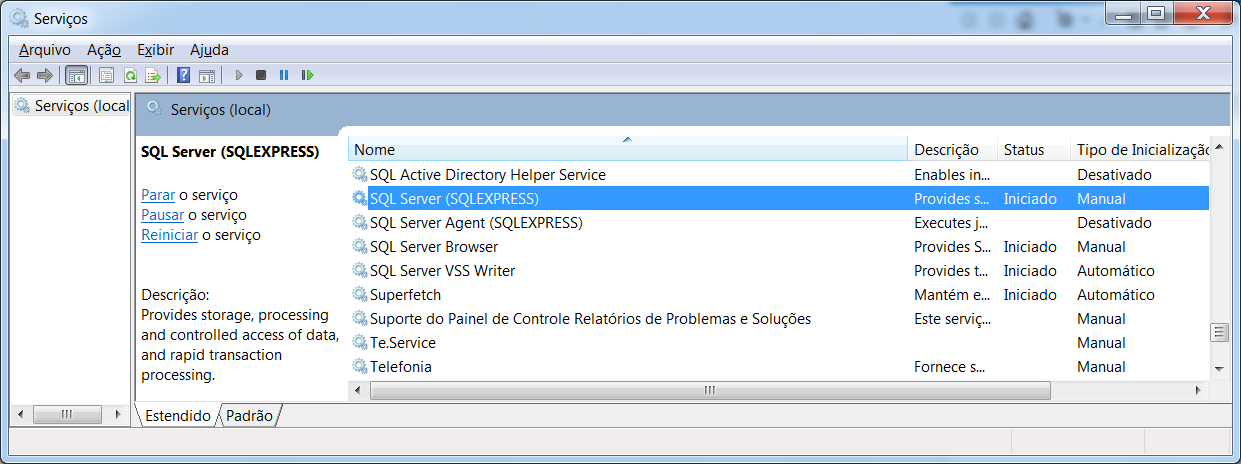
Projeto Web capaz de fornecer dados para aplicações externas através de endereços web que retornam XML ou JSON.

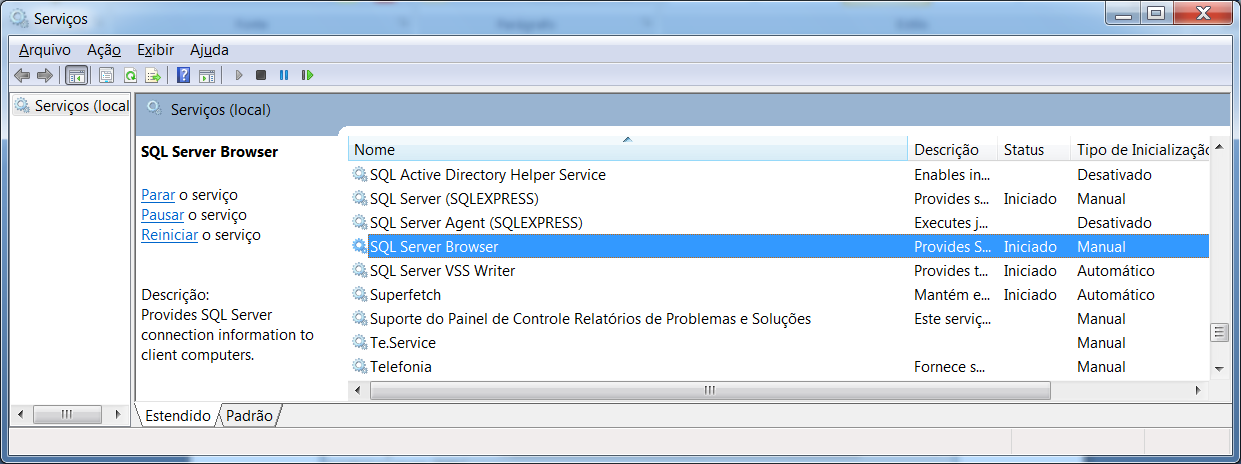


* Estrutura do Projeto...

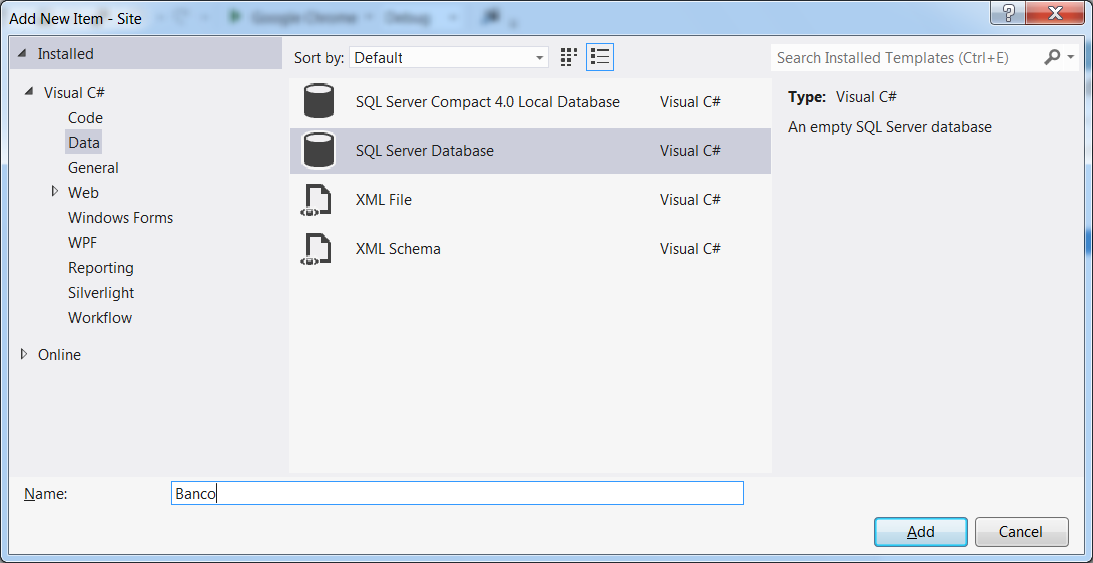


Base de dados...

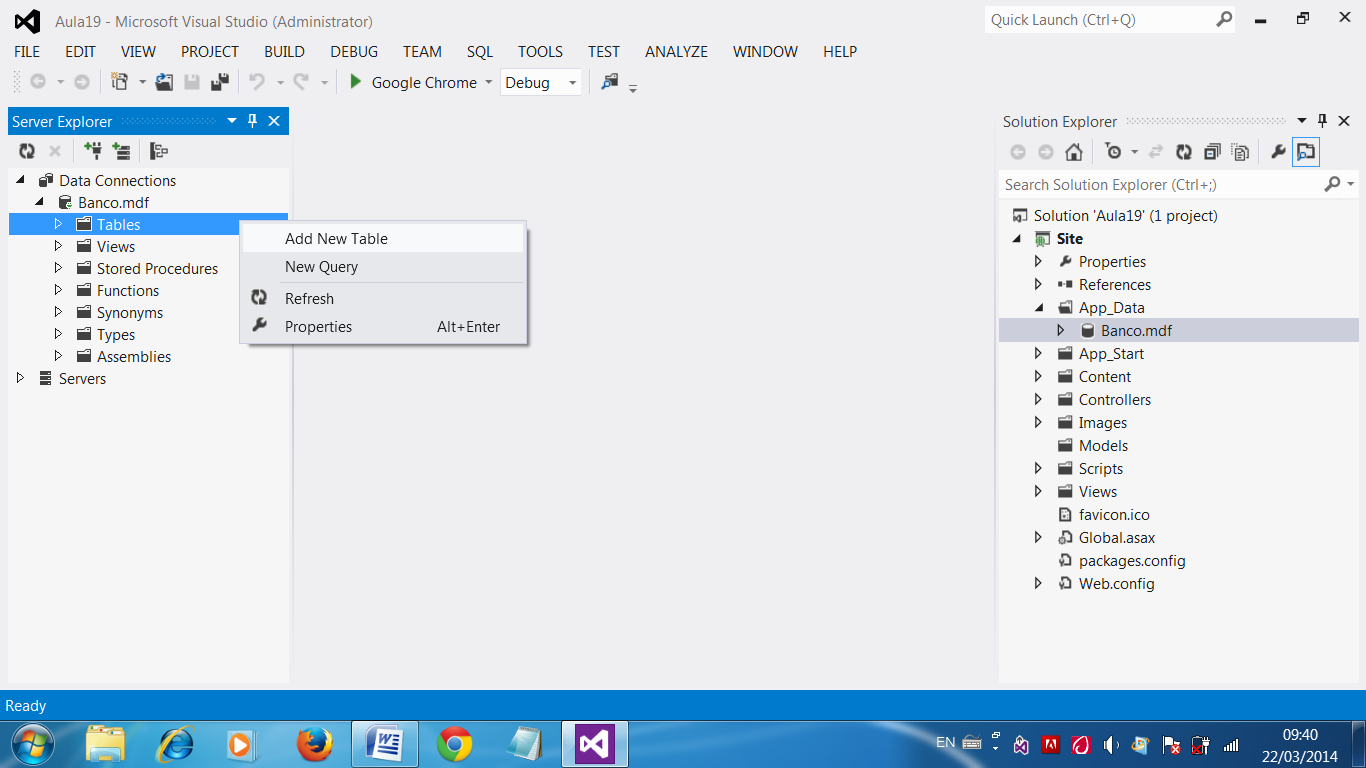


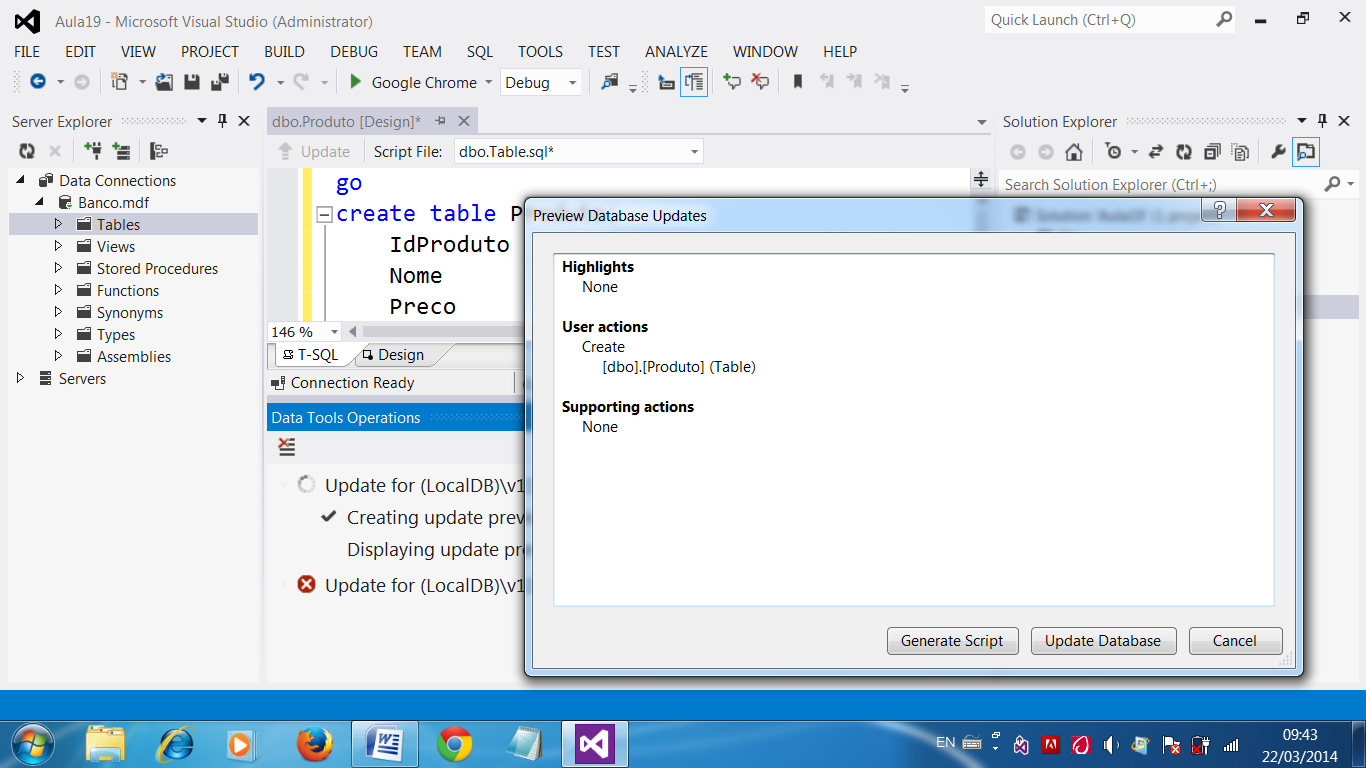


Criando a base de dados...



Criando a tabela de Produtos...





go

create table Produto(

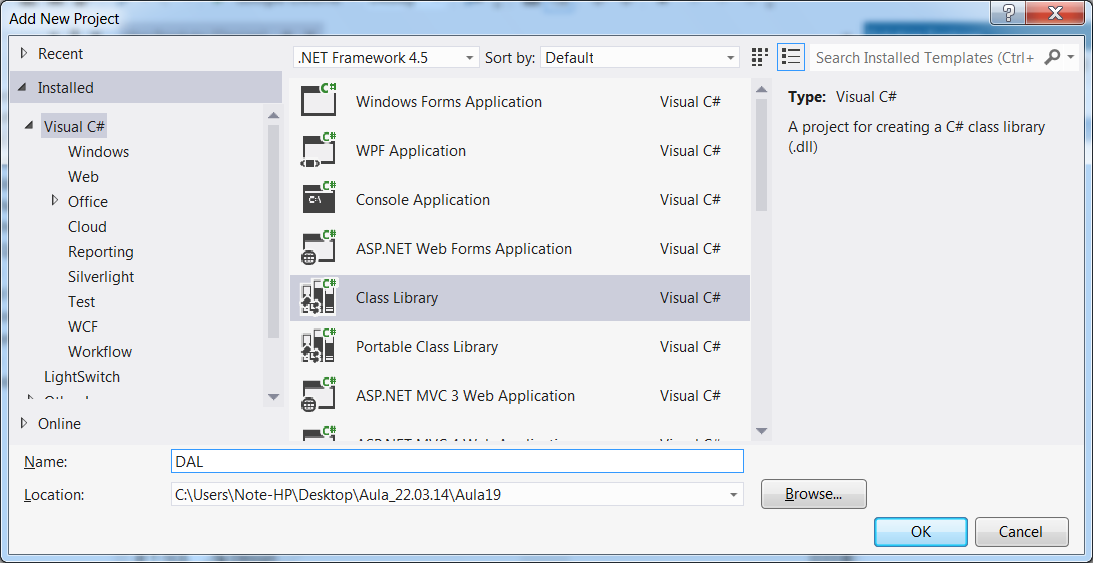
IdProduto integer identity,

Nome nvarchar(50) not null,

Preco decimal(10,2) not null,

Quantidade integer not null,

primary key(IdProduto))

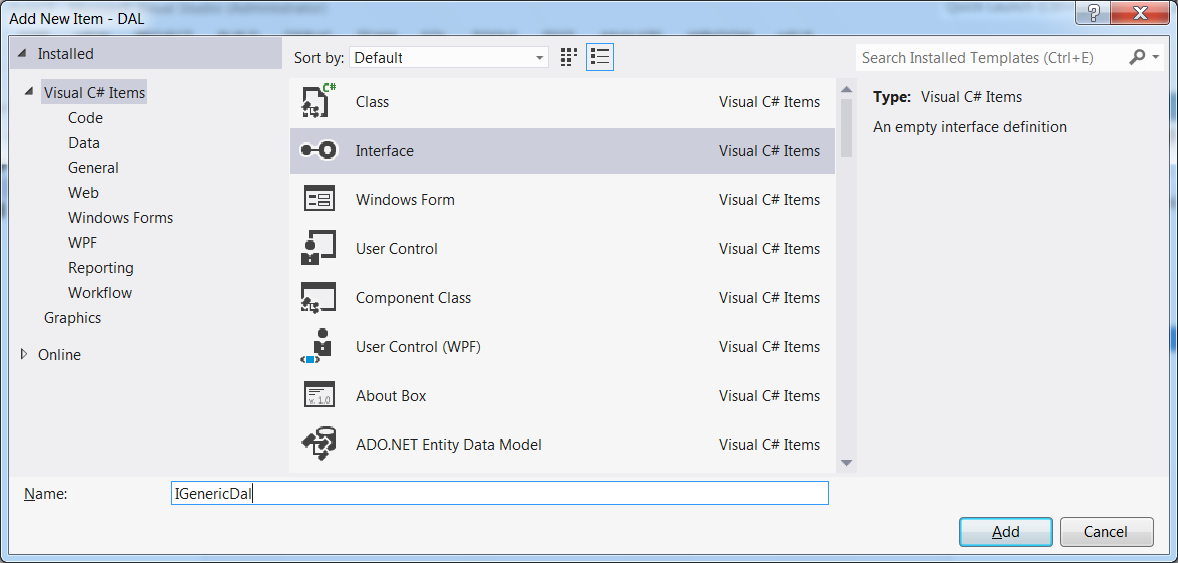


Pattern GenericDal

Padrão de projeto que permita à aplicação disponibilizar uma camada DAL com as operações principais de banco de dados como inserir, excluir, editar e consultar já implementadas de maneira genérica, ou seja, de forma que qualquer classe possa utilizá-las.

**Passo 1:**

Criar uma interface para definir quais operações da DAL serão generalizadas.



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DAL.Generics

{

public interface IGenericDal<T, K> //T e K são genericos

where T : class //T é um generico de Classe

where K : struct //K é um generico de Estrutura (Tipo)

{

void Inserir(T entidade);

void Excluir(T entidade);

void Salvar();

List<T> ListarTodos();

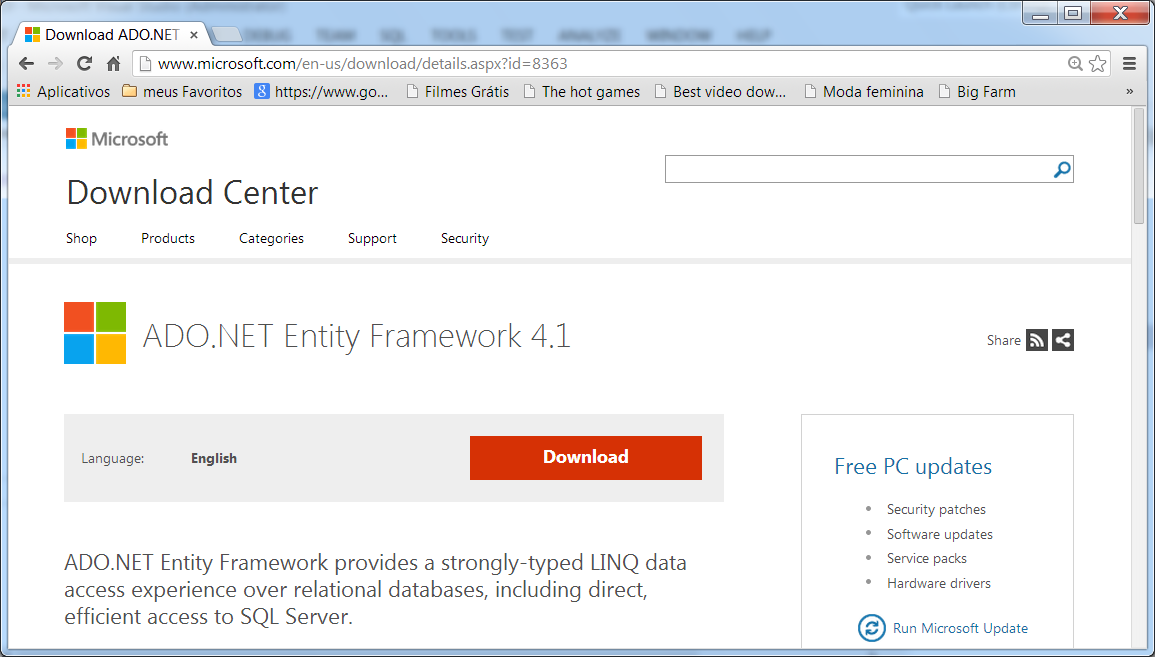
T ObterPorId(K Id);

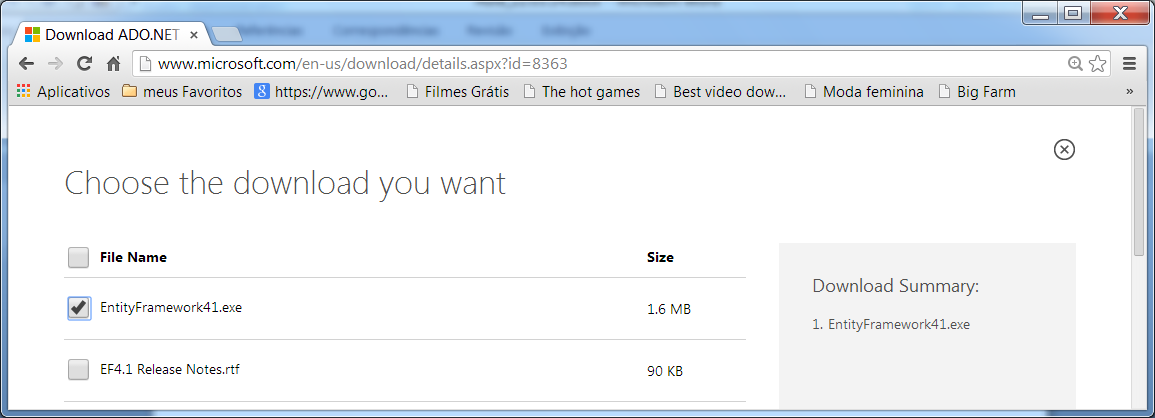
}

}

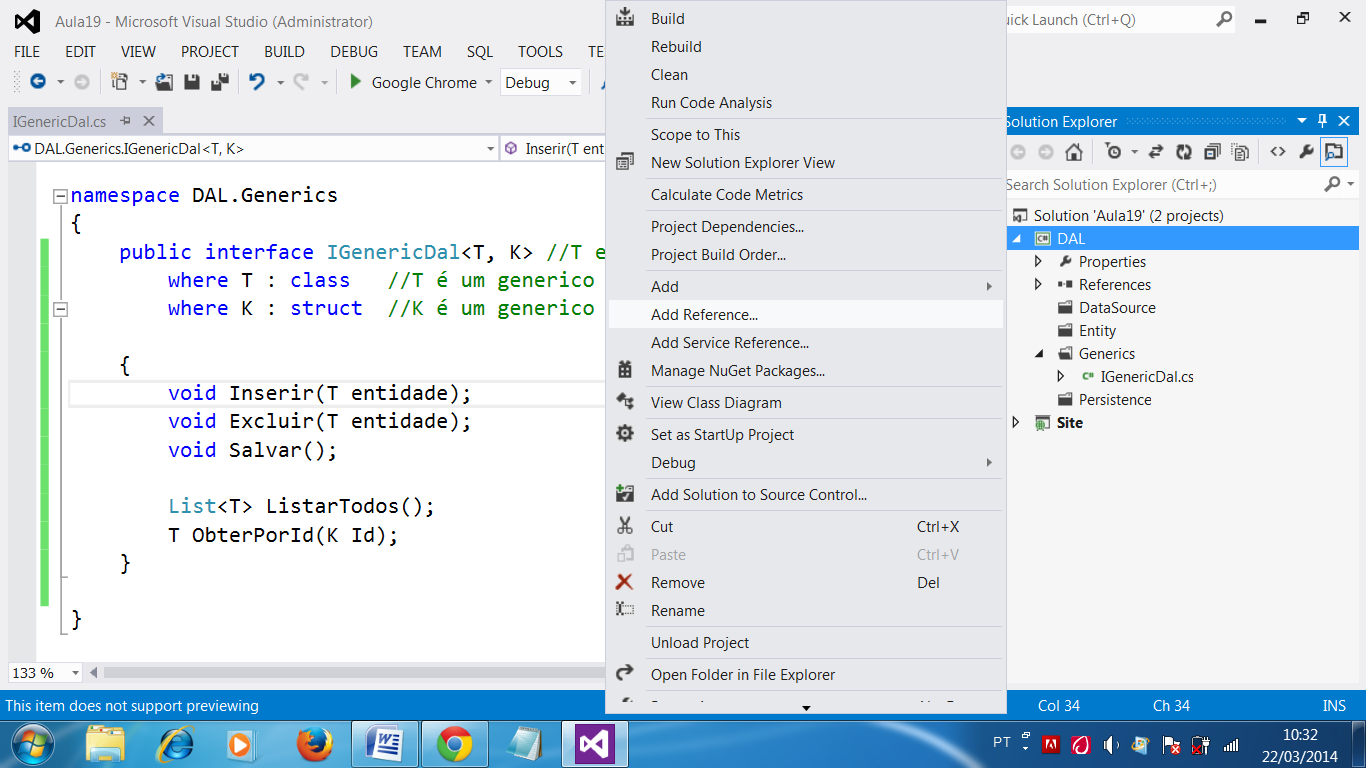
**Passo 2:**

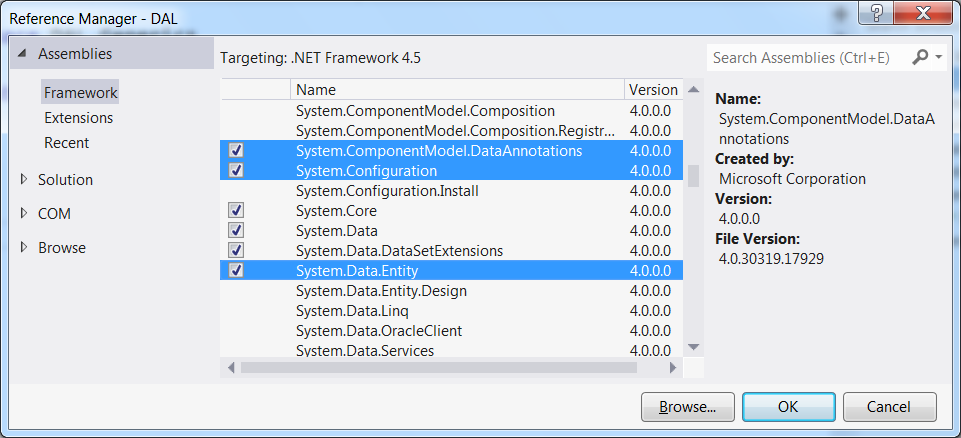
Criar uma Classe que implemente a interface ainda sem definir o tipo que será utilizado nos métodos, mas já com o EntityFramework

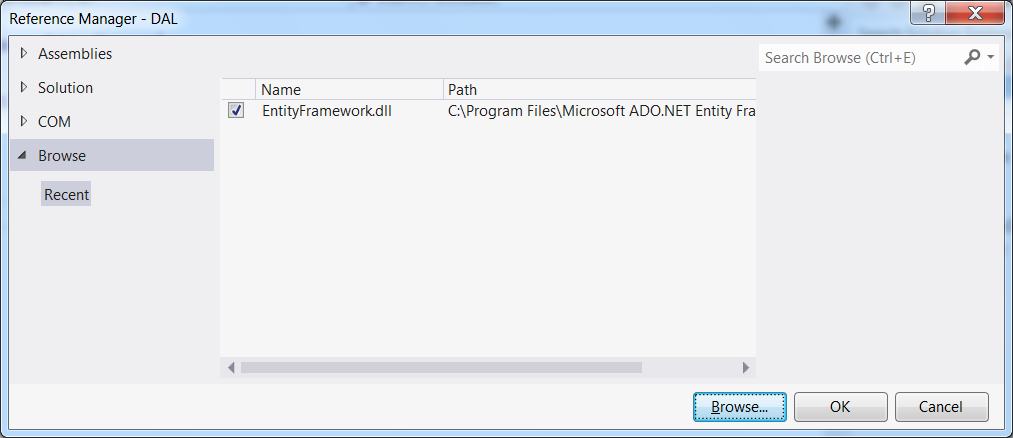


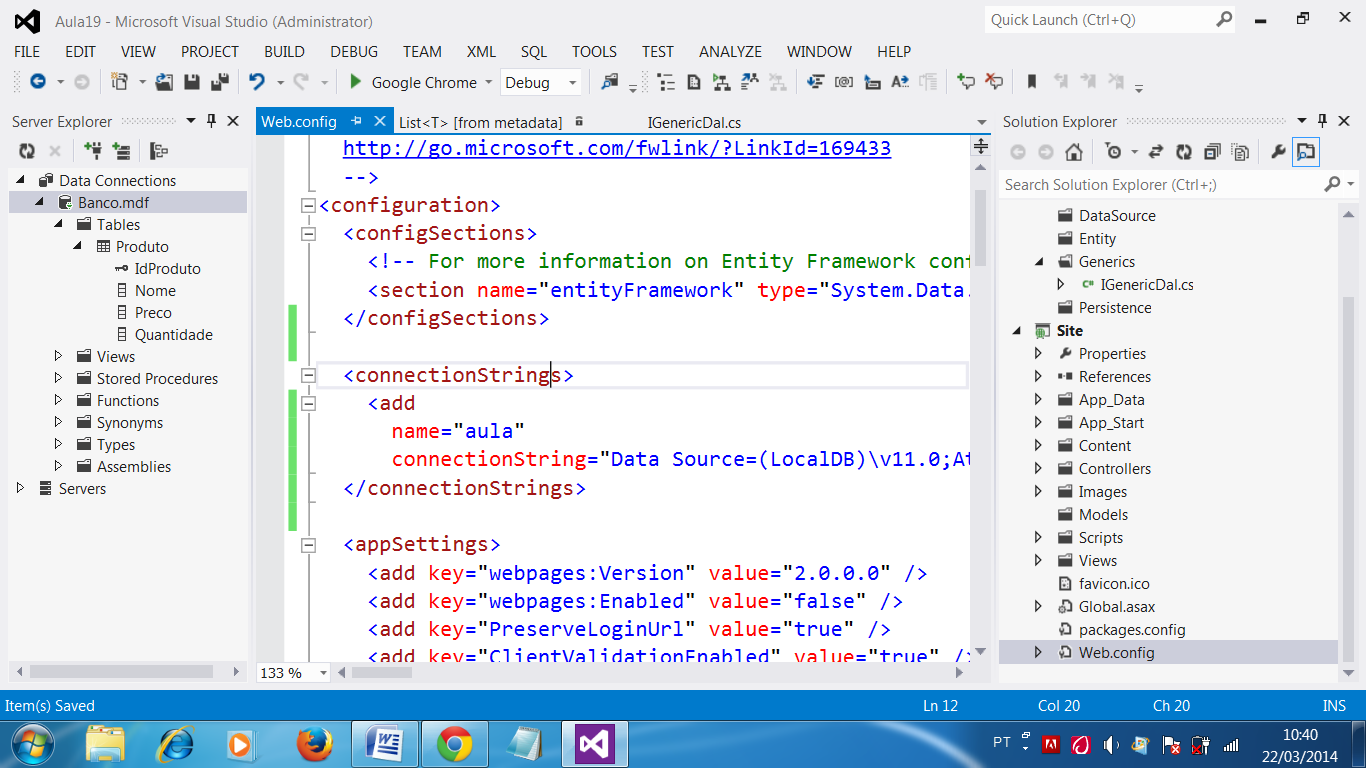


Adicionando bibliotecas no projeto...









<connectionStrings>

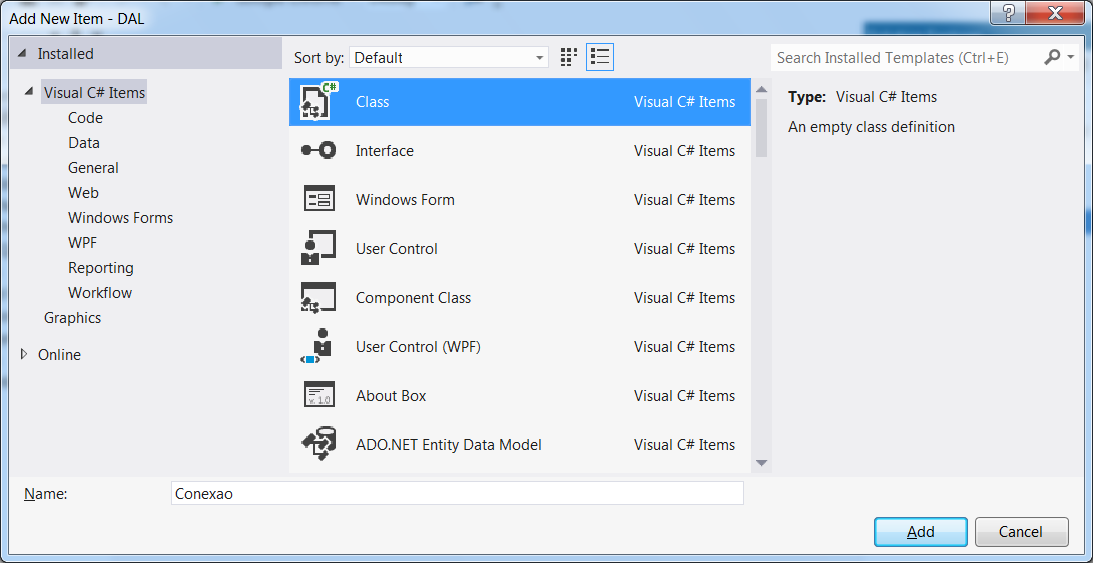
<add

**name="aula"**

connectionString="Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Note-HP\Desktop\Aula\_22.03.14\Aula19\Site\App\_Data\Banco.mdf;Integrated Security=True" />

</connectionStrings>

Criando a Classe de conexão com o banco...



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.Entity;

using System.Configuration;

namespace DAL.DataSource

{

public class Conexao : DbContext //DbContext -> Classe do EF

//para acesso ao BD

{

//Construtor default (vazio)

public Conexao()

: base(ConfigurationManager.ConnectionStrings["aula"].

ConnectionString)

{

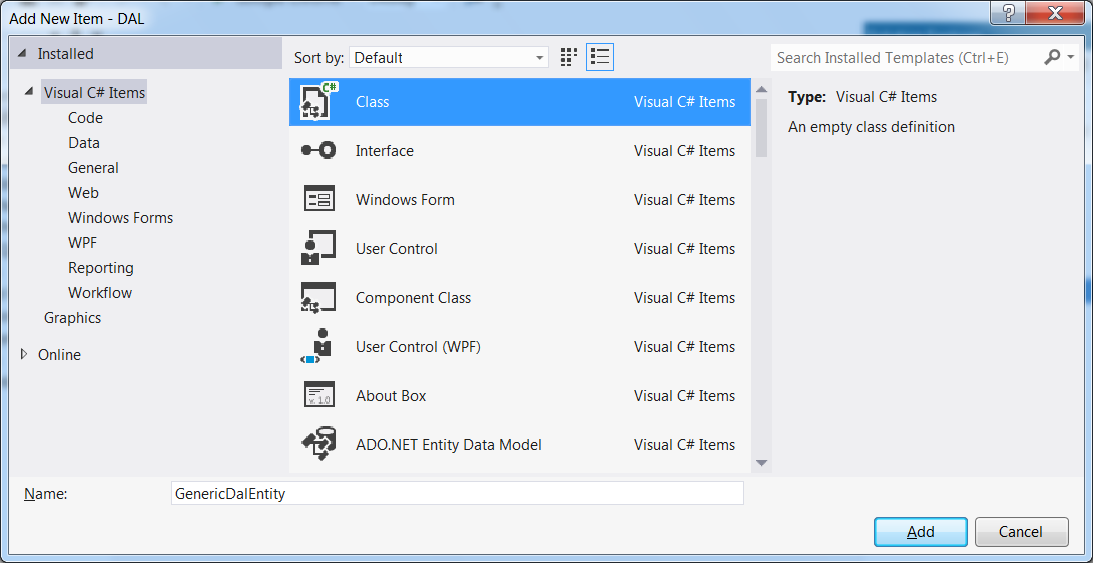
//Passando o caminho da connectionstring

}

}

}

Implementando o DAL Genérico sem entidades



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DAL.DataSource; //conexão

namespace DAL.Generics

{

//Classe abstrata -> não pode ser instanciada, apenas herdada

public abstract class GenericDalEntity<T, K> : IGenericDal<T, K>

where T : class

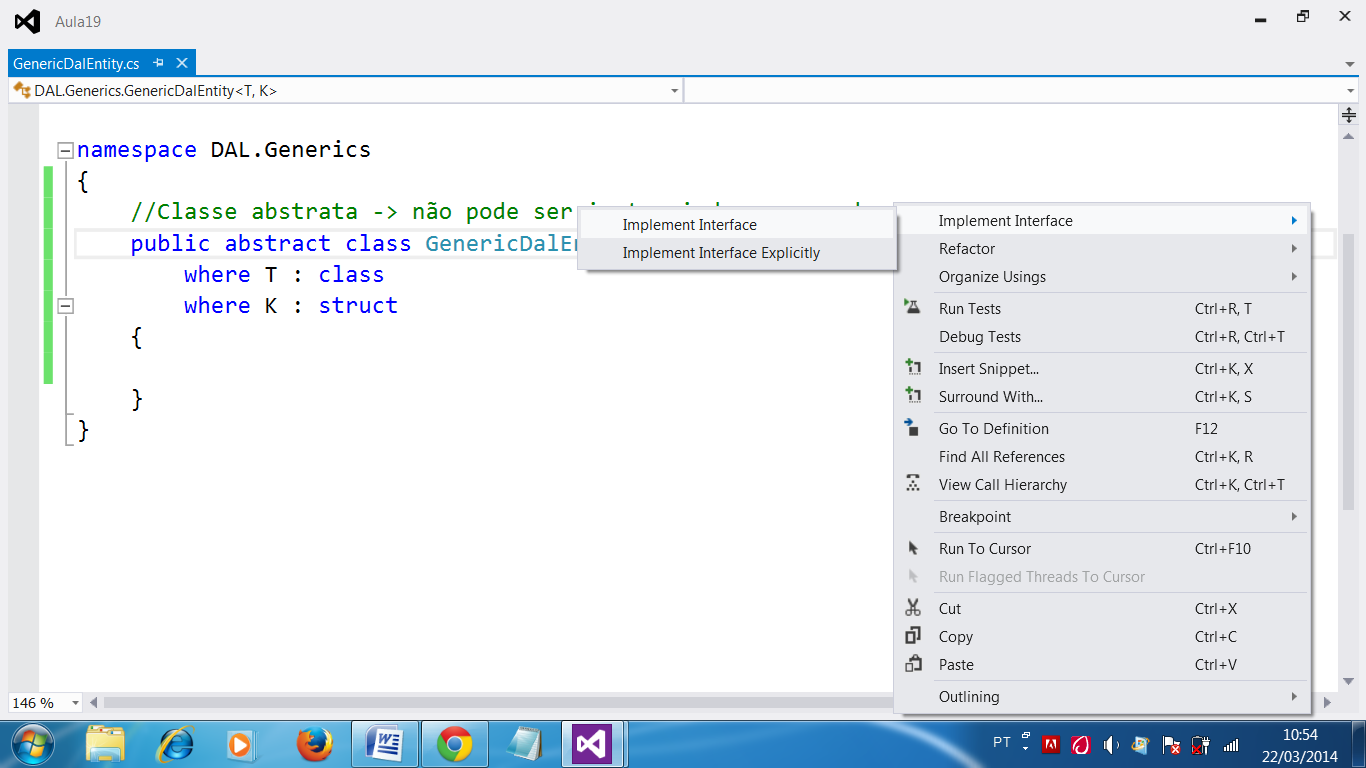
where K : struct

{

}

}

Implementando a interface...



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DAL.DataSource; //conexão

namespace DAL.Generics

{

//Classe abstrata -> não pode ser instanciada, apenas herdada

public abstract class GenericDalEntity<T, K> : IGenericDal<T, K>, IDisposable

where T : class

where K : struct

{

//Atributo para a classe de conexão

protected Conexao Con; //null

public GenericDalEntity() //Construtor [ctor] + 2x[tab]

{

Con = new Conexao(); //inicializo o atributo de conexão

}

public void Inserir(T entidade)

{

try

{

Con.Set<T>().Add(entidade);

}

catch

{

throw;

}

}

public void Excluir(T entidade)

{

try

{

Con.Set<T>().Remove(entidade);

}

catch

{

throw;

}

}

public void Salvar()

{

Con.SaveChanges();

}

public List<T> ListarTodos()

{

try

{

return Con.Set<T>().ToList();

}

catch

{

throw;

}

}

public T ObterPorId(K Id)

{

try

{

return Con.Set<T>().Find(Id);

}

catch

{

throw;

}

}

public void Dispose()

{

Con.Dispose(); //destruir a conexão

}

}

}

Incluir no Projeto a entidade Produto...

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.ComponentModel.DataAnnotations; //Mapeamento

namespace DAL.Entity

{

[Table("Produto")]

public class Produto

{

[Key]

[DatabaseGenerated(DatabaseGeneratedOption.Identity)]

[Column]

public int IdProduto { get; set; }

[Column]

public string Nome { get; set; }

[Column]

public decimal Preco { get; set; }

[Column]

public int Quantidade { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.Entity;

using System.Configuration;

using DAL.Entity;

namespace DAL.DataSource

{

public class Conexao : DbContext

//DbContext -> Classe do EF para acesso ao BD

{

//Construtor default (vazio)

public Conexao()

: base(ConfigurationManager.ConnectionStrings

["aula"].ConnectionString)

{

//Passando o caminho da connectionstring

}

//Agregar a Classe de entidade -> Produto

**public DbSet<Produto> Produto { get; set; }**

}

}

Criando a Classe de persistência...

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DAL.Entity;

using DAL.Generics;

namespace DAL.Persistence

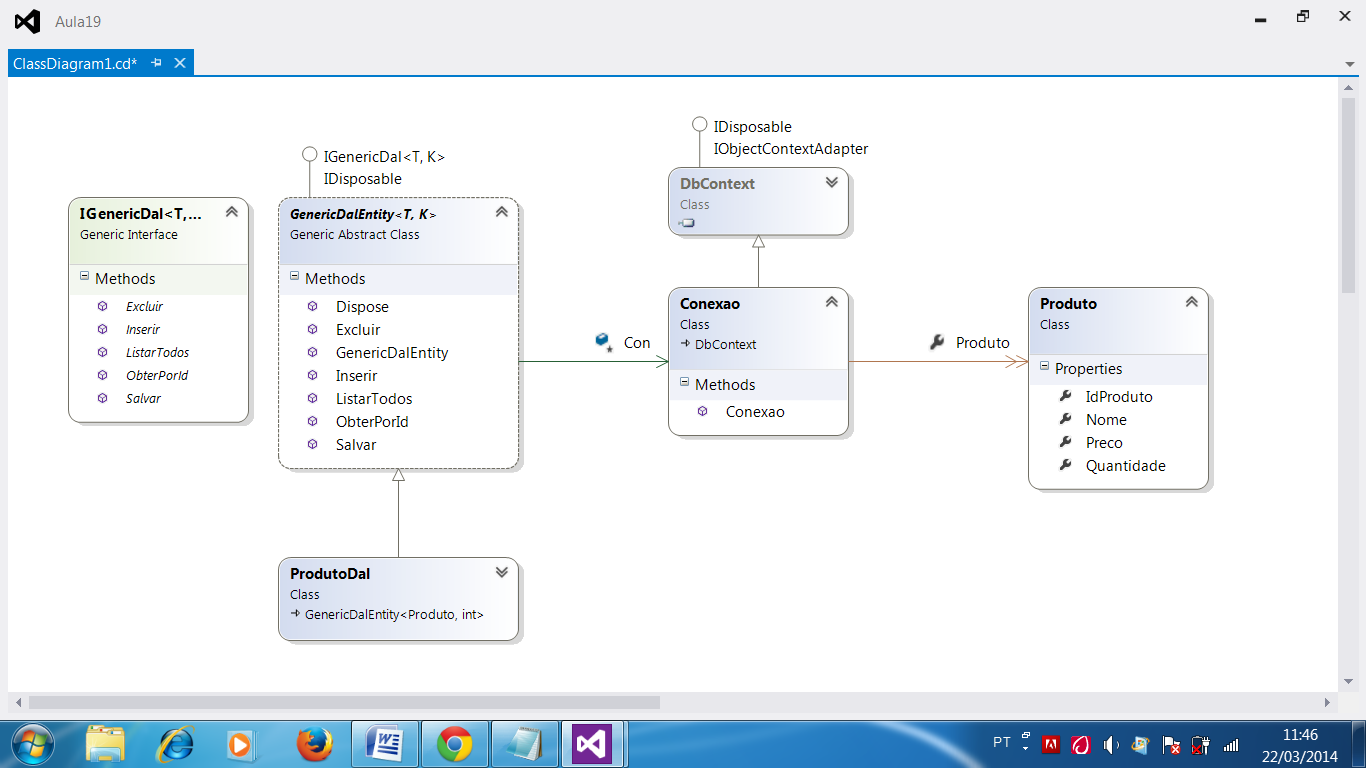
{

**public class ProdutoDal : GenericDalEntity<Produto, Int32>**

{

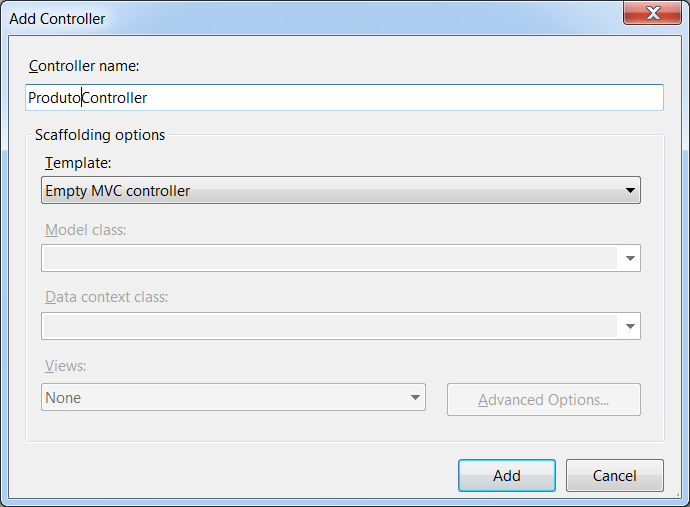
}

}



Criando um WebAPI

URL para obter dados de uma aplicação utilizando JSON.



using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Http;

using DAL.Entity;

using DAL.Persistence;

namespace Site.Controllers

{

//API -> Classe para geração de conteudo para outros projetos

public class ProdutoController : ApiController

{

//Método para retornar os produtos que serão acessados por URL

public List<Produto> GetProdutos() //Nome padrão

{

try

{

using(ProdutoDal d = new ProdutoDal())

{

return d.ListarTodos();

//retornar uma lista de produtos do banco

}

}

catch

{

throw;

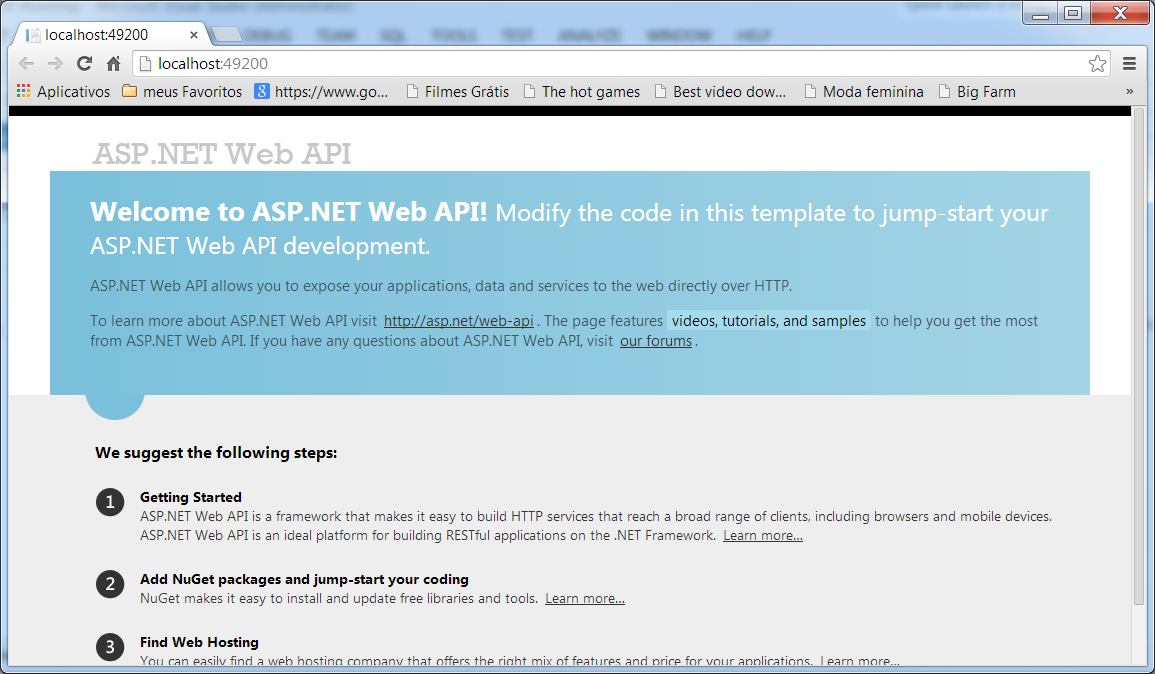
}

}

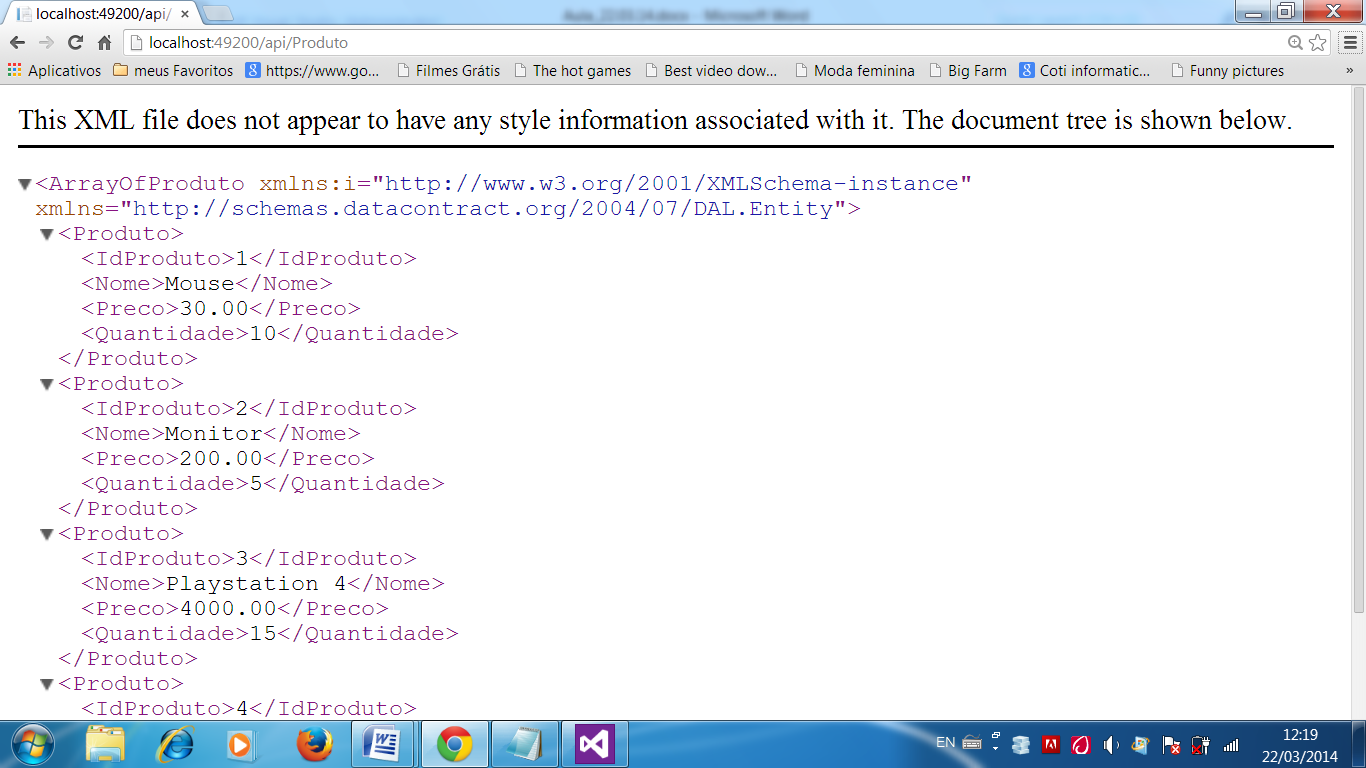
}

}

Executando...



[http://localhost:49200**/api/Produto**](http://localhost:49200/api/Produto)



Configurar o projeto para retornar conteúdo

em formato Json...

**Global.asax**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Http;

using System.Web.Mvc;

using System.Web.Optimization;

using System.Web.Routing;

using System.Net.Http.Formatting;

namespace Site

{

// Note: For instructions on enabling IIS6 or IIS7 classic mode,

// visit http://go.microsoft.com/?LinkId=9394801

public class WebApiApplication : System.Web.HttpApplication

{

protected void Application\_Start()

{

AreaRegistration.RegisterAllAreas();

WebApiConfig.Register(GlobalConfiguration.Configuration);

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

RouteConfig.RegisterRoutes(RouteTable.Routes);

BundleConfig.RegisterBundles(BundleTable.Bundles);

GlobalConfiguration.Configuration.Formatters.JsonFormatter.

MediaTypeMappings.Add(new QueryStringMapping

("json", "true", "application/json"));

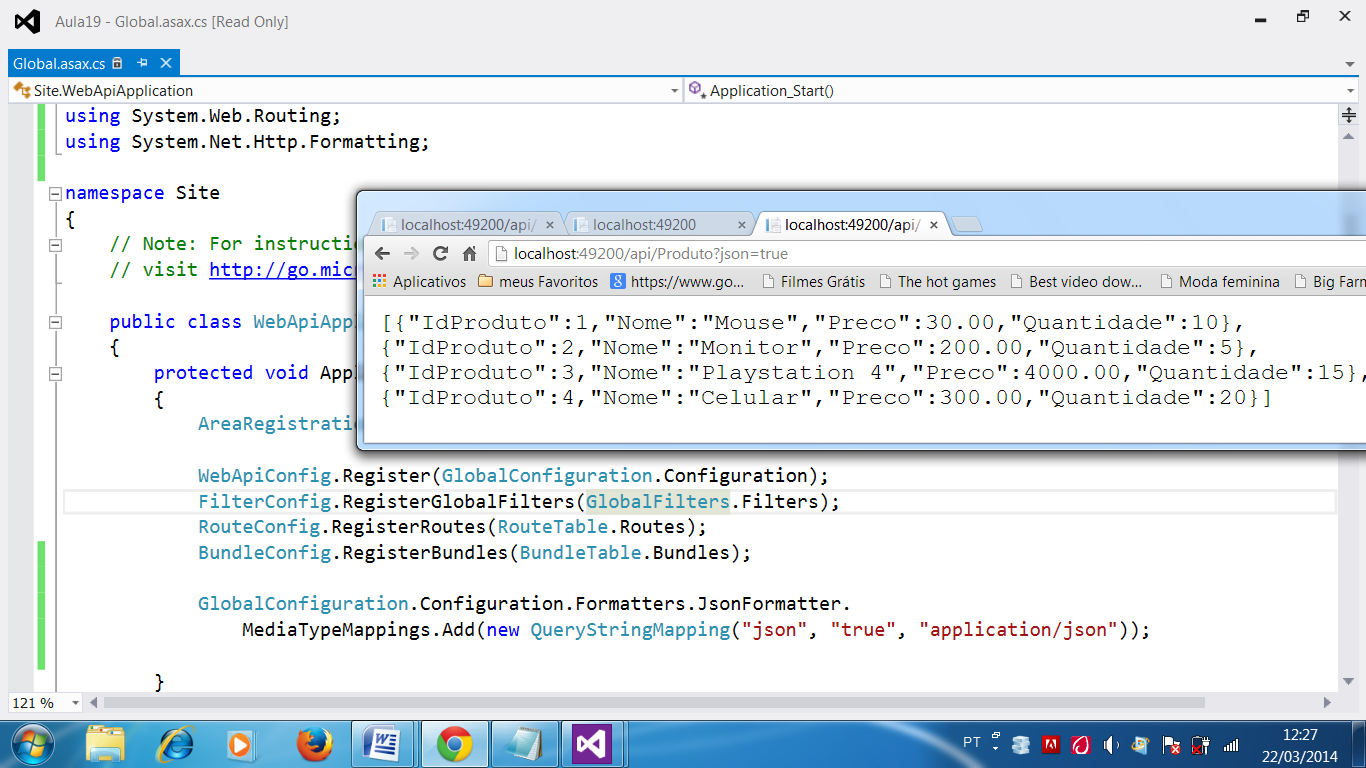
}

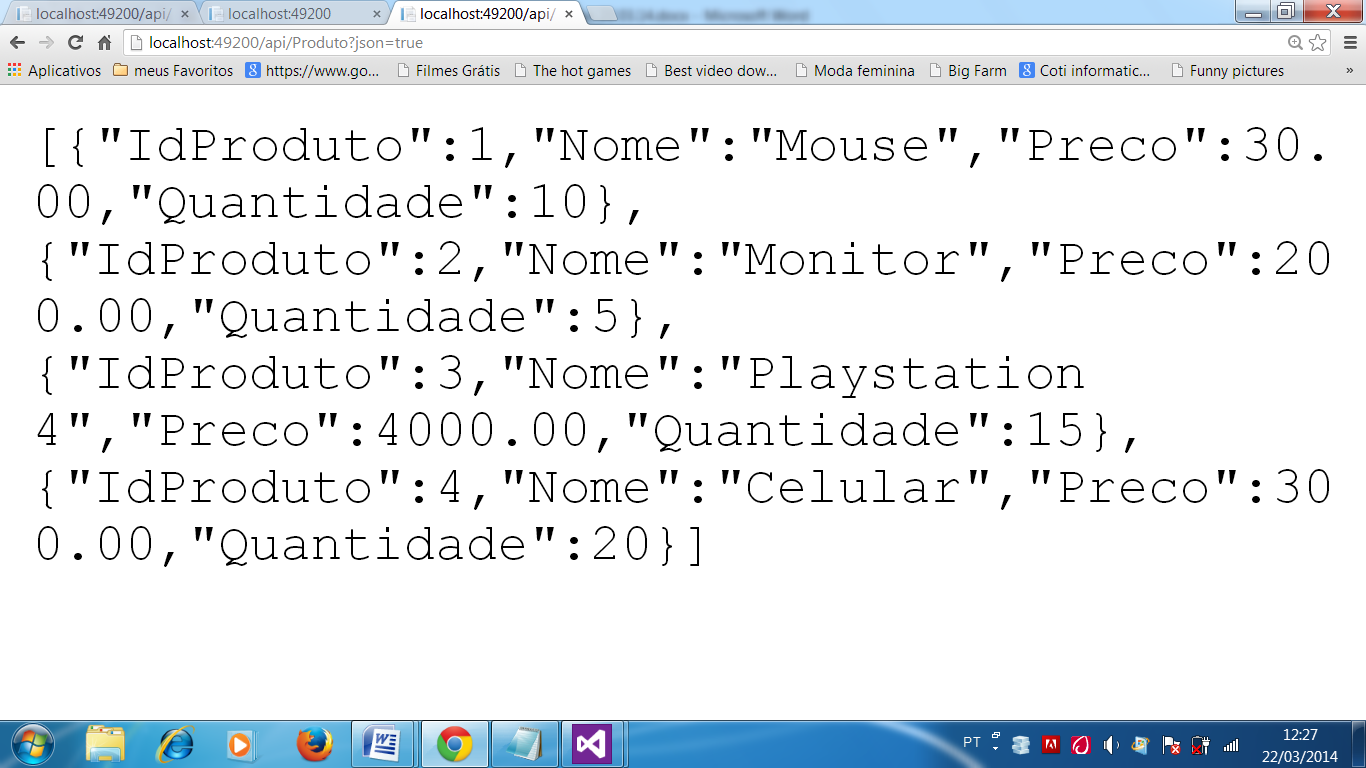
}

}

<http://localhost:49200/api/Produto?json=true>

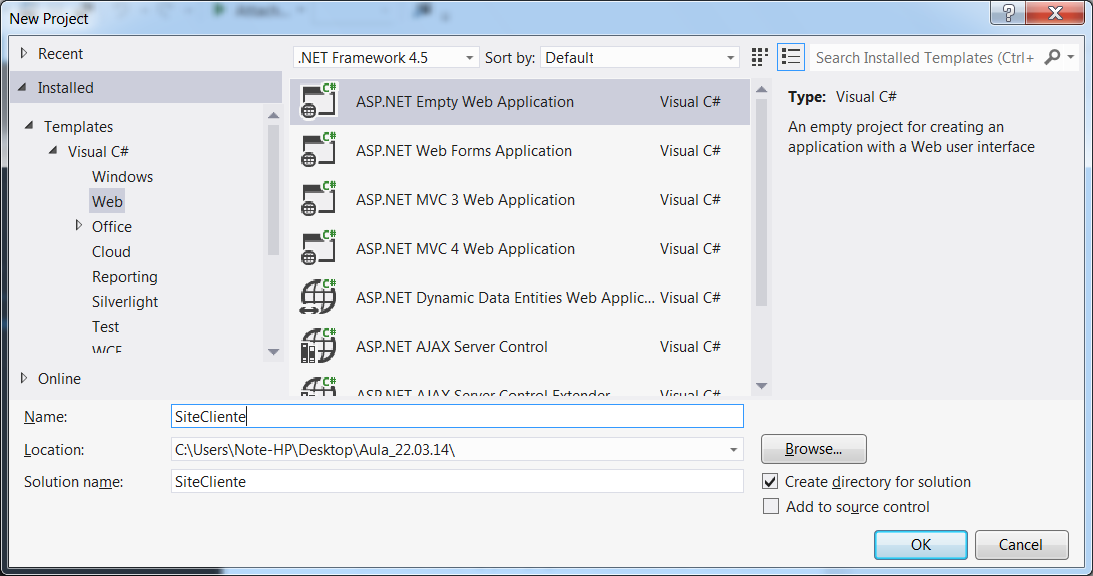
]

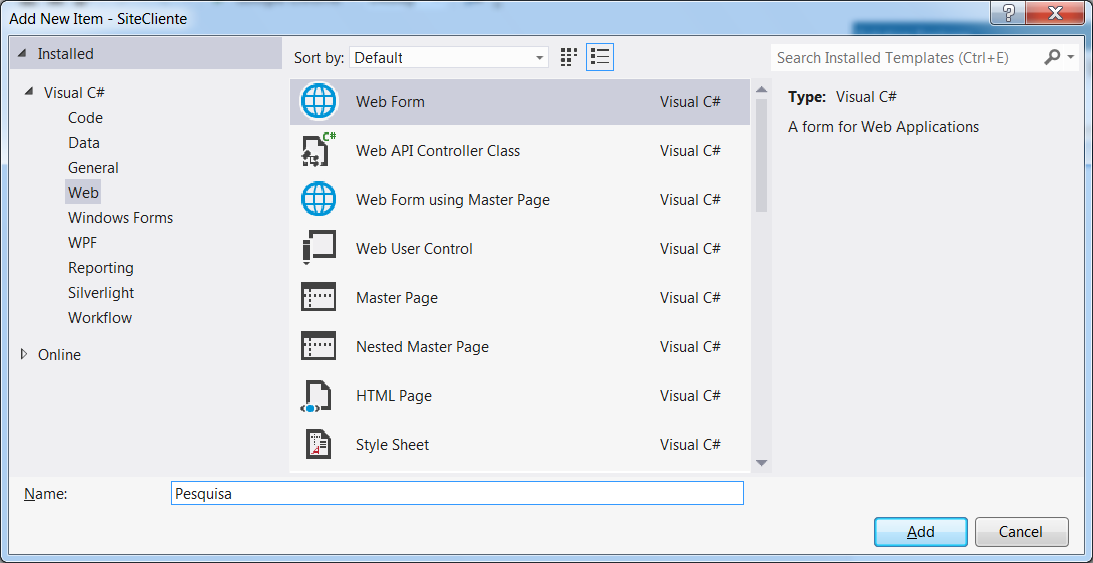




Criando o Cliente da WebAPI

Projeto para consumir os dados da WebAPI





<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Pesquisa.aspx.cs" Inherits="SiteCliente.Pesquisa" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<h3>Pesquisa de Produtos por WebAPI</h3>

<asp:GridView ID="gridProdutos" runat="server" Width="100%">

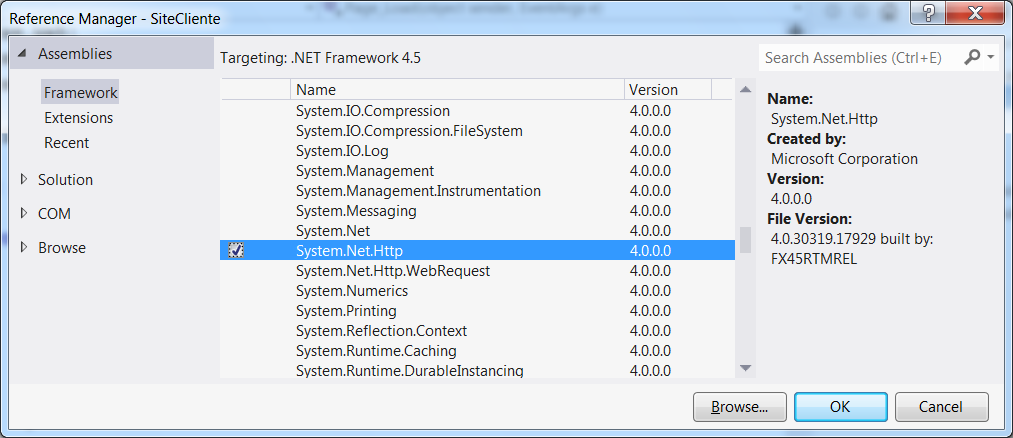
</asp:GridView>

</div>

</form>

</body>

</html>



using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace SiteCliente.Model

{

public class Produto

{

public int IdProduto { get; set; }

public string Nome { get; set; }

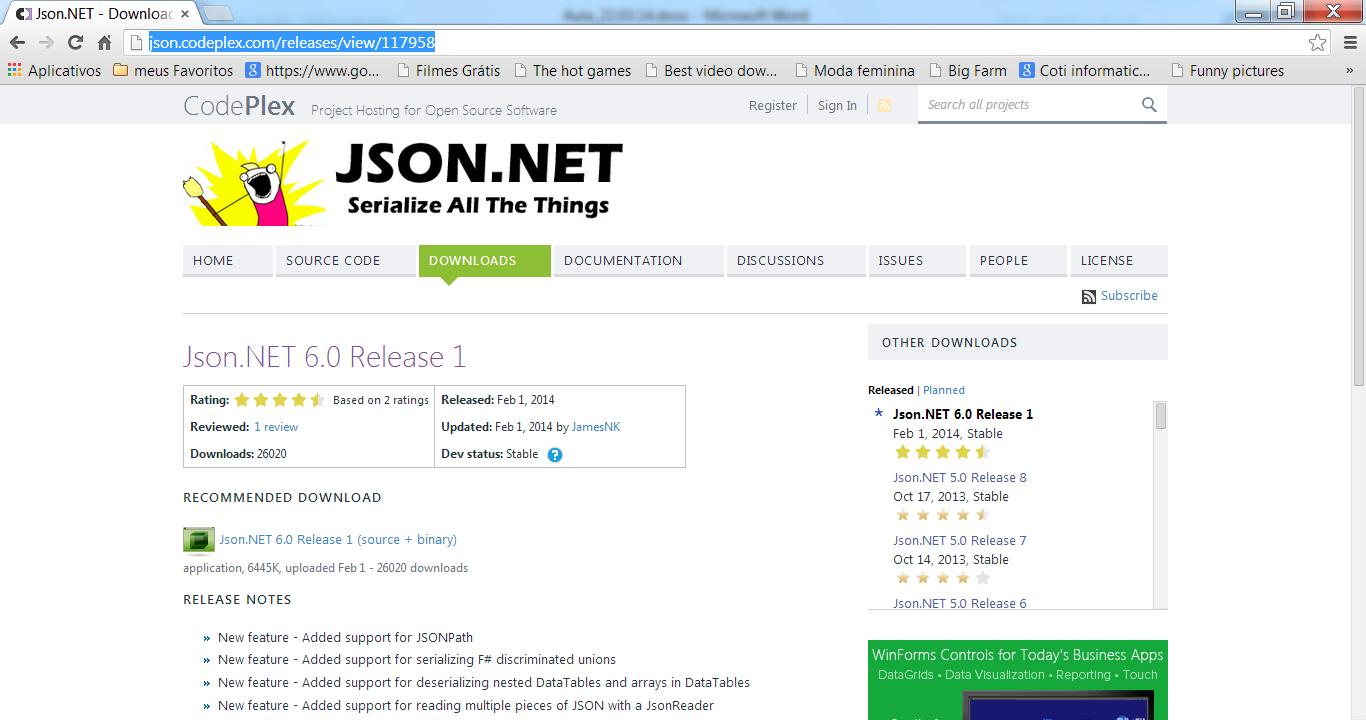
public decimal Preco { get; set; }

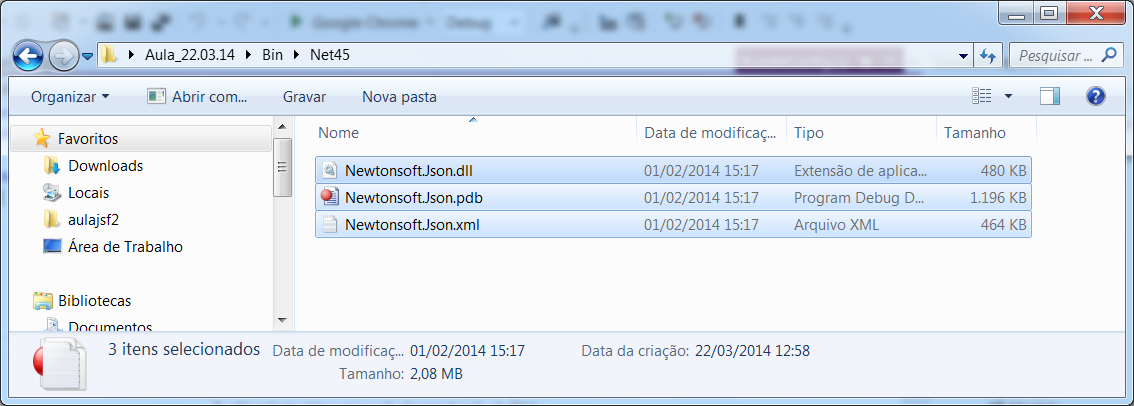
public int Quantidade { get; set; }

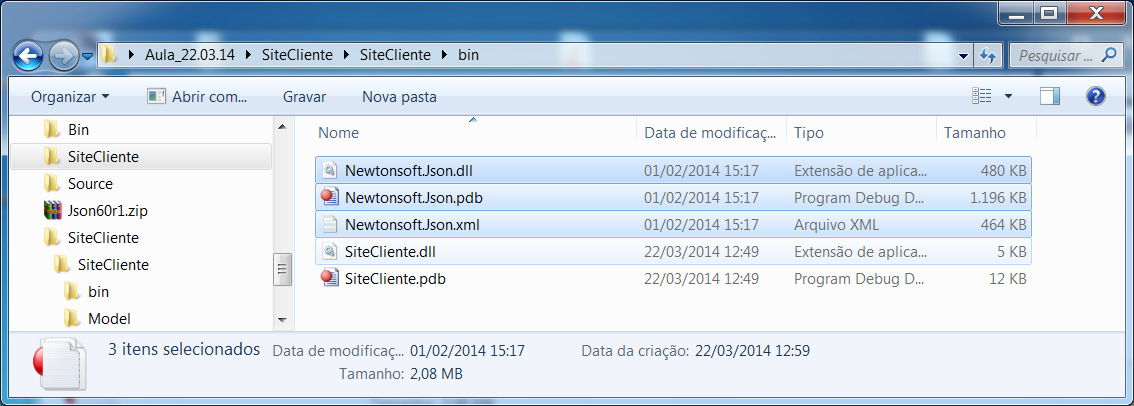
}

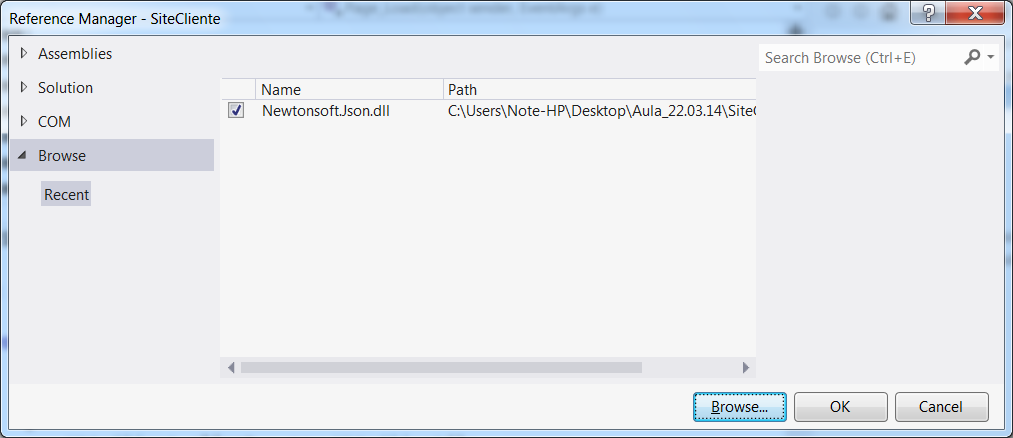
}

<http://json.codeplex.com/releases/view/117958>









using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Net.Http;

using Newtonsoft.Json;

using SiteCliente.Model;

namespace SiteCliente

{

public partial class Pesquisa : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

try

{

HttpClient client = new HttpClient();

client.BaseAddress = new Uri("http://localhost:49200");

//dominio do site

//acessar o endereço que fornece os dados

HttpResponseMessage resp = client.GetAsync

("api/Produto?json=true").Result;

//pegar os dados

string dados = resp.Content.ReadAsStringAsync().Result;

List<Produto> lista = new List<Produto>();

lista = JsonConvert.DeserializeObject<List<Produto>>(dados);

gridProdutos.DataSource = lista;

gridProdutos.DataBind();

}

catch(Exception ex)

{

Response.Write(ex.Message);

}

}

}

}

