CENTRO PAULA SOUZA

Prof. Ms. Eduardo Ribeiro

PROGRAMAÇÃO ORIENTADA A OBJETOS

Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

Apresentação

Prof. Ms. Eduardo Ribeiro

http://lattes.cnpq.br/8608842967266868

Contatos

Carlos.trindade@cps.sp.gov.br

/Duthesanto

profeduardoribeiro

14 9 9148-8887

Formação acadêmica

Graduação em Tecnologia em Processamento de Dados - FAI (2004)
Especialização em Sistema para Internet – UNIVEM (2008)
Licenciatura Plena em Informática – FATEC Marília (2008)
Mestrado em TELEVISÃO DIGITAL: INFORMAÇÃO E CONHECIMENTO – UNESP (2015)
Aperfeiçoamento Ensino e aprendizagem EJA - CEETEPS (2016)
Aperfeiçoamento em Ambiente Virtual de Aprendizagem Moodle – UNESP (2016)
Especialização em andamento em MÍDIAS NA EDUCAÇÃO - UFSJ (2017)
Doutorado em andamento em MÍDIA E TECNOLOGIA. (UNESP Bauru)

Apresentação

Profissional

CNP-M - 2006 - TECNOL. E SERV. DE INFORMÁTICA.

ETEC - 2006 - Centro Estadual de Educação Tecnológica Paula Souza.

2010 – Vice-superintendência do Centro Paula Souza, para atuar no desenvolvimento do projeto Sistema Integrado de Gestão Acadêmica.

Faculdade de Tecnologia de Garça – 2016.

Linguagens de programação

ASP; PHP; Coldfusion; C#; Visual Basic; Java; VB.NET; GeneXus.

Banco de Dados

IBM - DB2; MySQL; SQL SERVER.

Apresentação

Áreas de Pesquisa

- Inovação tecnológica para televisão digital
- Gestão do Conhecimento
- Gestão de Tecnologia da Informação
- Tecnologia na Educação
- Educação
- Mídias Digitais

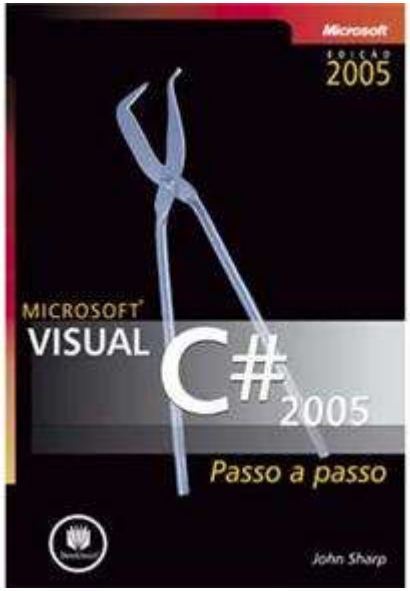
PROGRAMAÇÃO ORIENTADA A OBJETOS

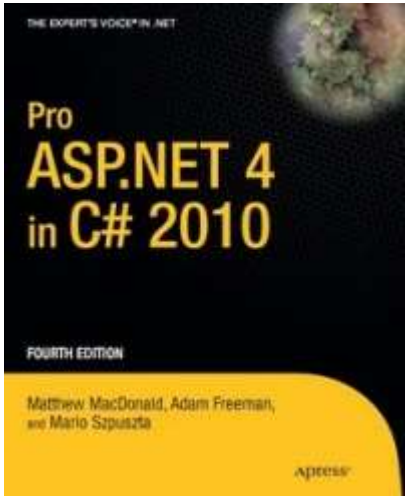
Ementa

Conceitos e evolução da tecnologia de orientação a objetos. Limitações e diferenças entre o paradigma da programação estruturada em relação à orientação a objetos. Conceito de objeto, classe, métodos, atributos, herança, polimorfismo, agregação, associação, dependência, encapsulamento, mensagem e suas respectivas notações na linguagem padrão de representação da orientação a objetos. Implementação de algoritmos orientado a objetos utilizando linguagens de programação. Aplicação e uso das estruturas fundamentais da orientação a objetos.

PROGRAMAÇÃO ORIENTADA A OBJETOS

Bibliografia:





PROGRAMAÇÃO ORIENTADA A OBJETOS

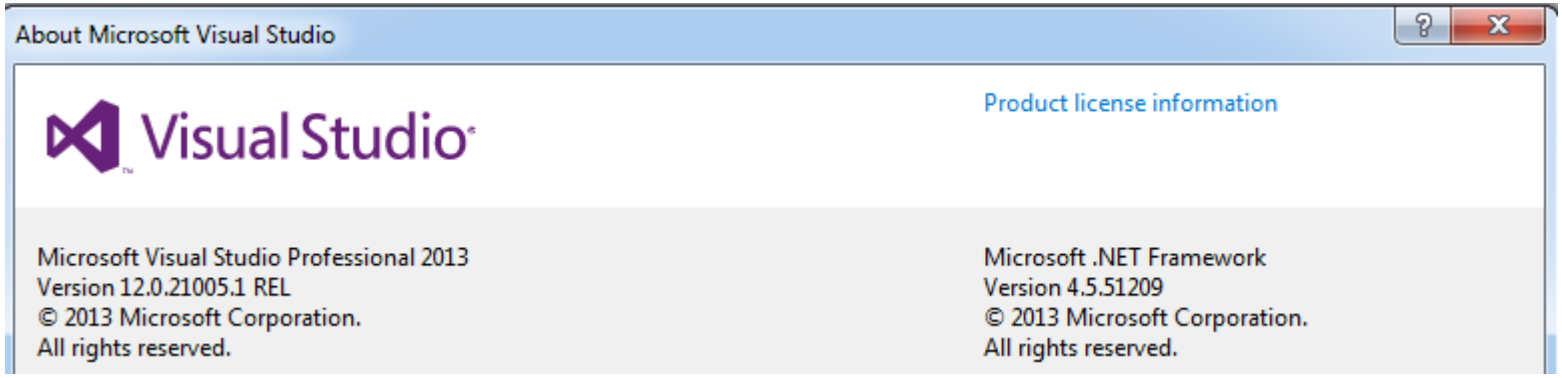
Avaliação

- Leitura de textos recomendados
- Resumos/trabalhos semanais
- Trabalhos em grupo e individuais

- (Atividades 1 + N1 = 10) + (Avaliação Qualitativa = N)
  - Atividades = 6,0
  - N1 = 4,0 *(Apresentação POO)*

- (N2 = 10) *(Entrega – 4p e Apresentação de Projeto WEB – 4p)*
  - Insert (cadastrar)
  - Delete (apagar)
  - Select (Buscar)
  - Update (atualizar)

PROGRAMAÇÃO ORIENTADA A OBJETOS



PROGRAMAÇÃO ORIENTADA A OBJETOS

INTRODUÇÃO A C# (Nivelamento)

Console application c#

Conceitos de Programação Orientada à Objeto

ASP.NET

web application c#

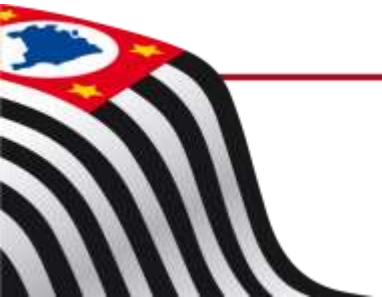
SQL SERVER

SQL Server

Stored Procedure

XML e WEB SERVICE

Capítulo 1: Visão Geral Plataforma .Net



Centro Paula Souza

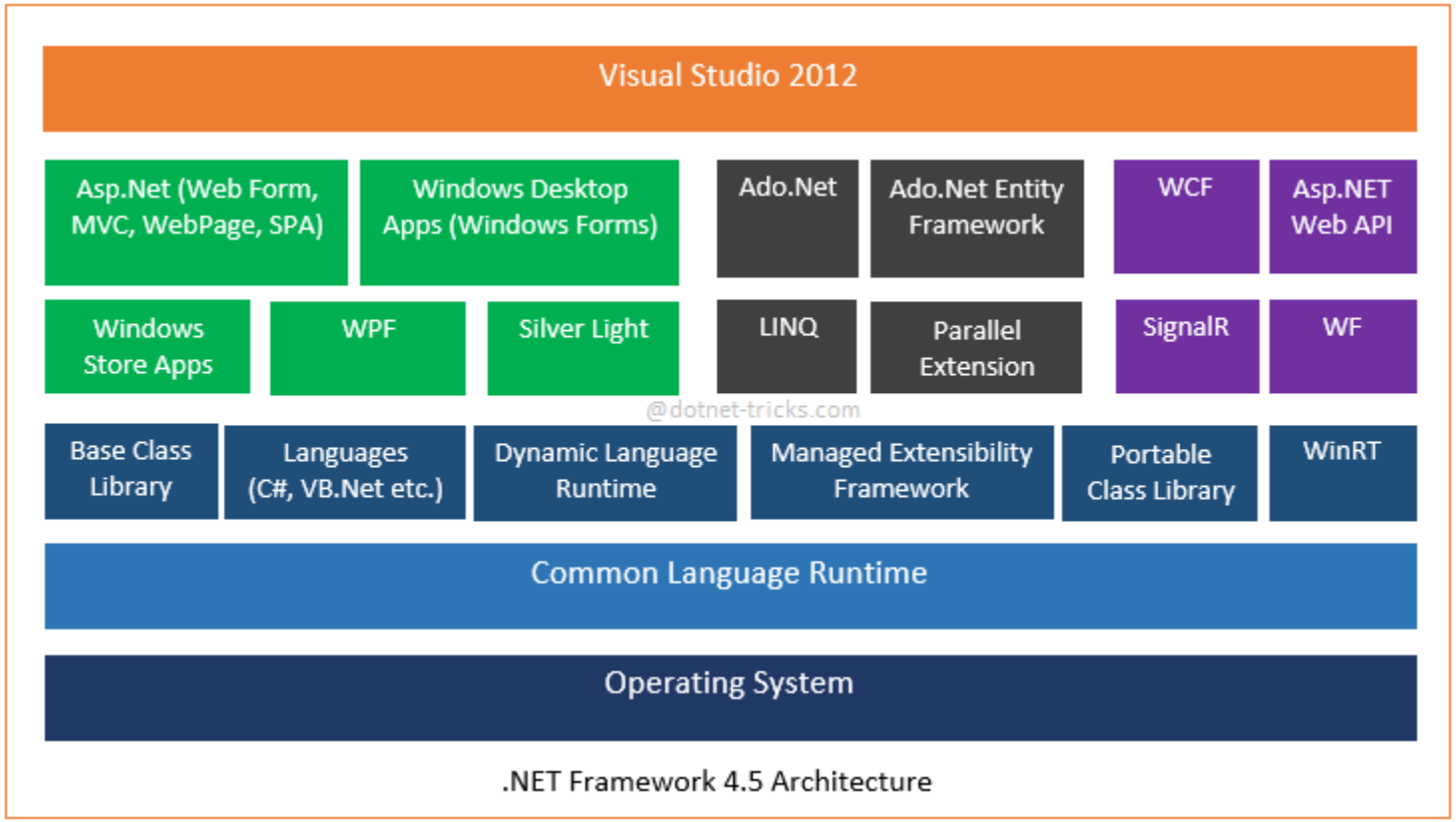
GOVERNO DO ESTADO SÃO PAULO

O que é o .NET Framework?

O .NET Framework é um ambiente de execução gerenciado que fornece uma variedade de serviços para os aplicativos em execução. Ele consiste em dois componentes principais: o **common language runtime** (CLR), que é o mecanismo de execução que gerencia a execução de aplicativos; e a **biblioteca de classes** do .NET Framework, que fornece uma biblioteca de testados, código reutilizável que os desenvolvedores possam chamar de seus próprios aplicativos.

https://msdn.microsoft.com/pt-br/library/hh425099(v=vs.110).aspx

Estrutura do .Net Framework



O que posso fazer com o .NET?

- ✓ O .NET permite desenvolver soluções como:
  1. Aplicativos Web
  2. Aplicativos para Servidores
  3. Aplicativos Movel (*Xamarin*)
  4. Aplicativos de Console
  5. Aplicativos de Banco de Dados
  6. Serviços Windows
  7. Serviços Web e muito mais...

Linguagens .Net

Microsoft Visual Basic.NET
Microsoft C#
Microsoft C++
Microsoft J#
Outras: *www.dotnetpowered.com*

O que é Visual Studio?

- Visual Studio é o ambiente de desenvolvimento integrado da Microsoft que permite criar aplicativos para **Web**, **Windows**, **Mac** e **Linux**. A ferramenta é voltada para desenvolvedores que trabalham com a linguagem de programação C# e com o framework .NET. http://www.techtudo.com.br/tudo-sobre/visual-studio.html

- O produto oferece todos os recursos e soluções para gerir, modelar, desenvolver e testar aplicações. Visual Studio apresenta também controles que visam o aperfeiçoamento da produtividade no código fonte.

Start Page - Microsoft Visual Studio

FILE EDIT VIEW PROJECT DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q)

Carlos Eduardo da Trindade Ribeiro

Attach... Debug

Toolbox

Search Toolbox

General

Start Page

Professional 2013

Start

New Project...

Open Project...

Open from Source Control...

Recent

teste1

Discover what's new in Professional 2013

You can find information about new features and enhancements in Professional 2013 by reviewing the following sections.

Learn about new features in Professional 2013
See what's new in .NET Framework 4.5.1
Explore what's new in Team Foundation Service

Relocate the What's New information

What's new on Microsoft Platforms

Windows
Windows Azure
ASP.NET vNext and Web
Windows Phone
Microsoft Office
SharePoint Development

Announcements

We are unable to download content due to network issues.

Click to retry

Keep page open after project loads

Show page on startup

Output

Show output from: Package Manager

```
True

Type 'get-help EntityFramework' to see all available Entity Framework commands.
```

Team Explorer - Connect

Connect | Offline

Select Team Projects...

Team Foundation Service

Sign Up

Services to help you ship high quality software. On time, every time. Focus on your code. We'll simplify the rest.

Hide

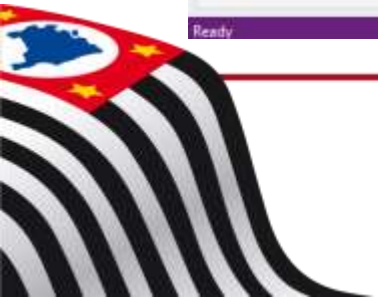
Local Git Repositories

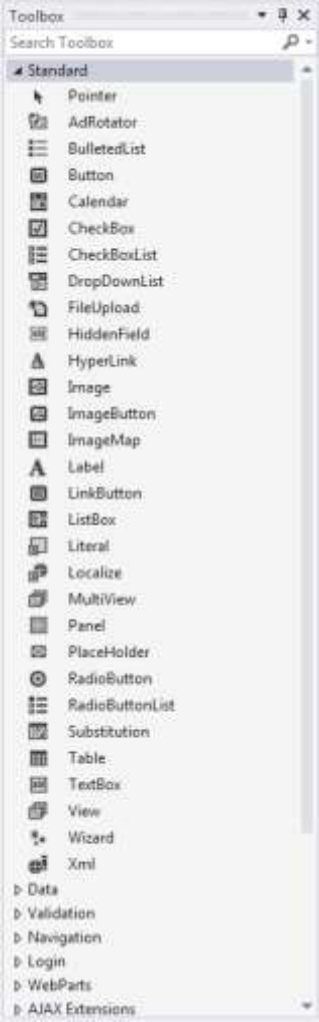
New | Add | Clone | View Options

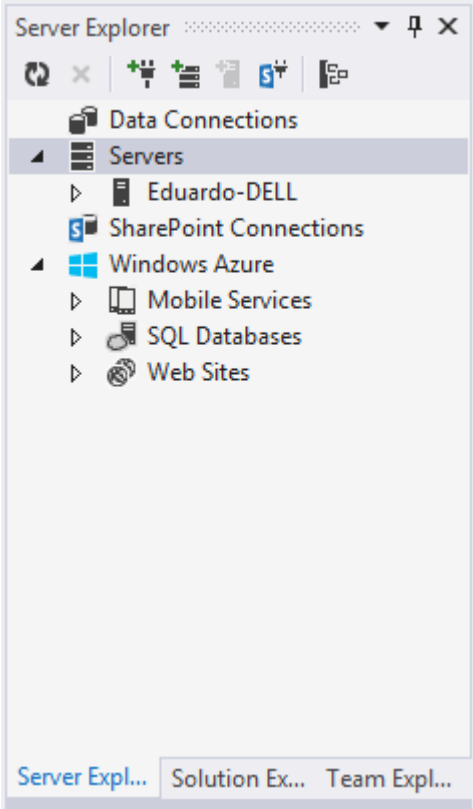
Add or clone a Git repository to get started.

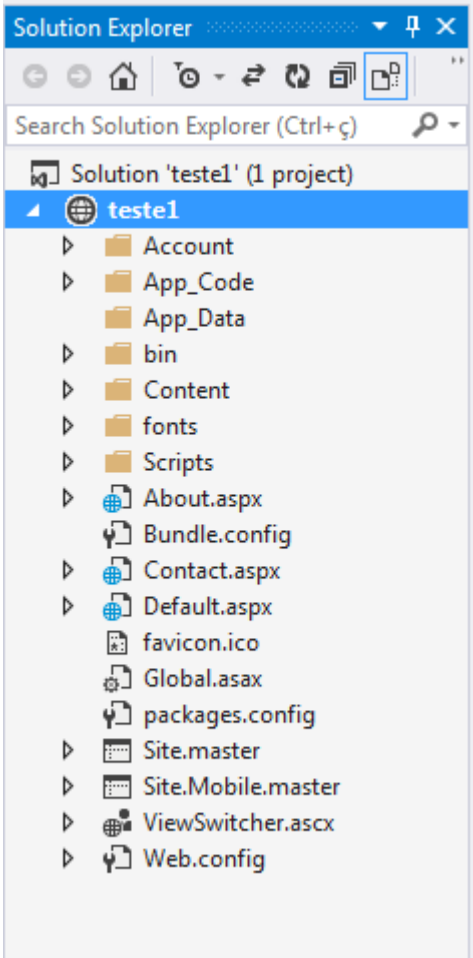
Server Explorer Solution Explorer Team Explorer

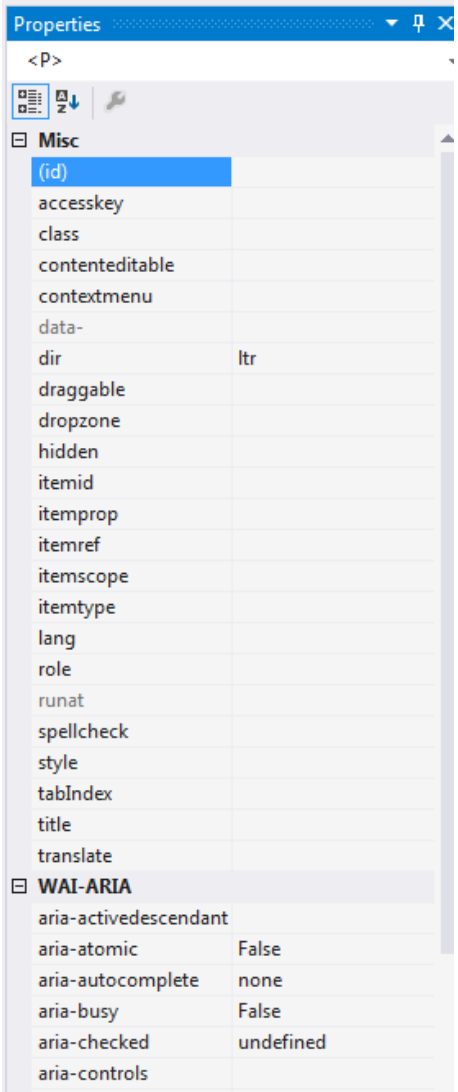
Ready




Toolbox
Search Toolbox
Standard
Pointer
AdRotator
BulletedList
Button
Calendar
CheckBox
CheckBoxList
DropDownList
FileUpload
HiddenField
HyperLink
Image
ImageButton
ImageMap
Label
LinkButton
ListBox
Literal
Localize
MultiView
Panel
PlaceHolder
RadioButton
RadioButtonList
Substitution
Table
TextBox
View
Wizard
Xml
Data
Validation
Navigation
Login
WebParts
AJAX Extensions


Server Explorer
Data Connections
Servers
Eduardo-DELL
SharePoint Connections
Windows Azure
Mobile Services
SQL Databases
Web Sites
Server Expl...
Solution Ex...
Team Expl...


Solution Explorer
Search Solution Explorer (Ctrl+ç)
Solution 'teste1' (1 project)
teste1
Account
App_Code
App_Data
bin
Content
fonts
Scripts
About.aspx
Bundle.config
Contact.aspx
Default.aspx
favicon.ico
Global.asax
packages.config
Site.master
Site.Mobile.master
ViewSwitcher.ascx
Web.config


Properties
<P>
Misc
(id)
accesskey
class
contenteditable
contextmenu
data-
dir
ltr
draggable
dropzone
hidden
itemid
itemprop
itemref
itemscope
itemtype
lang
role
runat
spellcheck
style
tabIndex
title
translate
WAI-ARIA
aria-activedescendant
aria-atomic
False
aria-autocomplete
none
aria-busy
False
aria-checked
undefined
aria-controls

Start Page

Professional 2013

Start

New Project...

Open Project...

Open from Source Control...

Recent

teste1

Keep page open after project loads

Show page on startup

Discover what's new in Professional 2013

You can find information about new features and enhancements in Professional 2013 by reviewing the following sections.

Learn about new features in Professional 2013
See what's new in .NET Framework 4.5.1
Explore what's new in Team Foundation Service

Relocate the What's New information

What's new on Microsoft Platforms

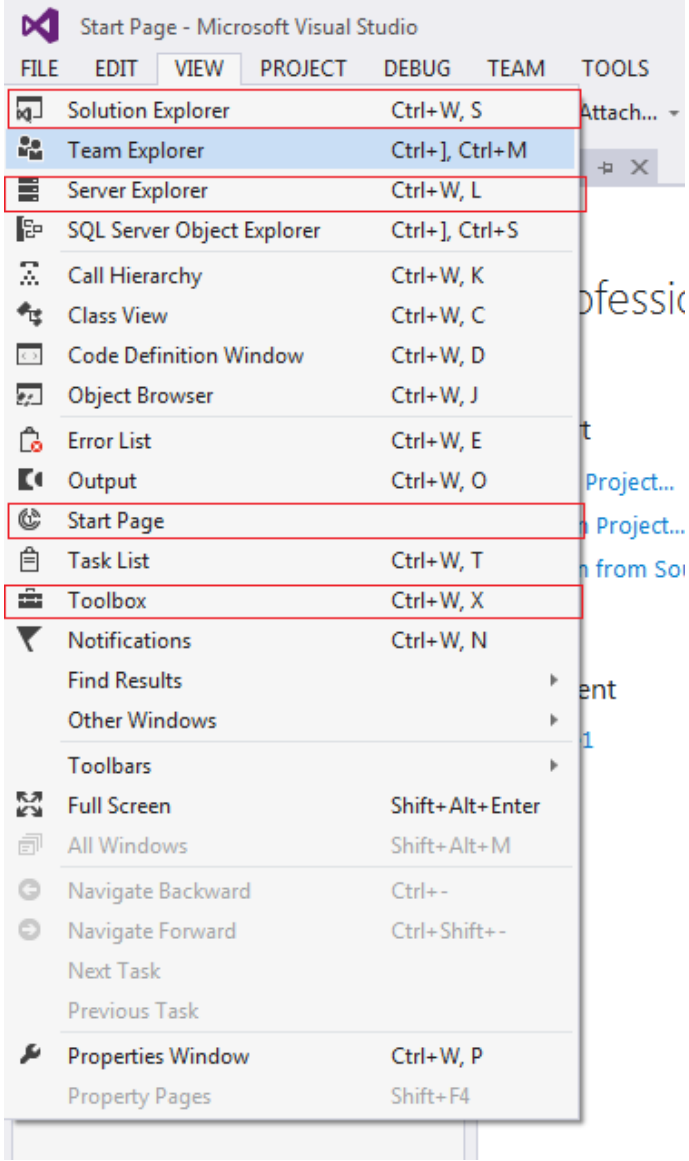
Windows
Windows Azure
ASP.NET vNext and Web
Windows Phone
Microsoft Office
SharePoint Development

Announcements

We are unable to download content due to network issues.

Click to retry

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO



Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

INTRODUÇÃO AO C# (CSHARP)

C# (CSharp) é uma linguagem de programação orientada a objetos criada pela Microsoft, faz parte da sua plataforma .Net. A companhia baseou C# na linguagem C++ e Java.

Tudo é Objeto

Em C#, ao contrário de linguagens como Java ou C++, tipos de dados e objetos interagem. C# fornece um “sistema unificado de tipos”, onde todos os tipos são tratados como objetos, sem perda de performance.

História Da Linguagem

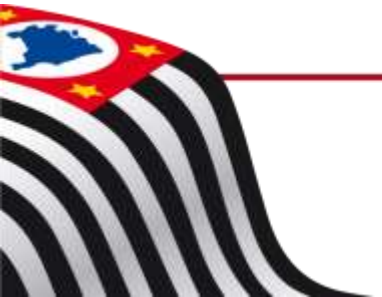
A linguagem C# foi criada junto com a arquitetura .NET. Embora existam várias outras linguagens que suportam essa tecnologia (como VB.NET, C++, J#), C# é considerada a linguagem símbolo do .NET pelas seguintes razões:

Foi criada praticamente do zero para funcionar na nova plataforma, sem preocupações de compatibilidade com código de legado.

O compilador C# foi o primeiro a ser desenvolvido e a maior parte das classes do .NET Framework foram desenvolvidas em C#.

A criação da linguagem, embora tenha sido feita por vários desenvolvedores, é atribuída principalmente a **Anders Hejlsberg**, entre suas criações mais conhecidas estão o **Turbo Pascal** e o **Delphi**.

Capítulo 2: Introdução a C#



Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

Qual a estrutura de um programa em C#?

- Execução do programa começa Main( )
- A palavra-chave using refere a recursos da Biblioteca de Classes do.NET Framework
- Instruções são comandos que executam ações
  - Um programa é feito de muitas instruções
  - Instruções são divididas por ponto-e-vírgula
  - Chaves são usadas para agrupar instruções

```
using System;
class MundoPacifico {
    static void Main() {
        Console.WriteLine (“Paz, Mundo!");
    }
}
```

Qual a estrutura de um programa em C#?

Dica:

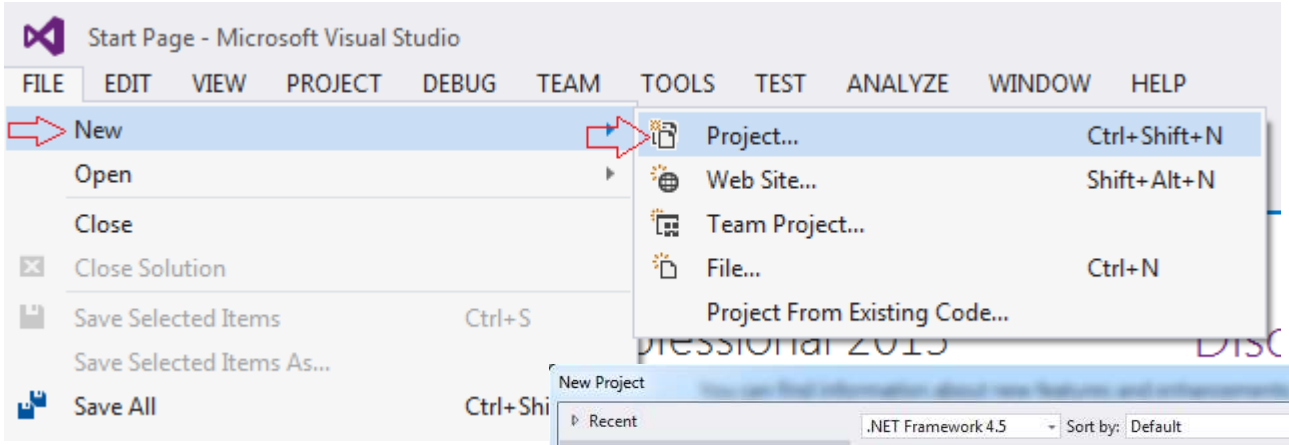
Você também poderia acessar ao método WriteLine da classe Console da seguinte maneira:

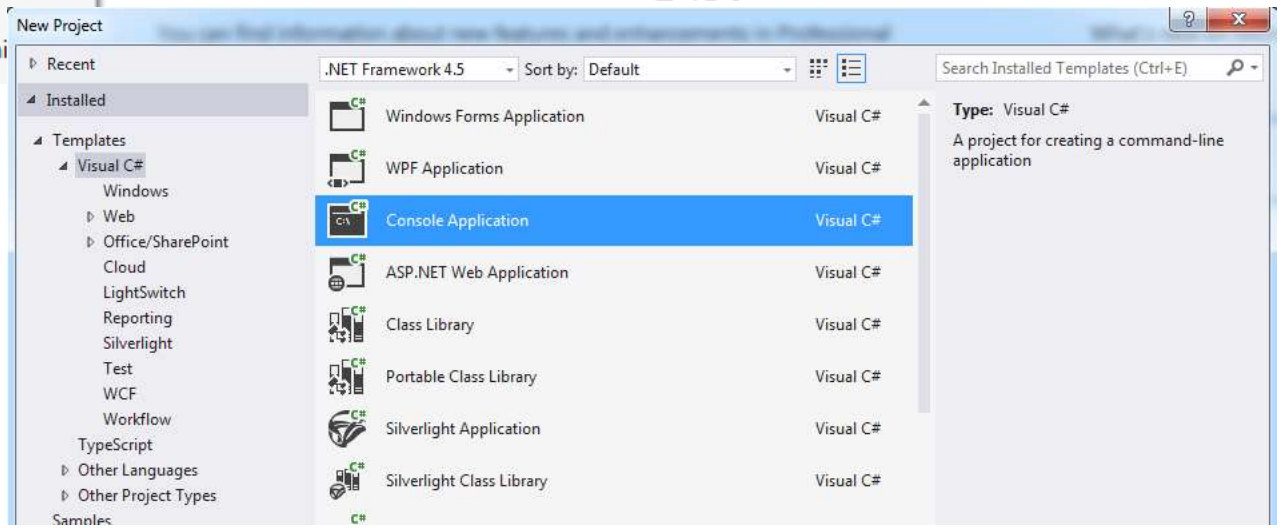
System.Console.WriteLine("Paz, Mundo!");

Mas utilizando a diretiva **using** você pode estar livre de incômodos ocasionados por erros de escritas, esquecimentos de declarações, etc...

As classes pertencentes ao namespace System são acessadas em tempo de execução!

1º Exemplo “Hello World”





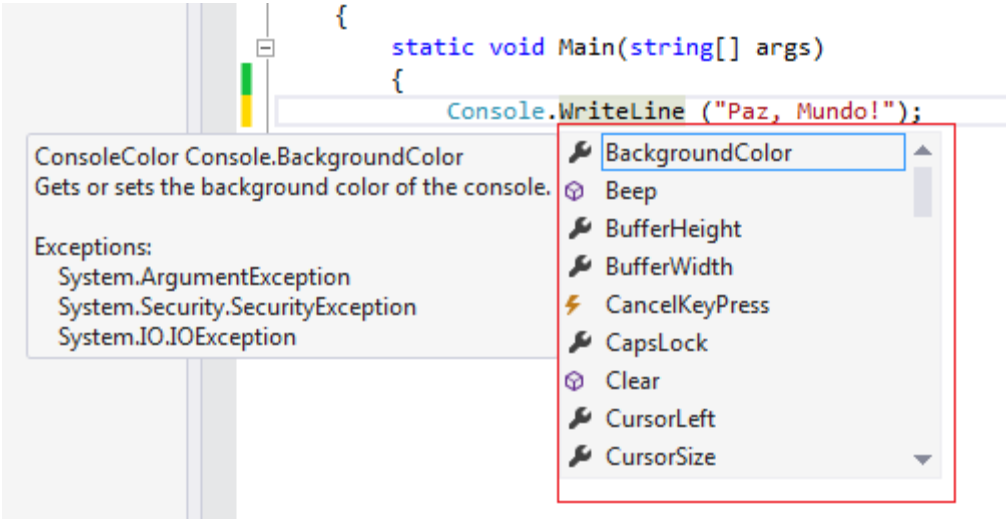
Centro Paula Souza

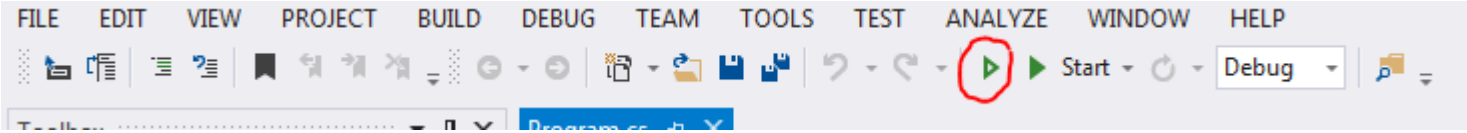
GOVERNO DO ESTADO SÃO PAULO

1º Exemplo “Hello World”

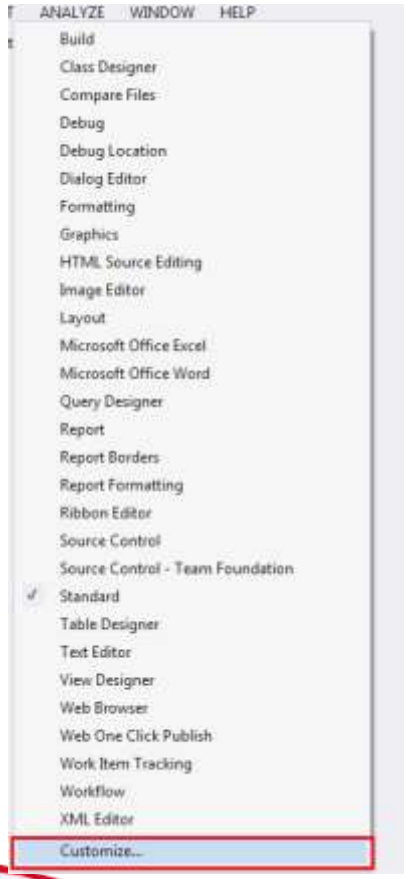
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CA_HelloWorld
{
    class MundoPacifico
    {
        public static void Main(string[] args)
        {
            Console.WriteLine ("Paz, Mundo!");
        }
    }
}
```

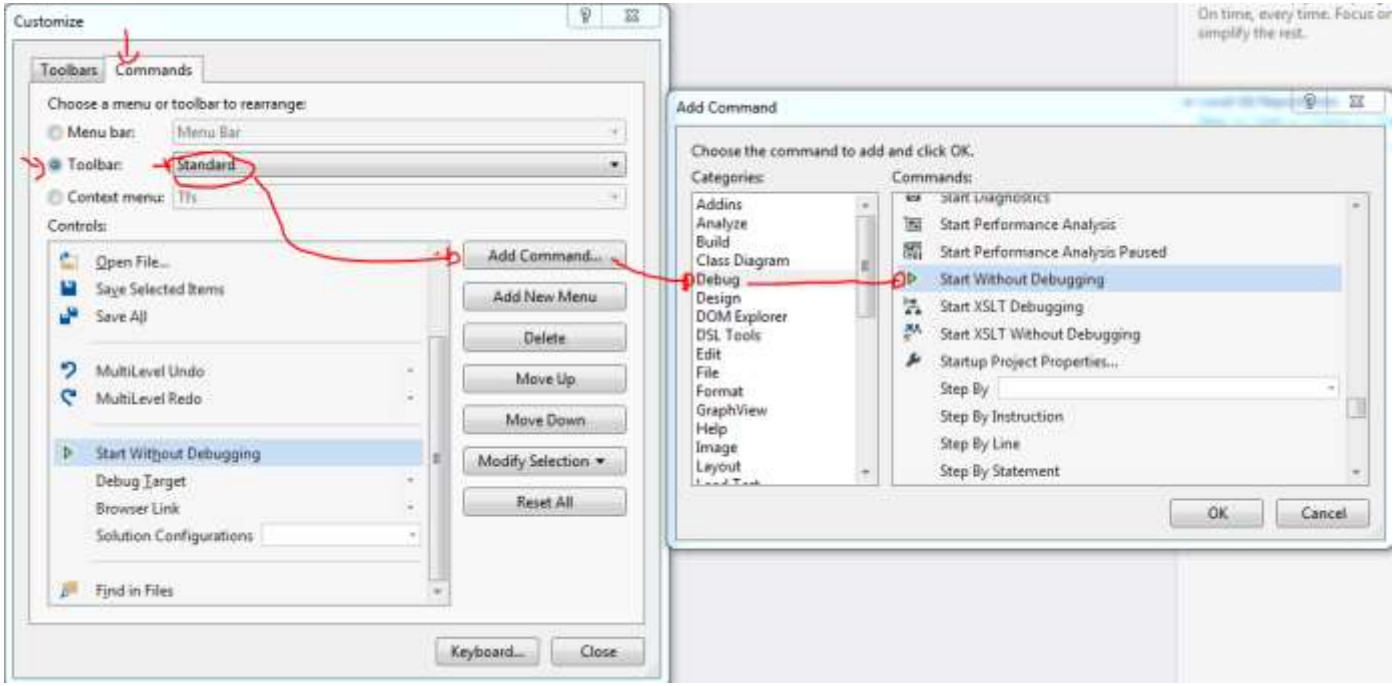
Ao digitar **Console.** será mostrado o IntelliSense que mostra todos os itens relacionados a classe ou parâmetros e ajuda a evitar erros de digitação.





Clique com botão direito na barra “Standard”. Start Without Debugging (Ctrl+F5)





Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

1º Exemplo “Hello World”

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

namespace CA_HelloWorld
{
    class MundoPacifico
    {
        public static void Main(string[] args)
        {
            Console.WriteLine ("Paz, Mundo!");
        }
    }
}
```

```
// Console.ReadLine();
// Console.ReadKey();
// System.Threading.Thread.Sleep(3000);
```

BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP

Start Debug

Program.cs

CA_HelloWorld.MundoPacifico

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

C:\Windows\system32\cmd.exe

```
Paz, Mundo!
Pressione qualquer tecla para continuar. . .
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

public static void Main()

Este é o ponto de entrada do programa quando executado a partir da console. Este método também pode ser chamado de outro código utilizando-se a sintaxe **Hello.Main()**. A definição **public static void** indica que o método **Main** é público (**public**), que pode ser acessado diretamente através da classe (**static**) e que não retorna nenhum valor (**void**).

Classe estática não pode ser instanciada, ou seja, não pode criar uma variável do tipo da classe.

Como formatar código no C#

Usar identação para indicar instruções relacionadas

C# é **case sensitive**.

Espaços em branco são ignorados

Comentários de uma linha através de //

Coloque comentários de linhas compostas entre /* e */

Variáveis e Tipos de Dados

Nomeando Variável

É recomendado na nomeação de variável:

- Evite usar underline;
- Não crie variáveis que se diferenciem apenas pela sua forma. Exemplo: minhaVariavel e outra chamada MinhaVariavel;
- Procure iniciar o nome com uma letra minúscula;
- Evite usar todas as letras maiúsculas;
- Quando o nome tiver mais que uma palavra, a primeira letra de cada palavra após a primeira deve ser maiúscula (conhecido como notação camelCase);
- Não use notação Húngara. Exemplo: msg (*Message)*

Declarando Variáveis

O C# pode armazenar diferentes tipos de dados: como inteiros, números de ponto flutuante, textos e caracteres. Assim que declaramos uma variável precisamos identificar que tipo de dado ela armazenará.

int contador;

Esse exemplo declara uma variável chamada contador do tipo *integer*. Ou seja ela deverá armazenar números inteiros.

Atribuindo Valor a Variáveis

Depois de declarar sua variável você precisa atribuir um valor a ela. No C# você não pode usar uma variável antes de colocar um valor nela, isso gera um erro de compilação.

int numeroFuncionario;
numeroFuncionario = 23;
Ou
int numeroFuncionario = 23;

Primeiro nos declaramos nossa variável do tipo integer. Depois atribuímos o valor 23 a ela. Entendemos pelo sinal de igual como recebe. Assim numeroFuncionario recebe 23.

Tipos de Dados

Nome abreviado	Classe do .NET	Type (Tipo)	Width	Intervalo (bits)
byte	Byte	Inteiro sem sinal	8	0 a 255
sbyte	SByte	inteiro com sinal com sinal	8	-128 a 127
int	Int32	inteiro com sinal com sinal	32	-2,147,483,648 to 2,147,483,647
uint	UInt32	Inteiro sem sinal	32	0 a 4294967295
short	Int16	inteiro com sinal com sinal	16	-32.768 a 32.767
ushort	UInt16	Inteiro sem sinal	16	0 a 65535
long	Int64	inteiro com sinal com sinal	64	-922337203685477508 to 922337203685477507
ulong	UInt64	Inteiro sem sinal	64	0 a 18446744073709551615
float	Single	Tipo de ponto flutuante de precisão simples	32	-3.402823e38 para 3.402823e38
double	Double	Tipo de ponto flutuante de precisão dupla	64	-1.79769313486232e308 para 1.79769313486232e308
char	Char	Um único caractere Unicode	16	Unicode símbolos usados no texto
bool	Boolean	Tipo booliano lógico	8	True ou false
object	Object	tipo de base de todos os outros tipos		
string	String	Uma sequência de caracteres		
decimal	Decimal	Preciso tipo fracionário ou integral que pode representar números Decimal com 29 dígitos significativos	128	±1.0 × 10e−28 para ±7.9 × 10e28

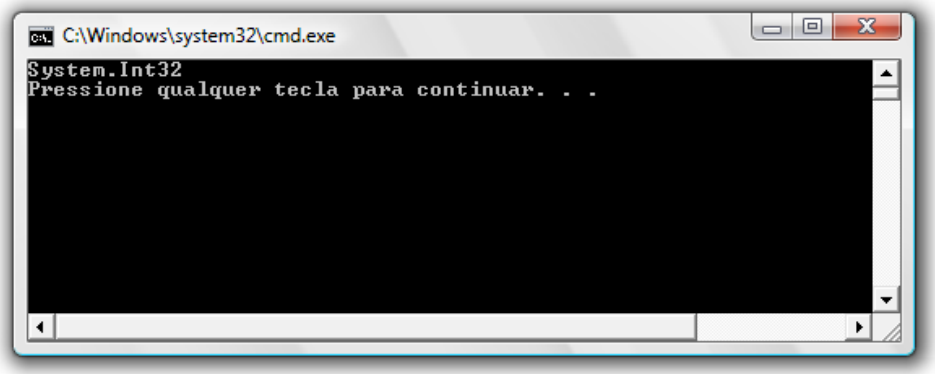
https://msdn.microsoft.com/pt-br/library/ms228360(v=vs.90).aspx

Dica: Para retornar o tipo de qualquer variável do C# você pode usar o método GetType(); Como no exemplo:

Console.WriteLine(minhaVariavel.GetType());

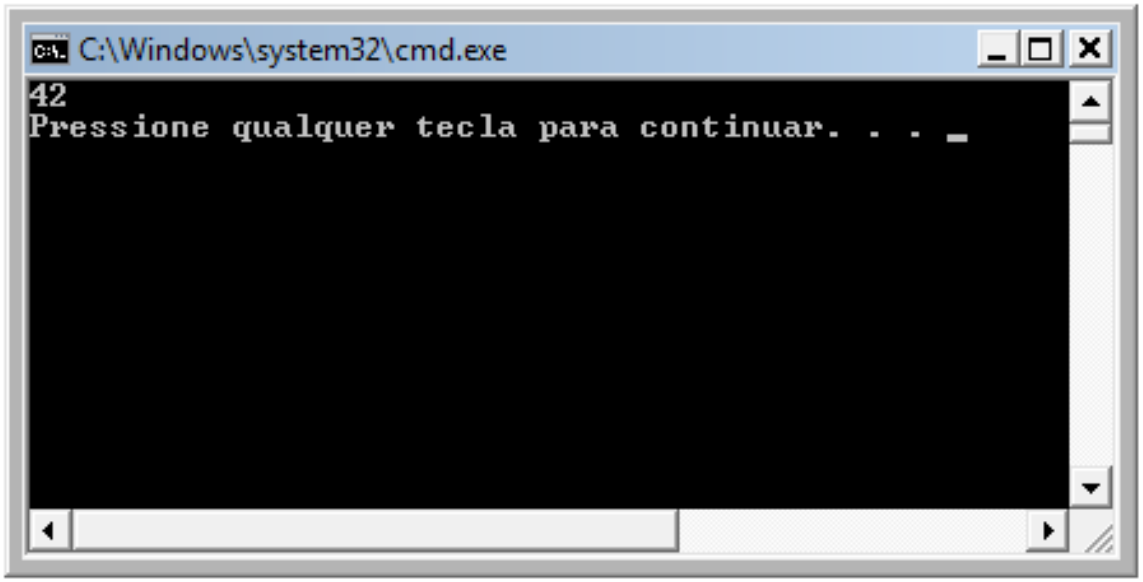
Isso retornaria o tipo da variável *minhaVariavel*

```
namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {
            int vContador;
            vContador = 10;
            Console.WriteLine(vContador.GetType());
        }
    }
}
```



```
namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {
            int contador;
            contador = 2;
            contador = contador + 40;
            Console.WriteLine(contador);
        }
    }
}
```

O código seguinte declara uma variável do tipo integer chamada contador e armazena o valor 2 nesta variável, depois incrementa o valor 40:



Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Dica: Perceba que para somar o valor a variável precisamos repetir o nome da variável.

Podemos fazer da seguinte forma também em C#:

```
namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {
            int contador;
            contador = 2;
            contador+=40;
            Console.WriteLine(contador);
        }
    }
}
```

Palavras-chave

- Estas palavras são chamadas de palavras reservadas ou Palavras-chave e cada uma tem um uso particular.

- Palavras-chave também não são permitidas como nome de variáveis.

https://msdn.microsoft.com/pt-br/library/x53a06bb.aspx

abstract	as	base	bool
break	byte	case	catch
char	checked	class	const
continue	decimal	default	delegate
do	double	else	enum
event	explicit	extern	false
finally	fixed	float	for
foreach	goto	if	implicit
in	in (generic modifier)	int	interface
internal	is	lock	long
namespace	new	null	object
operator	out	out (generic modifier)	override
params	private	protected	public
readonly	ref	return	sbyte
sealed	short	sizeof	stackalloc
static	string	struct	switch
this	throw	true	try
typeof	uint	ulong	unchecked
unsafe	ushort	using	virtual
void	volatile	while	

Exercício

- Crie um programa em Console Application

que imprima na tela os seguintes dados:

Nome

Sobrenome

Endereço

Cidade

CEP

```
namespace ConsoleApplication1
{
    class Program
    {
        public static void Main(string[] args)
        {
            string nomeUsuario, sobrenome, endereco, cidade, cep;
            nomeUsuario = "Eduardo";
            sobrenome = "Ribeiro";
            endereco = "Rua Aimores";
            cidade = "Tupã";
            cep = "17600-000";
            System.Console.WriteLine(nomeUsuario + " " + sobrenome);
            System.Console.WriteLine(endereco);
            System.Console.WriteLine(cidade);
            System.Console.WriteLine(cep);
        }
    }
}
```

Capítulo 3: Introdução a C#

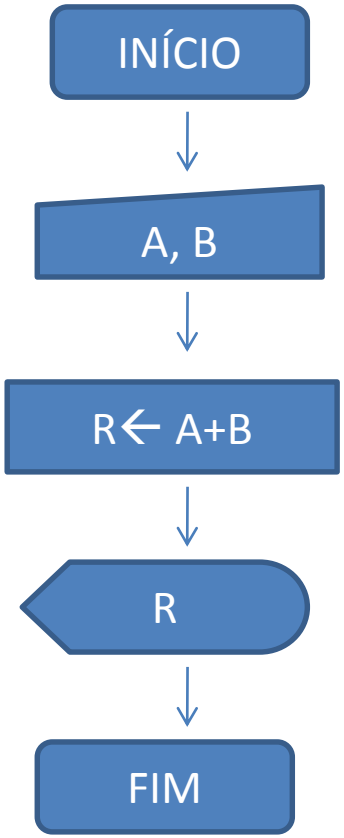
Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Atribuindo Valor a Variáveis

```
namespace ConsoleApplication1
{
    class Program
    {
        public static void Main(string[] args)
        {
            string nome;
            Console.WriteLine("Informe Seu Primeiro Nome");
            nome = System.Console.ReadLine();
            System.Console.WriteLine("Informe sua Idade");
            int idade = int.Parse(Console.ReadLine());
            Console.WriteLine("Nome Informado foi: " + nome);
            Console.WriteLine("Idade Informada: " + idade);
            System.Console.Read();
                    class System.Console
                    Represents the standard input, output, and error streams for console ap
        }
    }
}
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Diagrama de Bloco



1º Ler um número inteiro;
2º Ler um segundo número inteiro;
3º Efetuar a adição dos dois valores;
4º Apresentar o valor calculado.

```
namespace ConsoleApplication1
{
    class Program
    {
        public static void Main(string[] args)
        {
            int a, b, r;
            System.Console.WriteLine("Informe o Primeiro Número a ser calculado");
            a = int.Parse(Console.ReadLine()); //int.Parse Converte para inteiro
            Console.WriteLine("Informe o Segundo Número a ser calculado");
            b = int.Parse(Console.ReadLine());
            r = a + b;
           Console.WriteLine("Asoma de A+B é: "+ r);
           Console.Read();

        }
    }
}
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Comandos

- ***Comando Isolado***

```
if(a==true) System.Console.Write("Verdadeiro");
```

- ***Bloco de comandos***

```
{
    int x = 100;
    int y = 200;
    int z = x + y;
}
```

O CSharp possui um grupo variado de comandos de controle de fluxo do programa que definirão sua lógica de processamento. Portanto, na programação seqüencial, na programação estruturada ou programação orientada a objetos, cada comando é executado igualmente, ou seja, instrução por instrução.

Estes comandos podem ser encontrados dentro de um bloco ou na forma isolada.

Operadores Aritméticos

+	(Adição)
-	(Subtração)
*	(Multiplicação)
/	(Divisão)
%	(Resto)

Operadores de atribuição

=	(Atribuição simples)
+=	(Atribuição aditiva)
-=	(Atribuição Subtrativa)
*=	(Atribuição Multiplicativa)
/=	(Atribuição de divisão)
%=	(Atribuição de módulo)

Operadores relacionais

==	(Igualdade)
>	(Maior)
<	(Menor)
<=	(Menor igual)
>=	(Maior igual)
!=	(Diferente)

Operadores lógicos

&& (E)

|| (OU)

Precedência de Operador

- int X = 25 – 7 * 2;
- A expressão acima tem duas respostas dependendo do modo que se faça à conta, por exemplo, 25 - 7 = 18, 18 * 2 = 36, **x = 36**. Ou, 7*2 = 14, 25-14 = 11, **x = 11**.

Qual resposta é correta??

R: Neste caso a segunda, ou seja, **x = 11**, pois o operador de multiplicação '*' tem precedência sobre o operador de subtração '-', assim o calculo de multiplicação é feito primeiro.

Mas e se eu quisesse que a subtração fosse feita primeiro?

```
int X = (25 – 7) * 2;
```

Incremento e Decremento

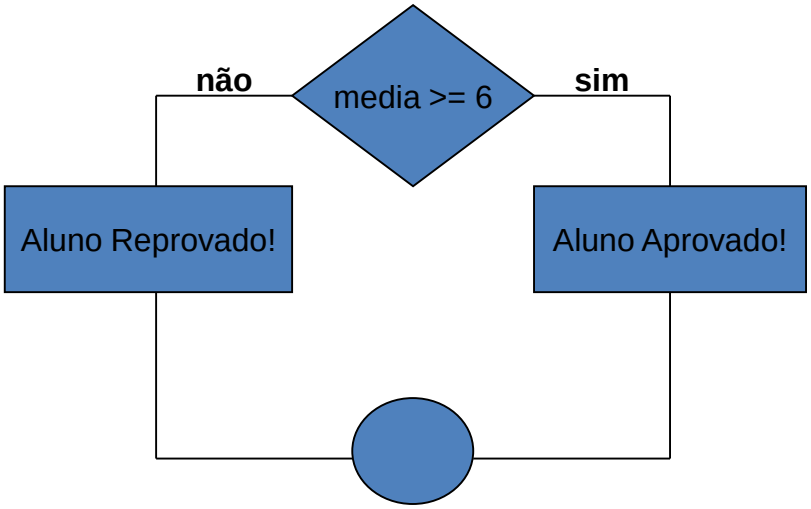
++ (Incremento)

-- (Decremento)

int X = 1; X++; ou int X = 1; X=X+1;

int X = 2; X--; ou int X = 2; X=X-1;

Estrutura de Condição



if e else

```
if (media >= 6)
{
  Console.WriteLine("Aluno Aprovado!");
}
else
{
   Console.WriteLine("Aluno Reprovado!");
}
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Estrutura de Condição If, Else.

```
namespace CA_HelloWorld
{
    class e_ou
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Informe um número");
            int a = int.Parse(Console.ReadLine())%2;
            if (a == 0)
            {
                Console.WriteLine("PAR");
            }
                else
            {
                Console.WriteLine("IMPAR");
            }
            Console.ReadKey();

        }
    }
}
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

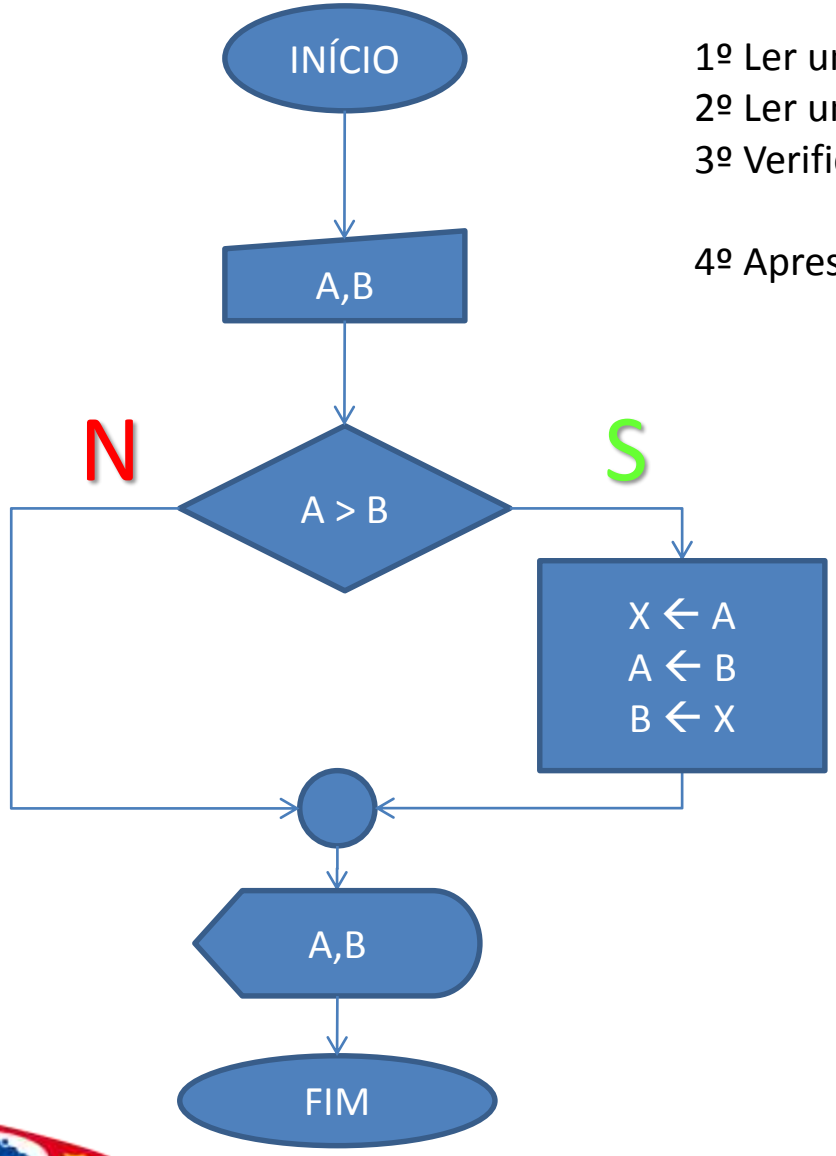
&& (E)

```
namespace CA_HelloWorld
{
    class e_ou
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Informe um número");
            int a = int.Parse(Console.ReadLine());
            Console.WriteLine("Informe outro número");
            int b = int.Parse(Console.ReadLine());
            if ((a + b > 10) && (a + b < 20))
            {
                Console.WriteLine("Certo");
            }
                else
            {
                Console.WriteLine("Errado");
            }
            Console.ReadKey();

        }
    }
}
```

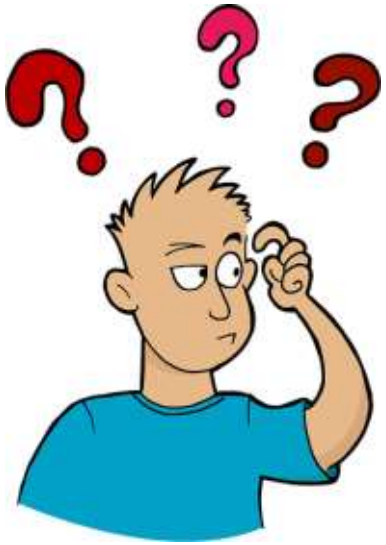
Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

II (ou)

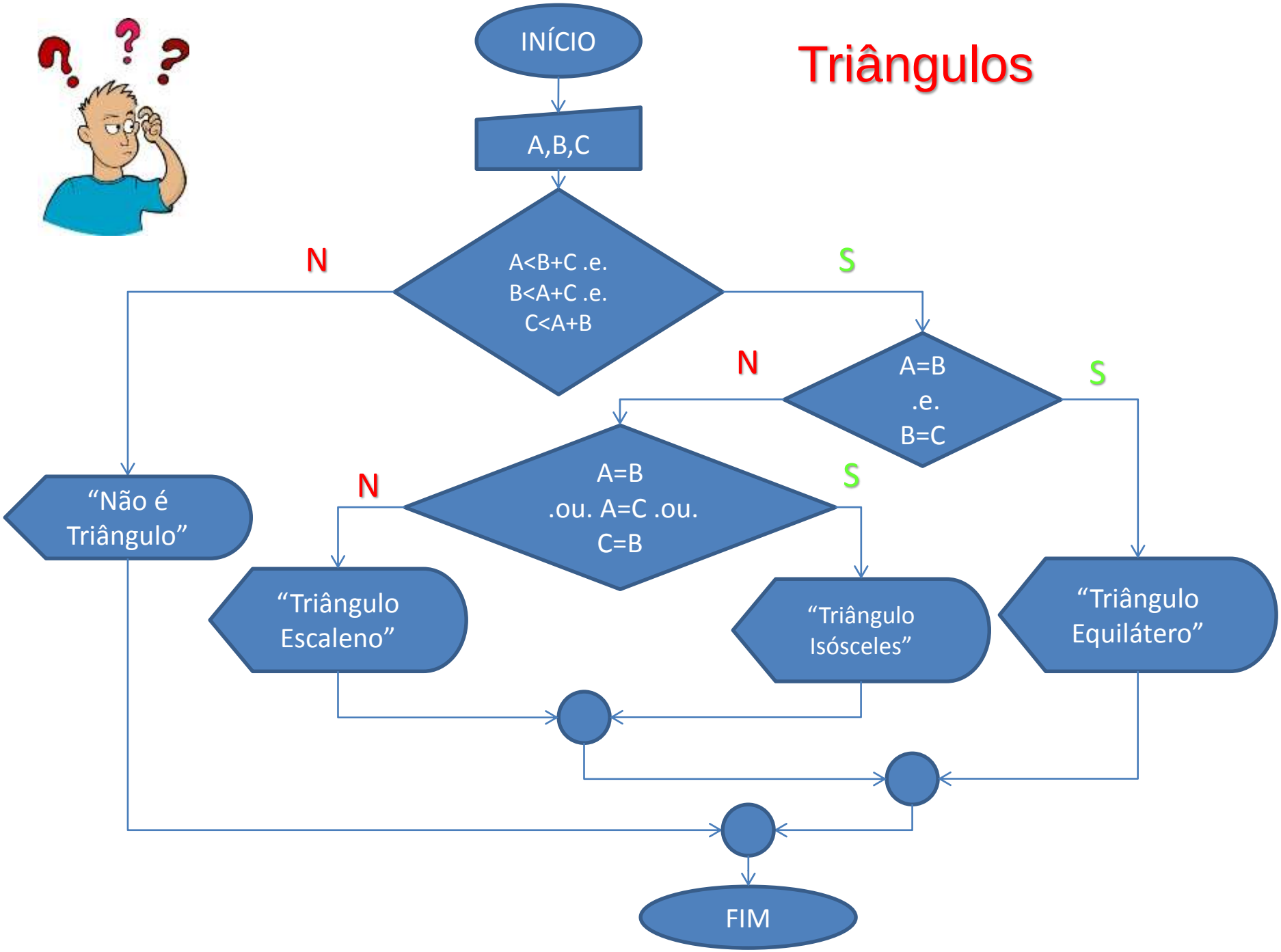
```
namespace CA_HelloWorld
{
    class e_ou
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Informe um número");
            int a = int.Parse(Console.ReadLine());
            Console.WriteLine("Informe outro número");
            int b = int.Parse(Console.ReadLine());
            if ((a + b > 10) || (a + b < 20))
            {
                Console.WriteLine("Certo");
            }
                else
            {
                Console.WriteLine("Errado");
            }
            Console.ReadKey();

        }
    }
}
```



1º Ler um valor para A
2º Ler um valor para B
3º Verificar se o valor de A é maior que valor de B
3.1 Se for verdadeiro, efetuar a troca(*) dos valores
4º Apresentar os valores das variáveis.



Triângulos


INÍCIO
A,B,C
A<B+C .e.
B<A+C .e.
C<A+B
N
S
A=B
.e.
B=C
N
S
A=B
.ou. A=C .ou.
C=B
N
S
"Não é Triângulo"
"Triângulo Escaleno"
"Triângulo Isósceles"
"Triângulo Equilátero"
FIM

```
#region Troca_valores
int a, b, x;

Console.WriteLine("Informe um valor para A!!");
a = int.Parse(Console.ReadLine());
Console.WriteLine("Informe um valor para B");
b = int.Parse(Console.ReadLine());

if (a > b)
{
    x = a;
    a = b;
    b = x;
}
Console.WriteLine("A:" + a);
Console.WriteLine("B:" + b);
Console.ReadKey();


#endregion Troca_valores
```

```
#region Triangulo
int a, b, c;

Console.WriteLine("Informe os lados do Triângulo!!");
Console.WriteLine("Lado:1");
a = int.Parse(Console.ReadLine());
Console.WriteLine("Lado:2");
b = int.Parse(Console.ReadLine());
Console.WriteLine("Lado:3");
c = int.Parse(Console.ReadLine());

if ((a < b + c) && (b < a + c) && (c < a + b))
{
    if ((a == b) && (b == c))
    {
         Console.WriteLine("Equilátero");
         Console.ReadKey();
    }
    else
    {
        if ((a != b) && (a != c) && (b != c))
        {
            Console.WriteLine("Escaleno");
            Console.ReadKey();
        }
        else
        {
            Console.WriteLine("Isósceles");
            Console.ReadKey();
        }
        Isósceles
    }
}
else
{
    Console.WriteLine("Nao é Triângulo");
    Console.ReadKey();
}
#endregion
```

```
#region Isósceles
if ((a == b) && (a != c) || (a == c) && (a != b) || (b == c) && (b != a))
{
    Console.WriteLine("Isósceles");
    Console.ReadKey();
}
#endregion
```

Capítulo 5: Introdução a C#

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Laços de Repetição ou Loop

Conhecidos como laço ou loop, os comandos de iteração executam repetidamente um comando ou bloco de comandos, a partir de uma determinada condição. Em C#, fazem parte dos comandos de iteração: **for, while, do.. while.**

Comando *for*

for (int i =0; i <= 10; i++) { //instruções }

```
class Declares
{
    public static void Main()
    {
        for (int x = 0; x <= 100; x++)
        {
            Console.WriteLine(x);
        }
    }

}
```

O comando *for* possui 3 declarações opcionais, separadas por ponto e vírgula (;), dentro dos parênteses: inicialização, condição e a iteração.

Comando While

while (expressão booleana) { //instruções }

```
class Declares
{
    public static void Main()
    {
        int x = 1;
        while (x <= 100)
        {
            Console.WriteLine(x);
            x++;
        }
    }

}
```

While é mais simples de ser entendido, pois sua syntax não requer que você coloque na mesma linha variável de inicialização, condição e o seu incremento. No loop 'while' apenas colocamos a condição que queremos testar

Comandos *do* e *while*

- Porque é que teríamos mais um loop do tipo 'while'? Se fossemos analisar com cuidado veríamos que o loop 'while' dependendo do caso pode nunca ser executado, ou seja, se a condição do loop 'while' retorna falsa de primeira ele nunca vai ser executado. No exemplo acima se atribuíssemos o valor 10 a variável contador em sua declaração o loop 'while' nunca começaria. Com o loop '**do.. while**' **o código será executado ao menos uma vez, pois a verificação da condição será feita apenas no final da instrução.**

Comandos *do* e *while*

do

{ //instruções }

while (expressão booleana)

```
class Declares
{
    public static void Main()
    {
        int x = 100;
        do
        {
            Console.WriteLine(x);
            x++;
        }
        while (x <= 100);
    }

}
```

Capítulo 6: Introdução a C#

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Função

- Vários programas são forçados a utilizar um mesmo trecho de código várias vezes, devido a funcionalidade que este representa.

- A repetição de código é desaconselhável na programação, mas então como fazer essa funcionalidade poder ser usada em qualquer parte do código?

- Função:
  - Para isto, criamos uma espécie de isolamento desta funcionalidade, este isolamento é chamado de função.

  - Características:
    - Funções englobam certas funcionalidades;
    - Funções podem ou não retornar valores;
    - Funções podem ou não receber informações.

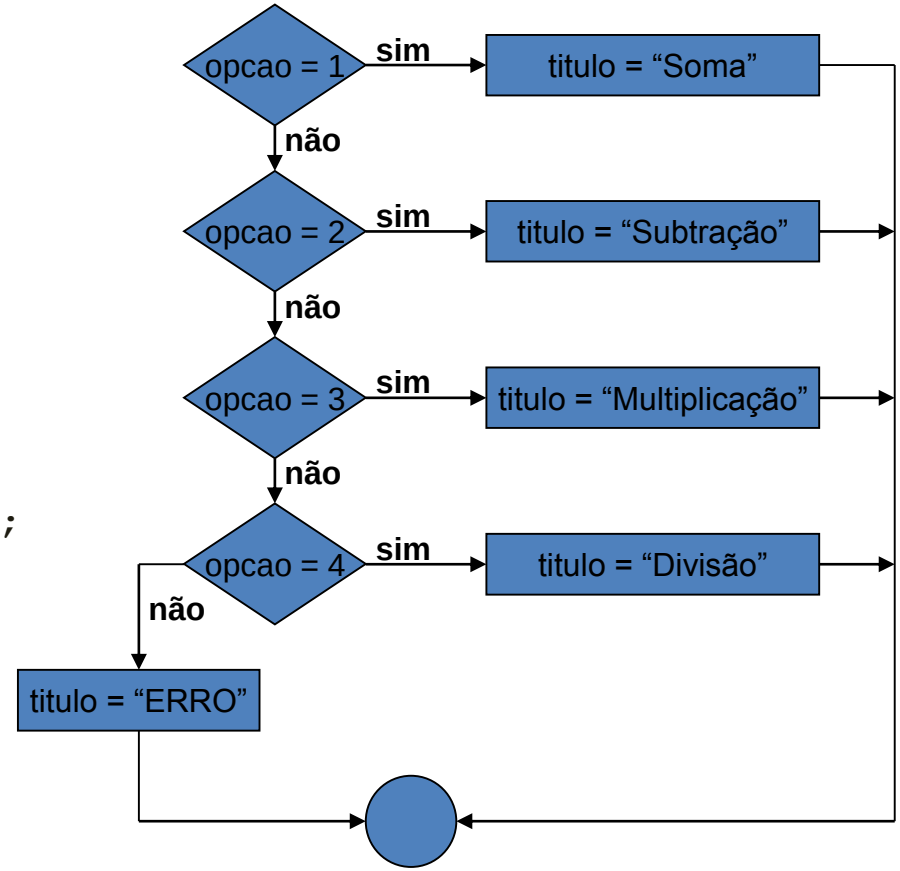
```
using System;

class Declares
{
    private static int a; //Variáveis de escopo de classe

    public static void Main()
    {
        mensagem();
        a = int.Parse(Console.ReadLine());
        escrever();
    }

    //Funções
    private static void mensagem()
    {
        Console.WriteLine("informe um número");
    }
    private static void escrever()
    {
        Console.WriteLine(a+4);
    }

}
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Comandos de Decisão

- switch case

```
string titulo;
switch (opcao)
 {
  case 1:
    titulo = "Soma";
    break;
  case 2:
    titulo = "Subtração";
    break;
  case 3:
    titulo = "Multiplicação";
    break;
  case 4:
    titulo = "Divisão";
    break;
  default:
    titulo = "ERRO";
    break;
 }
```



Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

```
int n;
do
{
    Console.Clear();

    Console.WriteLine("Informe um Nº 1 a 5, 6 = Sair, 7 Limpar Interface!!");
    n = int.Parse(Console.ReadLine());

    switch (n)
    {
        case 1:
            Console.WriteLine(" -- 1 -- ");
            mensagem();
            Console.ReadKey();
            break;
        case 2:
            Console.WriteLine(" -- 3 -- ");
            mensagem();
            Console.ReadKey();
            break;
        case 3:
            Console.WriteLine(" -- 3 -- ");
            mensagem();
            Console.ReadKey();
            break;
        case 4:
            Console.WriteLine(" -- 4 -- ");
            mensagem();
            Console.ReadKey();
            break;
```

```
            case 5:
                Console.WriteLine(" -- 5 -- ");
                mensagem();
                Console.ReadKey();
                break;
            case 7:
                Console.Clear();
                Console.ReadKey();
                break;
            default:
                if (n != 6)
                {
                Console.WriteLine(" Opção não Encontrada!! ");
                mensagem();
                Console.ReadKey();
                }
                break;


        }
    }  while (n != 6);
}

private static void mensagem()
{
   Console.WriteLine("Enter para Continuar");
}
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

```
case 4:
    int x, y;
    Console.WriteLine("Informe nº X");
    x = int.Parse(Console.ReadLine());
    Console.WriteLine("Informe nº Y");
    y = int.Parse(Console.ReadLine());
    divisao(x, y);
    mensagem();
    Console.ReadKey();
    break;
```

```
static void divisao(int numero1, int numero2)
{
    int x;
    x = numero1 / numero2;
    Console.WriteLine("Resultado: " + x);
}
```

file:///C:/Users/Eduardo/documents/visual studio 2013/Projects/CA_HelloWorld/C

```
Informe um Nº 1 a 5, 6 = Sair, 7 Limpar Interface!!
4
Informe nº X
10
Informe nº Y
2
Resultado: 5
Enter para Continuar
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Lista de Exercícios

1. Informar os números pares entre 100 a 200.
2. Escrever um programa em C# que lê 5 valores para a, um de cada vez, e conta quantos destes valores são negativos, escrevendo esta informação.
3. Escrever um programa em C# que informe a soma dos números entre 0 e 100.
4. Escrever um programa em C# que informe a soma dos números pares entre 0 e 100.
5. Informar a soma dos números múltiplos de 7 entre 100 e 200. **(2107)**
6. Escreva um programa que que ao informar um número ele calcule a tabuada de 0 a 10.
7. Construa um menu interativo para demonstrar os exercícios desenvolvidos.

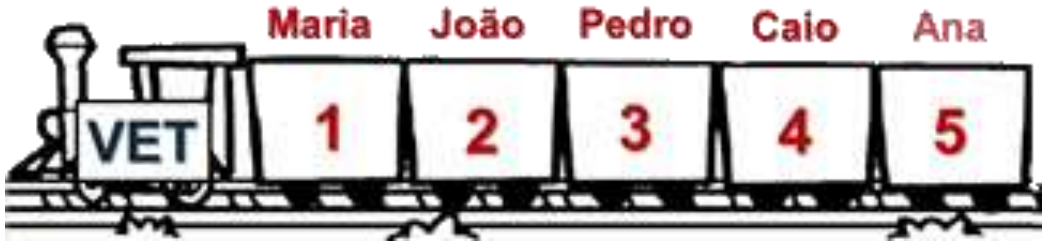
Queremos guardar uma lista de contas de nosso banco para poder trabalhar com elas. Uma primeira alternativa seria criar um conjunto de variáveis, no qual cada variável aponta para uma Conta diferente:

Mas, e se quisermos imprimir o saldo de todas elas? Precisaremos escrever N linhas, uma para cada Conta:

Array
VET

Arrays : Atribuição Direta

Vet[1] = “Maria”
Vet[2] = “João”
Vet[3] = “Pedro”
Vet[4] = “Caio”
Vet[5] = “Ana”



Trabalhando com arrays

```
#region arrays1

int[] numeros = new int[5];

numeros[0] = 1;
numeros[1] = 600;
numeros[2] = 257;
numeros[3] = 12;
numeros[4] = 42;

Console.WriteLine(numeros[2]);
Console.ReadKey();

#endregion
```

Arrays : Atribuição Dinâmica

Trabalhando com arrays

```
int[] numeros = new int[5];
int y;

for (int x = 0; x < 5; x++)
{
    Console.WriteLine("Informe 5 Nºs");
    numeros[x] = int.Parse(Console.ReadLine());
}
Console.Clear();
Console.WriteLine("Buscar: Informe um Nº de 0 a 4");
Console.WriteLine(numeros[y = int.Parse(Console.ReadLine())]);
Console.ReadKey();
```

Trabalhando com arrays

```
#region arrays3

string[] nome = new string[5];
int y;

for (int x = 0; x < 5; x++)
{
    Console.WriteLine("Informe um Nome");
    nome[x] = Console.ReadLine();
}
Console.Clear();
Console.WriteLine("Buscar: Informe um Nº de 0 a 4");
Console.WriteLine(nome[y = int.Parse(Console.ReadLine())]);
Console.ReadKey();

#endregion
```

Função Substring(_, _);

```
static void Main(string[] args)
{
    string cpf = "12345678909";
    string cpfParc = cpf .Substring(0, 9);
    Console.WriteLine(cpfParc);
}
```

Posição

Quantidade

Método TryParse

```
#region TryParse

int resultado;
string valor;
Console.WriteLine("Informe um Nº Inteiro");
valor = Console.ReadLine();
if (int.TryParse(valor, out resultado))
{
    Console.WriteLine("Verdadeiro " + resultado.ToString());
}
else
{
    Console.WriteLine("Falso " + resultado.ToString());
}
Console.ReadKey();
#endregion
```

CPF – Cálculo do Dígito Verificador

CP O número do CPF é composto de 11 dígitos, sendo os dois últimos os dígitos de verificação.

123.456.789-09

111.111.111-11

000.000.000-00

1º Dígito Verificador

Faça a soma da multiplicação dos 9 primeiros dígitos por **10, 9, 8, ... , 3, 2,** respectivamente:

1 * 10 = 10
2 * 9 = 18
3 * 8 = 24
4 * 7 = 28
5 * 6 = 30
6 * 5 = 30
7 * 4 = 28
8 * 3 = 24
9 * 2 = 18

Soma = 210

Valor = (210\11)*11 = 209

Resultado = Soma – Valor
Resultado = 210 - 209

0

- Se **Resultado** for igual à **1** ou à **0**, então o 1º dígito verificador é **0**;
- Caso contrário, o 1º dígito verificador é o resultado da subtração de **11 - Resultado**.

2º Dígito Verificador

Primeiro calculamos a soma da multiplicação dos 9 primeiros dígitos por **11, 10, 9, … , 4, 3,** respectivamente e em seguida somamos com **(Digito1*2)**, sendo que **Digito1** é o valor encontrado para o 1º dígito verificador.

1 * 11 = 11
2 * 10 = 20
3 * 9 = 27
4 * 8 = 32
5 * 7 = 35
6 * 6 = 36
7 * 5 = 35
8 * 4 = 32
9 * 3 = 27
0 * 2 = 0

Soma = 255

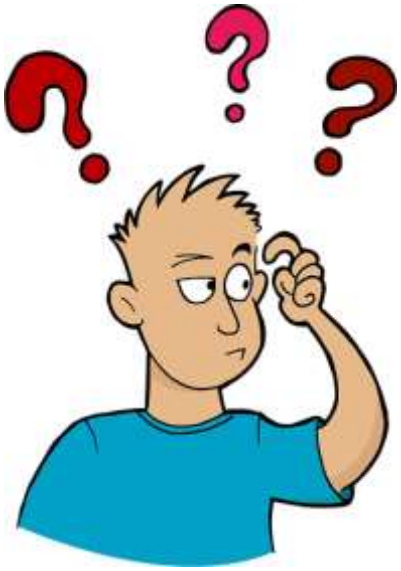
Valor = (255\11)*11 = 253

Resultado = Soma – Valor
Resultado = 255 - 253

11 – 2 = 9

- Se **Resultado** for igual à **1** ou à **0**, então o 1º dígito verificador é **0**;
- Caso contrário, o 1º dígito verificador é o resultado da subtração de **Resultado** de **11**.

Exercício



1. Construir um programa em C# que informe se o CPF informado é verdadeiro ou falso.

2. Construir um programa em C# que ao informar 9 dígitos o mesmo informe o digito verificador do CPF.

Capítulo 7: POO C#

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Introdução a Orientação a Objetos com C#

O termo orientação a objetos significa organizar o mundo real como uma coleção de objetos que incorporam estrutura de dados e um conjunto de operações que manipulam estes dados.

Introdução a Orientação a Objetos com C#

Classes
Em programação uma Classe representa uma entidade e como tal define o seu comportamento (Métodos) e as suas estruturas (Propriedades).

Entidade (Classe): Carro

Estrutura/Atributos (Propriedades): Cor, Modelo, Ano, Potência, Peso, Válvulas, etc
Comportamento/Ação (Métodos): Ligar, Desligar, Bater, afogar, etc

Atributo de Visibilidade
Private, public, protectec, readonly

Carros

```
public string Marca;
public string Modelo;
public int Ano;
```

Variáveis

```
public Ligar();
Private Concertar();
public Desligar();
```

Ação

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Introdução a Orientação a Objetos com C#

GRAU DE VISIBILIDADE

private - O método é acessível apenas pela própria classe.

public - Outras classe conseguem ver um método ou atributo.

protected - Um membro protegido é acessível dentro de sua classe e por instâncias da classe derivada.

Encapsulamento que em programação orientada a objetos significa separar o programa em partes, o mais isoladas possível. Objetos restringem a visibilidade de seus recursos (atributos e métodos) aos outros usuários. Todo objeto tem uma interface, que determina como os outros objetos podem interagir com ele. A implementação do objeto é encapsulada, isso é, invisível ou visível fora do próprio objeto.

```
class Point
{
    protected int x;
    protected int y;
}

class DerivedPoint: Point
{
    static void Main()
    {
        DerivedPoint dpoint = new DerivedPoint();

        // Direct access to protected members:
        dpoint.x = 10;
        dpoint.y = 15;
        Console.WriteLine("x = {0}, y = {1}", dpoint.x, dpoint.y);
    }
}
// Output: x = 10, y = 15
```

Introdução a Orientação a Objetos com C#

Classe????

Objeto????

Introdução a Orientação a Objetos com C#

Classe????

Objeto????

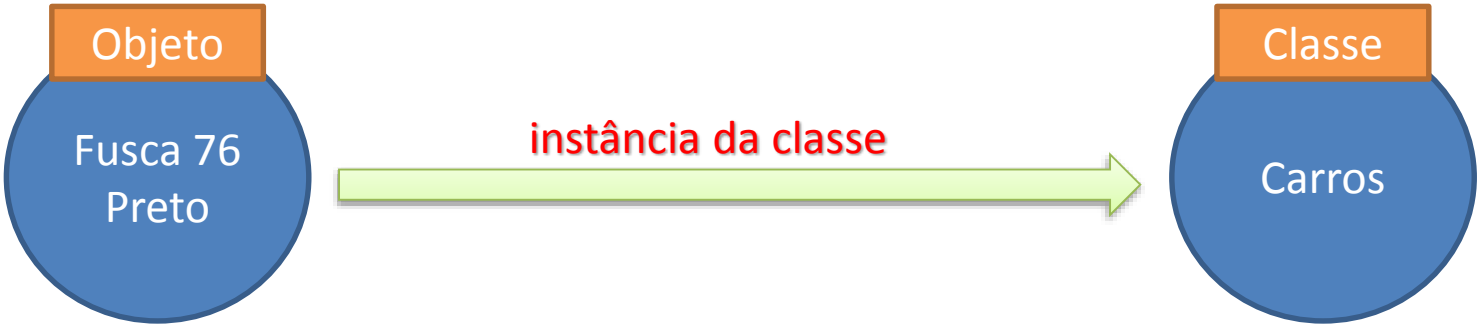
Introdução a Orientação a Objetos com C#

Comparação entre Classes e Objetos

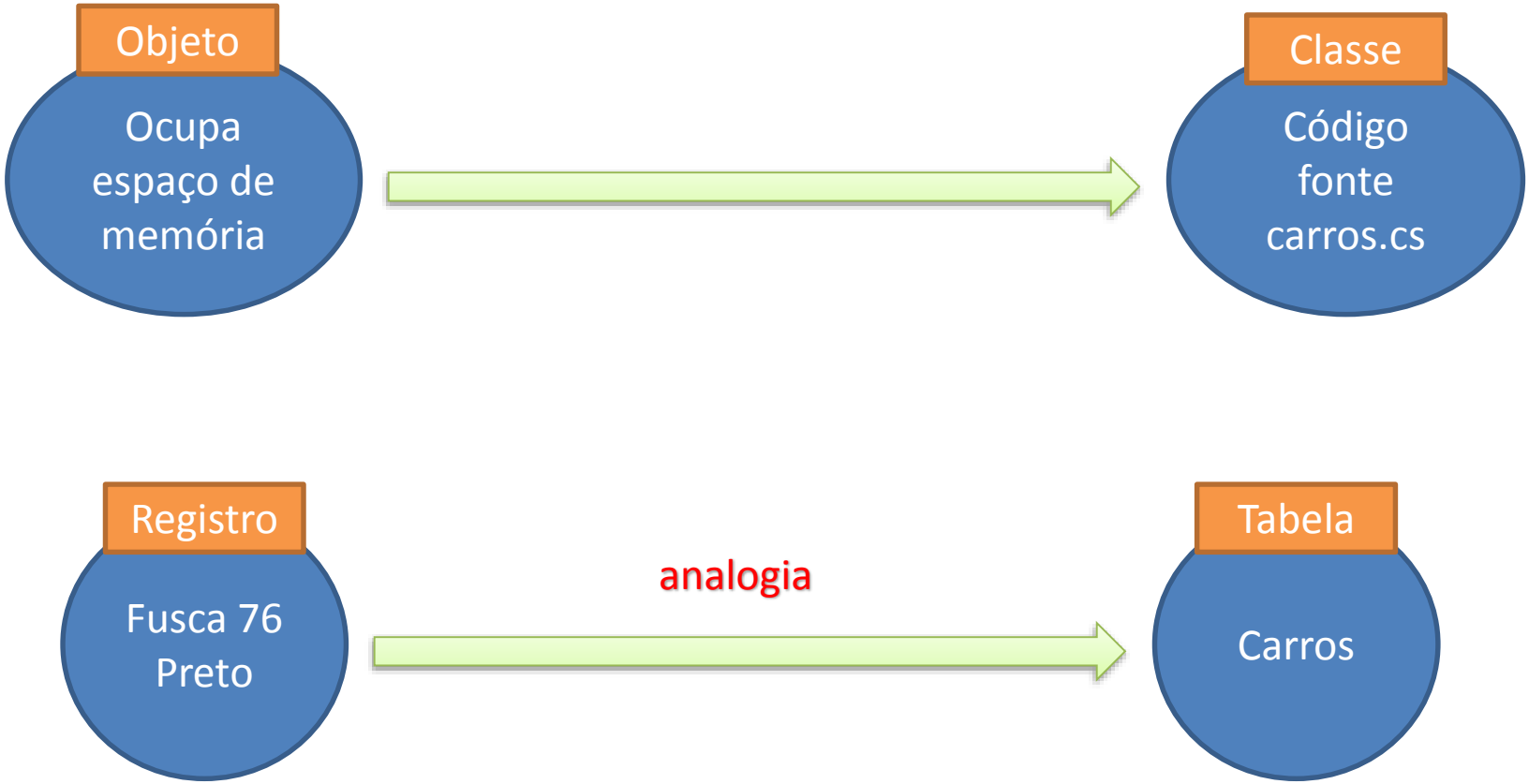
Uma Classe consiste na definição de uma entidade, (com as suas propriedades, métodos e eventos) enquanto um objeto representa um "exemplar" de uma entidade.

Ex: Classe **Carros**

Objeto: **Fusca**



Introdução a Orientação a Objetos com C#



Exercício

Crie 5 Classes e informe as suas Estruturas e seus comportamentos.

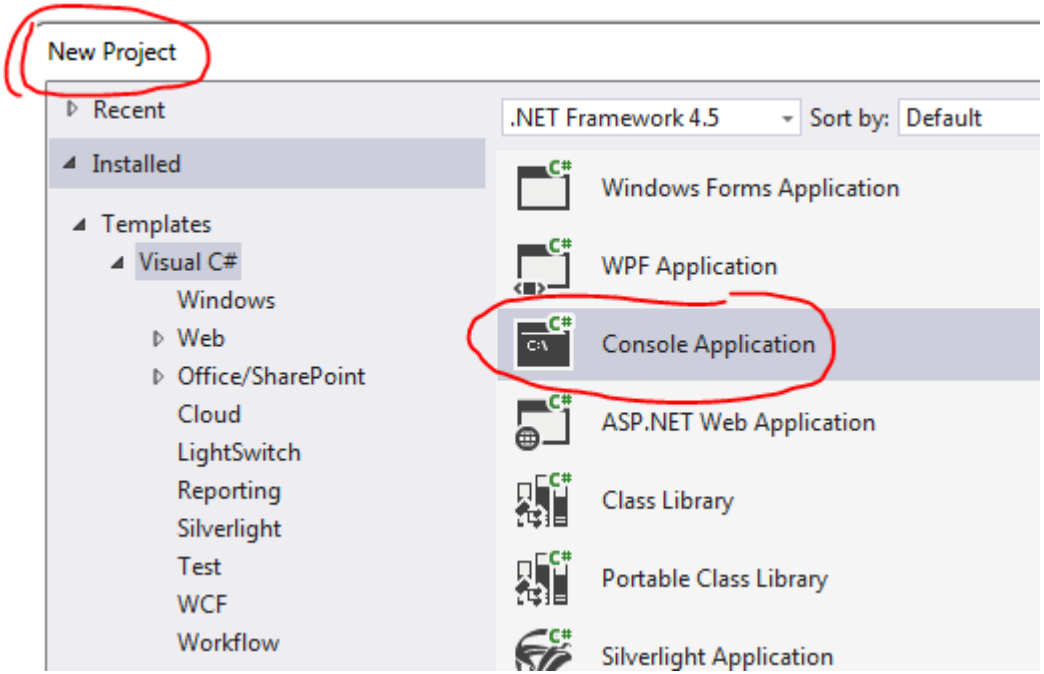
Carros

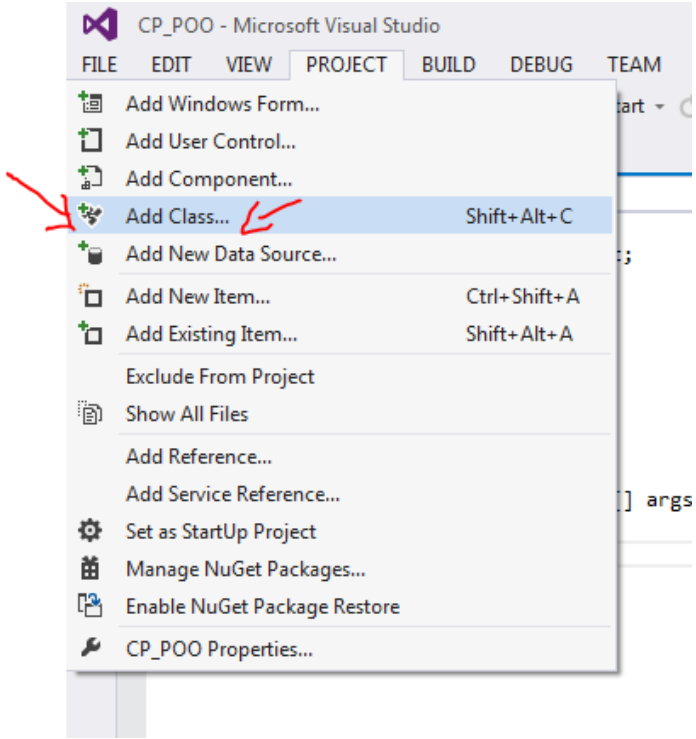
public string Marca;
public string Modelo;
public int Ano;

public Ligar();
Private Concertar();
public Desligar();

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

POO na Prática





```
namespace CP_POO
{
    class Program
    {
        static void Main(string[] args)
        {
            ClassCarro fusca = new ClassCarro();
            string estado;
            fusca.Ligar();
            estado = fusca.Situacao;
            Console.WriteLine(estado.ToString());
        }
    }

}
```

C:\Windows\system32\cmd.exe

```
Carro Ligado
Pressione qualquer tecla para continuar. . .
```

```
7   namespace CP_POO
8   {
9       class ClassCarro
10      {
11          //Atributos
12          public string Marca         { get; set; }
13          public string Cor           { get; set; }
14          public string Modelo        { get; set; }
15          public string Ano           { get; set; }
16          public string Potencia      { get; set; }
17          public string Peso          { get; set; }
18          public string Valvulas      { get; set; }
19          public string Situacao      { get; set; }
20
21          //Ações
22          public void Ligar()
23          {
24              Situacao = "Carro Ligado";
25          }
26
27          public void Desligar()
28          {
29              Situacao = "Carro Desligado";
30          }
31
32          public void Afogar()
33          {
34              Desligar();
35          }
36
37
38      }
39  }
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

```
7  namespace CP_POO
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             ClassCarro fusca = new ClassCarro();
14             string estado, modelo;
15             fusca.Modelo = "Fusca";
16             fusca.Ligar();
17             modelo = fusca.Modelo;
18             estado = fusca.Situacao;
19             Console.WriteLine(modelo.ToString());
20             Console.WriteLine(estado.ToString());
21         }
22     }
23
24 }
```

```
7  namespace CP_POO
8  {
9      class ClassCarro
10     {
11         //Atributos
12         public string Marca         { get; set; }
13         public string Cor           { get; set; }
14         public string Modelo        { get; set; }
15         public string Ano           { get; set; }
16         public string Potencia      { get; set; }
17         public string Peso          { get; set; }
18         public string Valvulas      { get; set; }
19         public string Situacao      { get; set; }
20
21         //Ações
22         public void Ligar()
23         {
24             Situacao = "Ligando o veículo: " + Modelo;
25         }
26
27         public void Desligar()
28         {
29             Situacao = "Desligando o veículo: " + Modelo;
30         }
31
32         public void Afogar()
33         {
34             Desligar();
35         }
36
37     }
38 }
```

C:\Windows\system32\cmd.exe

```
Fusca
Ligando o veículo: Fusca
Pressione qualquer tecla para continuar. . .
```

Introdução a Orientação a Objetos com C#

Além do conceito de classes e objetos, existem vários outros conceitos que devem ser observados numa linguagem de programação, para que esta possa ser considerada POO:

1. Herança
2. Construtores
3. Sobrecargas
4. Polimorfismo
5. Propriedades

Introdução a Orientação a Objetos com C#

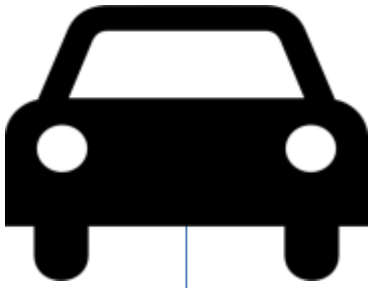
Herança

O reuso de código é uma das grandes vantagens da programação orientada a objetos. Muito disso se dá por uma questão que é conhecida como *herança*. Essa característica otimiza a produção da aplicação em tempo e linhas de código.

Vamos imaginar uma família: a criança, por exemplo, está herdando características de seus pais. Os pais, por sua vez, herdam algo dos avós, o que faz com que a criança também o faça, e assim sucessivamente. Na orientação a objetos, a questão é exatamente assim.

Introdução a Orientação a Objetos com C#

Classe

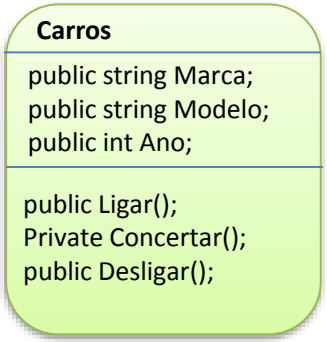


Herança

Velho

Novo

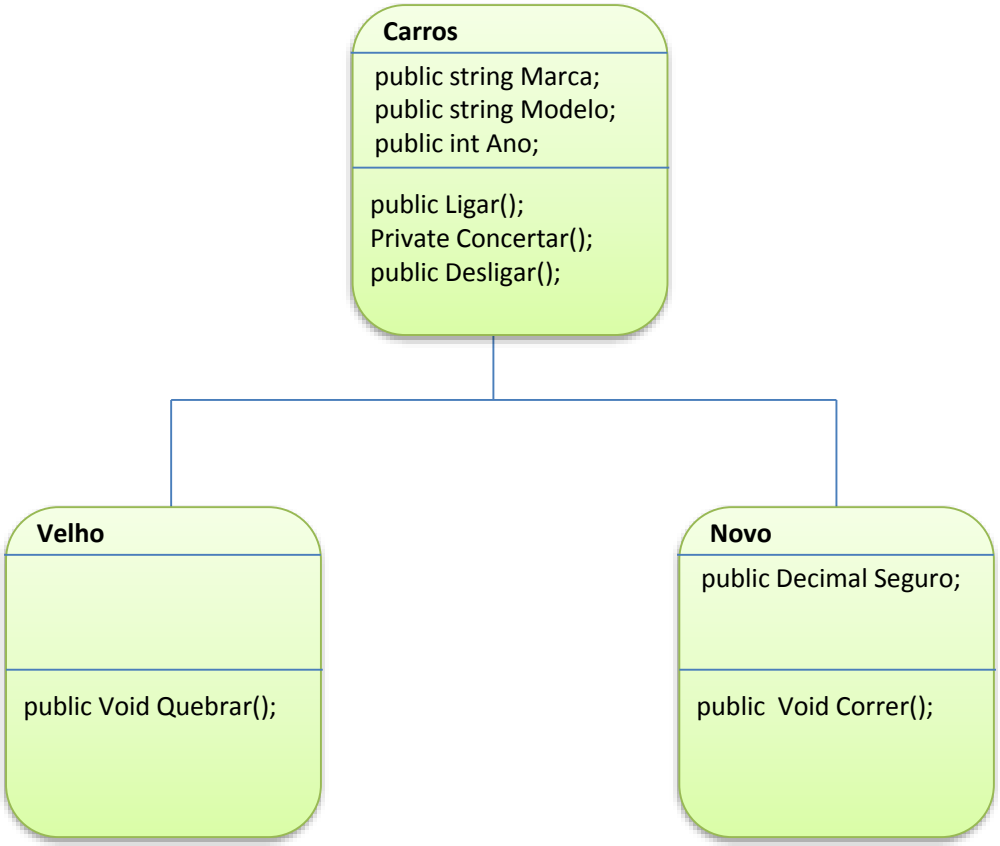
Introdução a Orientação a Objetos com C#




```
namespace CP_POO
{
    class Program
    {
        static void Main(string[] args)
        {
            Carros Fusca76Branco = new Carros();
            Fusca76Branco.Ligar();
        }

    }
}
```

```
class Carros
{
    public string Marca;
    public string Modelo;
    private int Ano;

    public void Ligar()
    {
        Console.WriteLine("Ligando o Veículo: " + Modelo);
    }

    public void Desligar()
    {
        Console.WriteLine("Desligando o Veículo: " + Modelo);
    }
}
```

Introdução a Orientação a Objetos com C#



Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Introdução a Orientação a Objetos com C#

```
namespace CP_POO
{
    class Program
    {
        static void Main(string[] args)
        {
            CarroVelho Fusca76Branco = new CarroVelho();
            Fusca76Branco.Ligar();
            Fusca76Branco.Quebrar();

        }

    }
}
```

```
class Carros
{
    public string Marca;
    public string Modelo;
    private int Ano;

    public void Ligar()
    {
        Console.WriteLine("Ligando o Veículo: " + Modelo);
    }

    public void Desligar()
    {
        Console.WriteLine("Desligando o Veículo: " + Modelo);
    }
}
```

```
namespace CP_POO
{
    class CarroVelho : Carros
    {
        public void Quebrar()
        {
            Console.WriteLine("O " + Modelo + " está Quebrado! Arrume um arame");
        }

    }
}
```

Introdução a Orientação a Objetos com C#

```
static void Main(string[] args)
{
    CarroVelho Fusca76Branco = new CarroVelho();
    Fusca76Branco.Ligar();
    Fusca76Branco.Quebrar();
}
```

void Carros.Desligar()

- Desligar
- Equals
- GetHashCode
- GetType
- Ligar
- Marca
- Modelo
- Quebrar
- ToString

```
class Carros
{
    public string Marca;
    public string Modelo;
    private int Ano;

    public void Ligar()
    {
        Console.WriteLine("Ligando o Veículo: " + Modelo);
    }

    public void Desligar()
    {
        Console.WriteLine("Desligando o Veículo: " + Modelo);
    }
}
```

```
namespace CP_POO
{
    class CarroVelho : Carros
    {
        public void Quebrar()
        {
            Console.WriteLine("O " + Modelo + " está Quebrado! Arrume um arame");
        }

    }
}
```

Introdução a Orientação a Objetos com C#

Classes

Em programação uma Classe representa uma entidade e como tal define o seu comportamento (Métodos) e as suas estruturas (Propriedades).

Entidade (Classe): **CarroNovo**

Estrutura/Atributos (Propriedades): Cor, Modelo, Ano, Potência, Peso, Válvulas, **Seguro,** etc

Comportamento/Ação (Métodos): Ligar, Desligar, Bater, afogar, **Correr**, etc

Entidade (Classe): **CarroVelho**

Estrutura/Atributos (Propriedades): Cor, Modelo, Ano, Potência, Peso, Válvulas, etc

Comportamento/Ação (Métodos): Ligar, Desligar, Bater, afogar, **Quebrar**, etc

Introdução a Orientação a Objetos com C#

Entidade (Classe): **CarroNovo**

Estrutura/Atributos (Propriedades): Cor, Modelo, Ano, Potência, Peso, Válvulas, **Seguro,** etc
Comportamento/Ação (Métodos): Ligar, Desligar, Bater, afogar, **Correr**, etc

C:\Windows\system32\cmd.exe

```
Ligando o Veículo: Fusca 76
O Fusca 76 está Quebrado! Arrume um arame
Ligando o Veículo: Ferrari F150
O Ferrari F150 está em alta velocidade!!!
Pressione qualquer tecla para continuar. . .
```

Introdução a Orientação a Objetos com C#

Entidade (Classe): **CarroNovo**

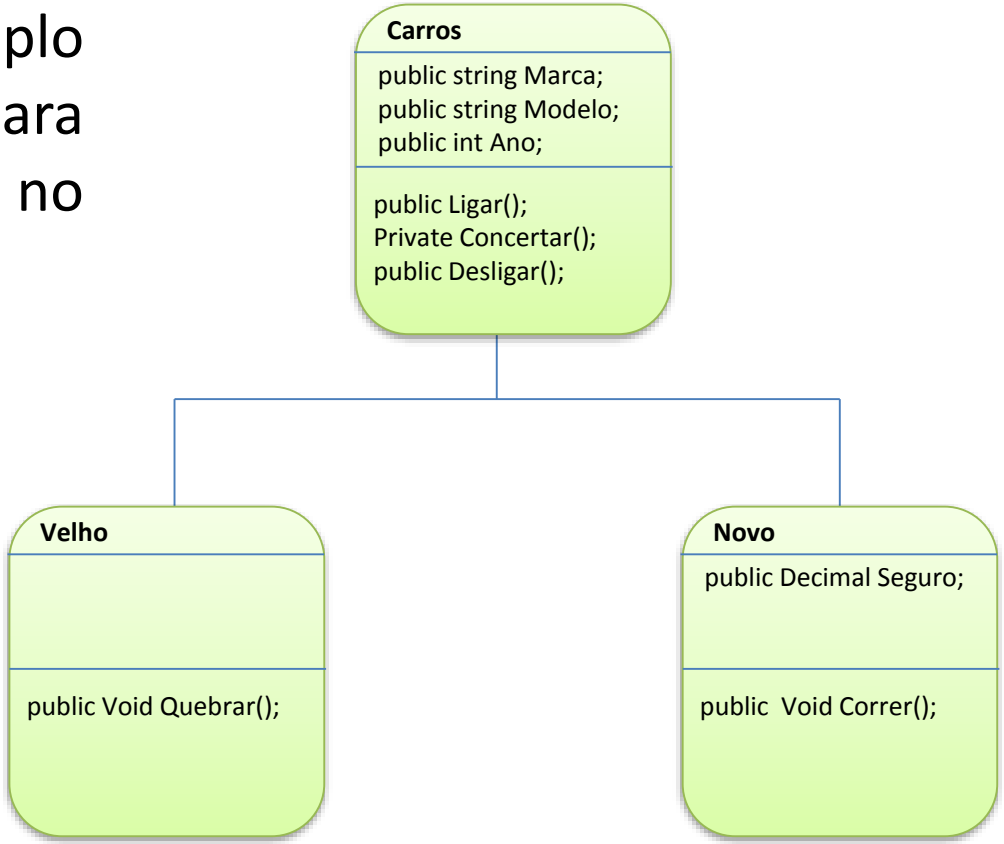
Estrutura/Atributos (Propriedades): Cor, Modelo, Ano, Potência, Peso, Válvulas, **Seguro,** etc
Comportamento/Ação (Métodos): Ligar, Desligar, Bater, afogar, **Correr**, etc

```
class Program
{
    static void Main(string[] args)
    {
        CarroVelho Fusca76Branco = new CarroVelho();
        Fusca76Branco.Modelo = "Fusca 76";
        Fusca76Branco.Ligar();
        Fusca76Branco.Quebrar();

        CarroNovo Ferrari = new CarroNovo();
        Ferrari.Modelo = "Ferrari F150";
        Ferrari.Correr();
    }

}
```

```
class CarroNovo : Carros
{
    public int Seguro;

    public void Correr()
    {
        Console.WriteLine("O " + Modelo + " está em alta velocidade!!!");
    }
}
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Utilizando o exemplo apresentado, crie heranças para as classes desenvolvidas no exercício anterior.



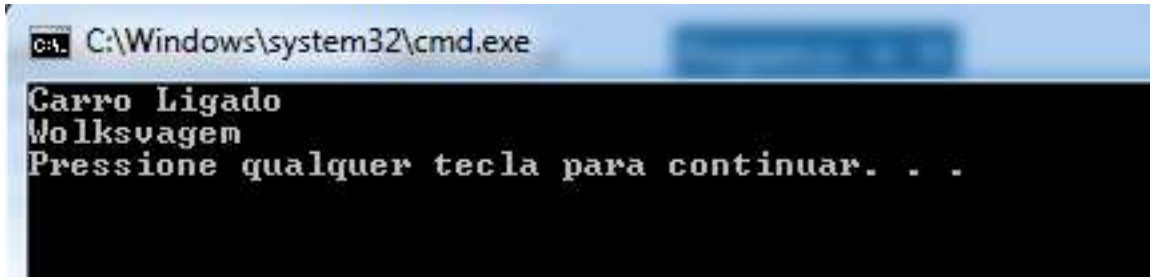
Construtor

Um Construtor é um método especial que roda automaticamente quando uma classe é instanciada.
Possui o mesmo nome da classe, e pode receber parâmetros (assinatura), mas não retorna valores.
Toda classe deve ter um construtor, se não for escrito, o compilador gerará um automaticamente um que não faz absolutamente nada.
Normalmente um construtor é usado para inicializar valores importantes ou executar uma tarefa quando o objeto é instanciado.

Construtor

```
7   namespace CP_POO
8   {
9       class Program
10      {
11          static void Main(string[] args)
12          {
13              ClassCarro fusca = new ClassCarro();
14              string estado, marca;
15              fusca.Ligar();
16              marca = fusca.Marca;
17              estado = fusca.Situacao;
18              Console.WriteLine(marca.ToString());
19              Console.WriteLine(estado.ToString());
20          }
21      }
22
23  }
```

```
class ClassCarro
{
    //Metodo Contrutor (valor Inicial da Variavel)
    public ClassCarro()
    {
        Marca       =   "Wolksvagem";
        Cor         =   "Preto";
        Modelo      =   "Fusca";
        Ano         =   "2016";
        Potencia    =   "145 CV";
        Peso        =   "1100";
        Valvulas    =   "16V";
        Situacao    =   "Desligado";
    }
```



Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Construtor

```
class CarroNovo : Carros
{
    public int Seguro;

    public CarroNovo(string Marca, string Modelo)
    {
        this.Marca = Marca;
        this.Modelo = Modelo;
    }

    public void Correr()
    {
        Console.WriteLine("O " + Modelo + " está em alta velocidade!!!");
    }
}
```

```
public CarroVelho(string Marca, string Mo
 {
     this.Marca = Marca;
os.AnoVeiculo
                Equals
                GetHashCode
                GetType
                Ligar
                Marca
                MemberwiseClone
                Modelo
                Quebrar
                ToString
 }

 public voi
 {
     Consol                                  + " e
 }

 public voi
 {
     if (Es
         Console.WriteLine("Esta muito fri
     else
         Console.WriteLine("Ligando o Veíc
 }
```

```
class CarroVelho : Carros
{
    public CarroVelho(string Marca, string Modelo)
     {
         this.Marca = Marca;
         this.Modelo = Modelo;
      }

    public void Quebrar()
    {
        Console.WriteLine("O " + Modelo + " está Quebrado! Arrume um arame");
    }
}
```

Construtor

```
7   namespace CP_POO
8   {
9       class Program
10      {
11          static void Main(string[] args)
12          {
13              CarroVelho Fusca76Branco = new CarroVelho();
```

CarroVelho.CarroVelho(string Marca, string Modelo)

Error:
'CP_POO.CarroVelho' does not contain a constructor that takes 0 arguments

```
19              Ferrari.Modelo = "Ferrari F150";
20              Ferrari.Ligar();
21              Ferrari.Correr();
22          }
23
24      }
25  }
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Construtor

```
namespace CP_POO
{
    class Program
    {
        static void Main(string[] args)
        {
            CarroVelho Fusca76Branco = new CarroVelho("VW", "Fusca");
            Fusca76Branco.Modelo = "Fusca 76";
            Fusca76Branco.Ligar();
            Fusca76Branco.Quebrar();

            CarroNovo Ferrari = new CarroNovo("Ferrari", "F150");
            Ferrari.Modelo = "Ferrari F150";
            Ferrari.Ligar();
            Ferrari.Correr();
        }

    }
}
```

C:\Windows\system32\cmd.exe

```
Ligando o Veículo: Fusca
O Fusca está Quebrado! Arrume um arame
Ligando o Veículo: F150
O F150 está em alta velocidade!!!
Pressione qualquer tecla para continuar. . .
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Sobrecarga

A sobrecarga de métodos (overload) é um conceito do polimorfismo que consiste basicamente em **criar variações de um mesmo método**, ou seja, a criação de dois ou mais métodos com nomes totalmente iguais em uma classe. A Sobrecarga permite que utilizemos o mesmo nome em mais de um método contanto que suas listas de argumentos sejam diferentes para que seja feita a separação dos mesmos.

```
public class calculadora{
 public int calcula( int a, int b){
    return a+b;
  }
  public double calcula( double a, double b){
     return a+b;
  }
   public String calcula( String a, String b){
     return a+b;
}
```

http://www.devmedia.com.br/sobrecarga-e-sobreposicao-de-metodos-em-orientacao-a-objetos/33066

Introdução a Orientação a Objetos com C#

Polimorfismo

Outro ponto essencial na programação orientada a objetos é o chamado polimorfismo. Na natureza, vemos animais que são capazes de alterar sua forma conforme a necessidade, e é dessa ideia que vem o polimorfismo na orientação a objetos. Como sabemos, os objetos filhos herdam as características e ações de seus “ancestrais”. Entretanto, em alguns casos, é necessário que as ações para um mesmo método seja diferente. Em outras palavras, o polimorfismo consiste na alteração do funcionamento interno de um método herdado de um objeto pai.

Como um exemplo, temos um objeto genérico “Eletrodoméstico”. Esse objeto possui um método, ou ação, “Ligar()”. Temos dois objetos, “Televisão” e “Geladeira”, que não irão ser ligados da mesma forma. Assim, precisamos, para cada uma das classes filhas, reescrever o método “Ligar()”.

Introdução a Orientação a Objetos com C#

Classe

Entidade (Classe): Carro

Estrutura (Propriedades): Cor, Modelo, Ano, Potência, Peso, Válvulas, etc.
Comportamento/Ação (Métodos): **Ligar**, **Desligar**, Bater, afogar, etc.

Introdução a Orientação a Objetos com C#

Classe

```
class CarroVelho : Carros
{
    public CarroVelho(string Marca, string Modelo)
     {
         this.Marca = Marca;
         this.Modelo = Modelo;
      }

    public void Quebrar()
    {
        Console.WriteLine("O " + Modelo + " está Quebrado! Arrume um arame");
    }

    public void Ligar(bool EstaFrio)
    {
        if (EstaFrio == true)
            Console.WriteLine("Esta muito frio, o " + Modelo + " nao liga");
        else
            Console.WriteLine("Ligando o Veículo: " + Modelo);
    }
}
```

Polimorfismo

```
class CarroNovo : Carros
{
    public int Seguro;

    public CarroNovo(string Marca, string Modelo)
    {
        this.Marca = Marca;
        this.Modelo = Modelo;
    }

    public void Correr()
    {
        Console.WriteLine("O " + Modelo + " está em alta velocidade!!!");
    }

    public void Ligar(bool EstaFrio)
    {
        if (EstaFrio == true)
            Console.WriteLine("Esta muito frio, a " + Modelo + " liga sem problemas");
        else
            Console.WriteLine("Ligando o Veículo: " + Modelo);
    }
}
```

Polimorfismo

```
Ferrari.;
Ferrari.
```

- Desligar
- Equals
- GetHashCode
- GetType
- Ligar
- Marca
- Modelo
- Seguro
- ToString

void CarroNovo.Ligar(bool EstaFrio) (+ 1 overload(s))

```
namespace CP_POO
{
    class Program
    {
        static void Main(string[] args)
        {
            CarroVelho Fusca76Branco = new CarroVelho("VW", "Fusca");
            Fusca76Branco.Ligar(true);
            Fusca76Branco.Quebrar();

            CarroNovo Ferrari = new CarroNovo("Ferrari", "F150");
            Ferrari.Ligar(true);
            Ferrari.Correr();
        }

    }
}
```

```
Ferrari.Ligar();
```

▲ 1 of 2 ▼ void Carros.Ligar()

```
Ferrari.Ligar();
```

▲ 2 of 2 ▼ void CarroNovo.Ligar(**bool EstaFrio**)

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Polimorfismo

```
namespace CP_POO
{
    class Program
    {
        static void Main(string[] args)
        {
            CarroVelho Fusca76Branco = new CarroVelho("VW", "Fusca");
            Fusca76Branco.Ligar(true);
            Fusca76Branco.Quebrar();

            CarroNovo Ferrari = new CarroNovo("Ferrari", "F150");
            Ferrari.Ligar(true);
            Ferrari.Correr();
        }

    }
}
```

C:\Windows\system32\cmd.exe

```
Esta muito frio, o Fusca nao liga
O Fusca está Quebrado! Arrume um arame
Esta muito frio, a F150 liga sem problemas
O F150 está em alta velocidade!!!
Pressione qualquer tecla para continuar. . .
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Introdução a Orientação a Objetos com C#

Diferença entre **sobrecarga** com **polimorfismo**. A diferença entre eles é que a sobrecarga consiste em implementações diferentes de um método na mesma classe, enquanto polimorfismo consiste em mudar o comportamento de um método de uma classe em classes descendentes.

Propriedades

Get: é um método que permite o acesso de **leitura** a um atributo.

Set: é um método que permite o acesso de **escrita** a um atributo.

Propriedades

```
namespace CP_POO
{
    class Program
    {
        static void Main(string[] args)
        {
            CarroVelho Fusca76Branco = new CarroVelho("VW", "Fusca");
            Fusca76Branco.Ligar(true);
            Fusca76Branco.Quebrar();
            Fusca76Branco.AnoVeiculo = 1969;

            CarroNovo Ferrari = new CarroNovo("Ferrari", "F150");
            Ferrari.Ligar(true);
            Ferrari.Correr();
            Ferrari.AnoVeiculo = 2015;
        }

    }
}
```

C:\Windows\system32\cmd.exe

```
Esta muito frio, o Fusca nao liga
O Fusca está Quebrado! Arrume um arame
Ano do Veículo: 1969
Esta muito frio, a F150 liga sem problemas
O F150 está em alta velocidade!!!
Ano do Veículo: 2015
Pressione qualquer tecla para continuar. . .
```

```
class Carros
{
    public string Marca;
    public string Modelo;
    private int Ano;

    public int AnoVeiculo
    {
        get
        {
            return Ano;
        }
        set
        {
            if (value < 1920)
            {
                Ano = 0;
                Console.WriteLine("Ano do Veículo: " + Ano);
            }

            else
            {
                Ano = value;
                Console.WriteLine("Ano do Veículo: " + Ano);
            }
        }
    }

    public void Ligar()
    {
        Console.WriteLine("Ligando o Veículo: " + Modelo);
    }

    public void Desligar()
    {
        Console.WriteLine("Desligando o Veículo: " + Modelo);
    }
}
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Propriedades

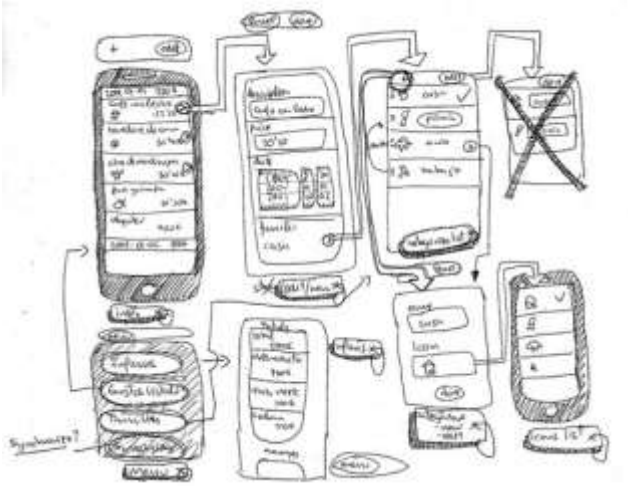
```
namespace CP_POO
{
    class Program
    {
        static void Main(string[] args)
        {
            CarroVelho Fusca76Branco = new CarroVelho("VW", "Fusca");
            Fusca76Branco.Ligar(true);
            Fusca76Branco.Quebrar();
            Fusca76Branco.AnoVeiculo = 1900;

            CarroNovo Ferrari = new CarroNovo("Ferrari", "F150");
            Ferrari.Ligar(true);
            Ferrari.Correr();
            Ferrari.AnoVeiculo = 2015;
        }

    }
}
```

C:\Windows\system32\cmd.exe

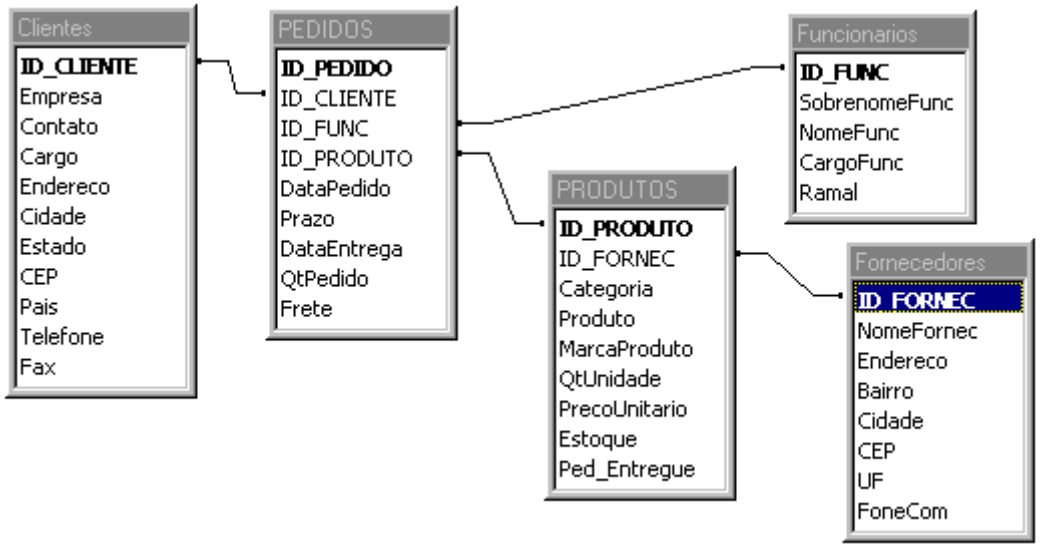
```
Esta muito frio, o Fusca nao liga
O Fusca está Quebrado! Arrume um arame
Ano do Veículo: 0
Esta muito frio, a F150 liga sem problemas
O F150 está em alta velocidade!!!
Ano do Veículo: 2015
Pressione qualquer tecla para continuar. . .
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Capítulo 8: Trabalho N1



Banco de Dados

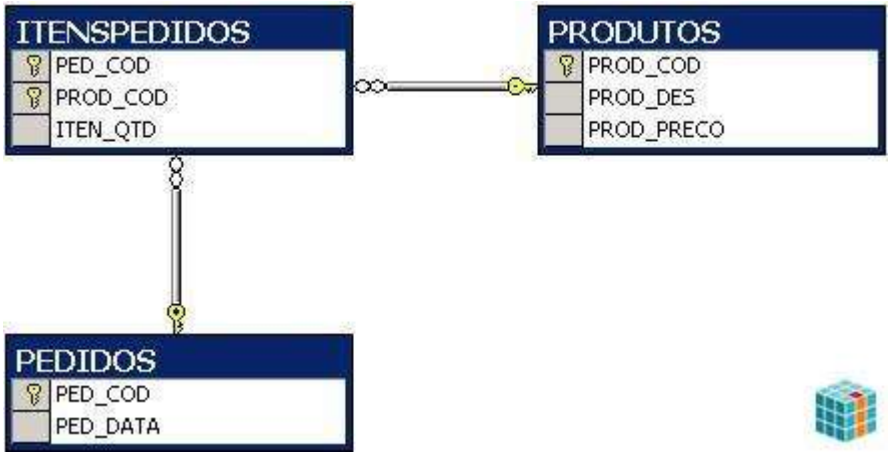


Banco de Dados

chave primária (PK)

chave estrangeira (FK)

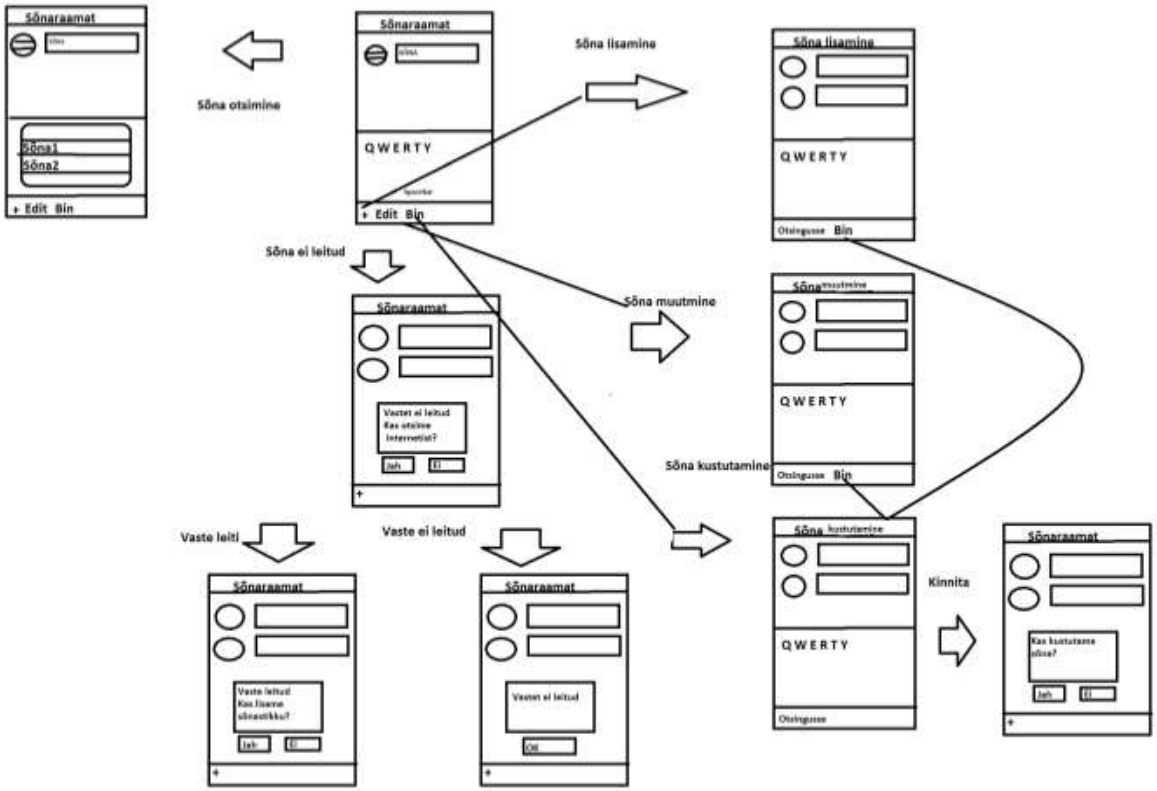
chave composta



Cidade

Cidade	Estado	Populacao
Rio Claro	SP	190373
Itu	SP	156100
Rio Claro	RJ	17826

Wireframes



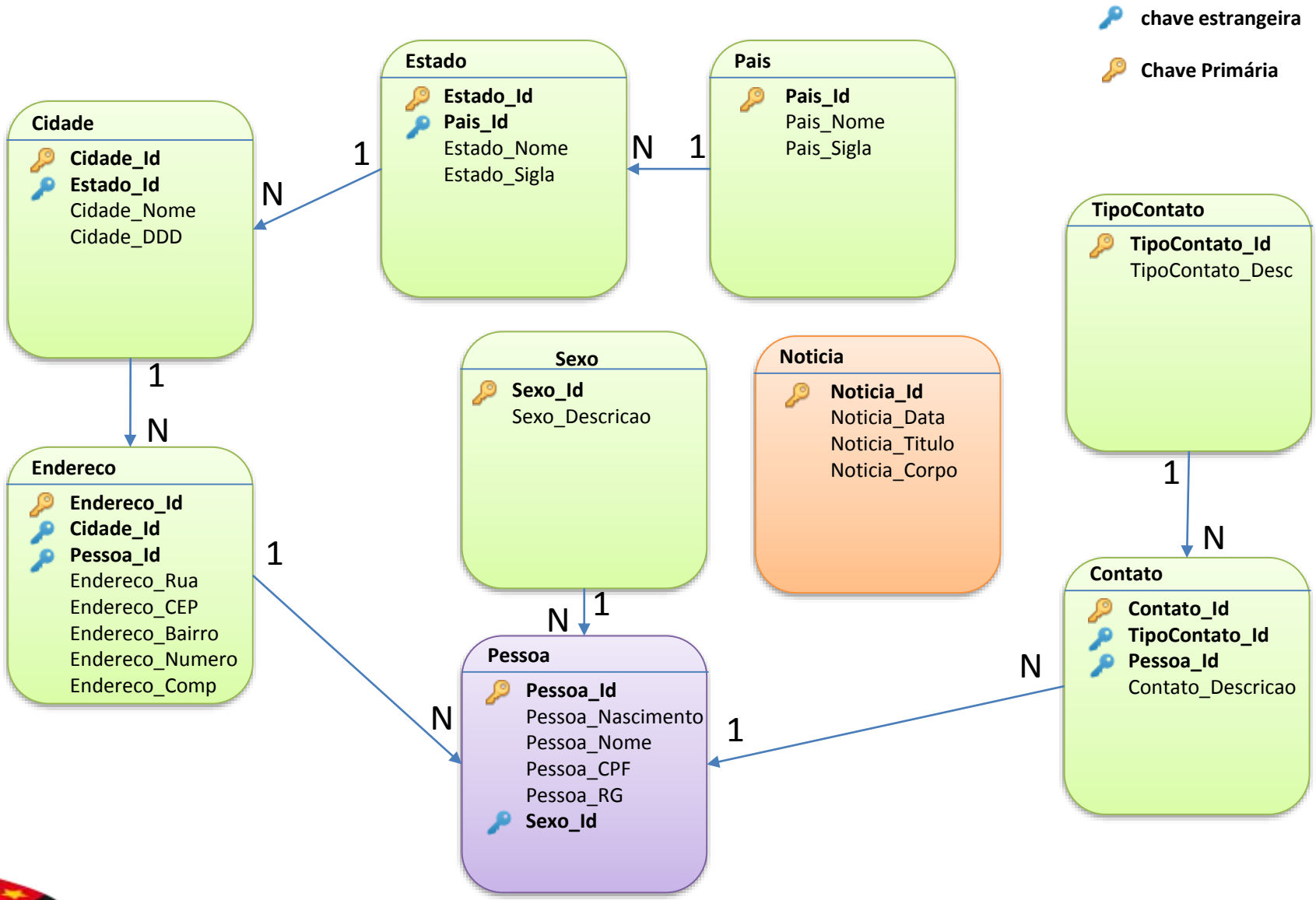
Wireframes

Projeto (proposta)

Pessoa

Endereço

Notícias



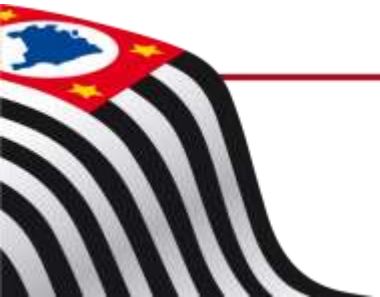
Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

Desenvolva um projeto em Console Application Utilizando os conceitos da Programação Orientada a Objetos (apresentação N1 07/04)

Tema: ????

Capítulo 8: ASP.NET



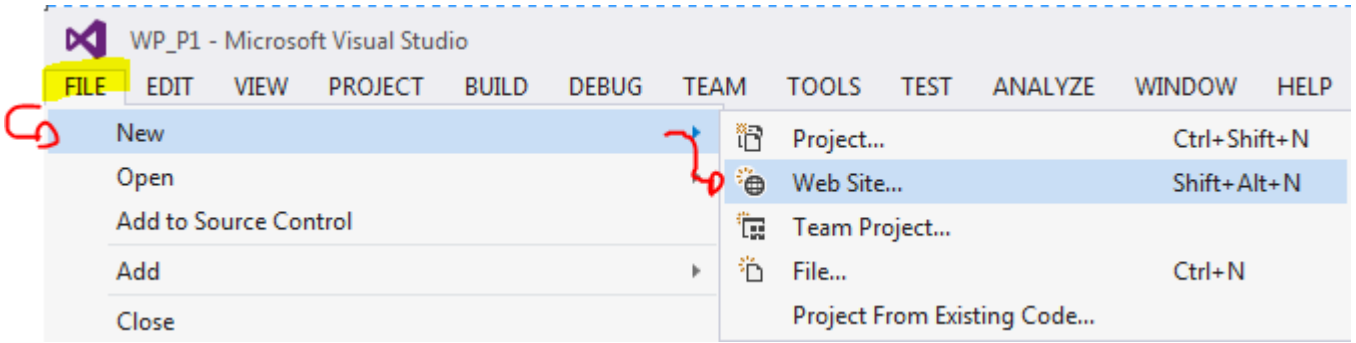
Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

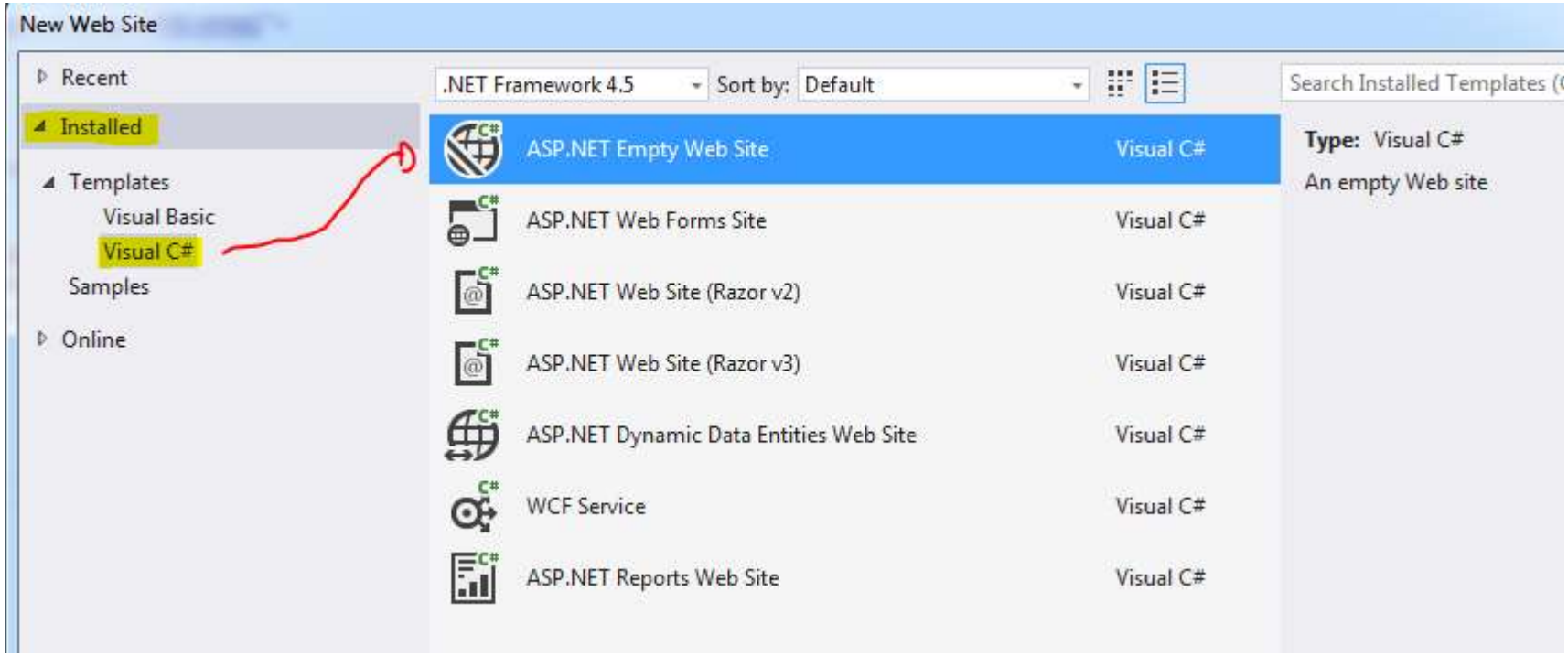
- **ASP.NET** é a plataforma da Microsoft para o desenvolvimento de aplicações Web e é o sucessor da tecnologia ASP. É um componente do IIS que permite através de uma linguagem de programação integrada na .NET Framework criar páginas dinâmicas.

- O ASP.NET é baseado no Framework .NET herdando todas as suas características, por isso, como qualquer aplicação .NET, as aplicações para essa plataforma podem ser escritas em várias linguagens, como C# e Visual Basic .NET.

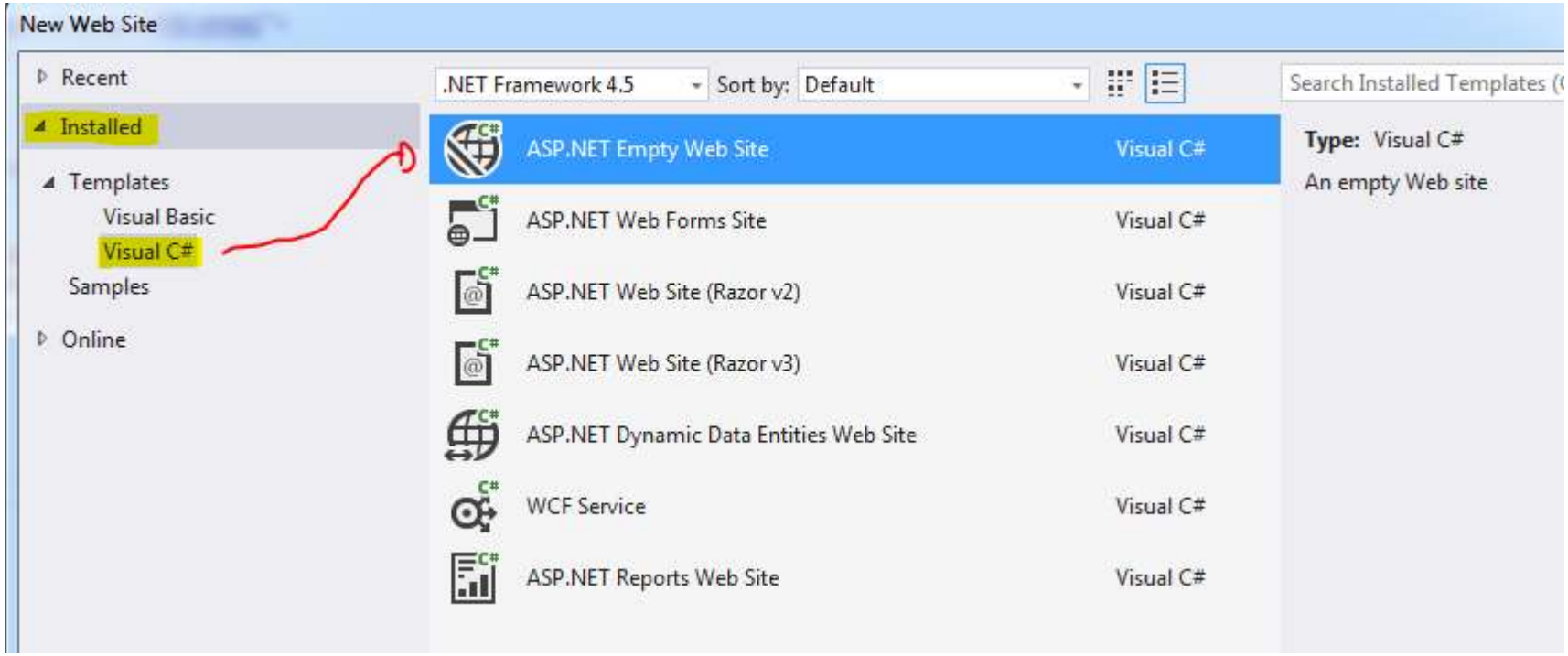
WEB FORMS


WP_P1 - Microsoft Visual Studio
FILE
EDIT
VIEW
PROJECT
BUILD
DEBUG
TEAM
TOOLS
TEST
ANALYZE
WINDOW
HELP
New
Open
Add to Source Control
Add
Close
Project...
Ctrl+Shift+N
Web Site...
Shift+Alt+N
Team Project...
File...
Ctrl+N
Project From Existing Code...

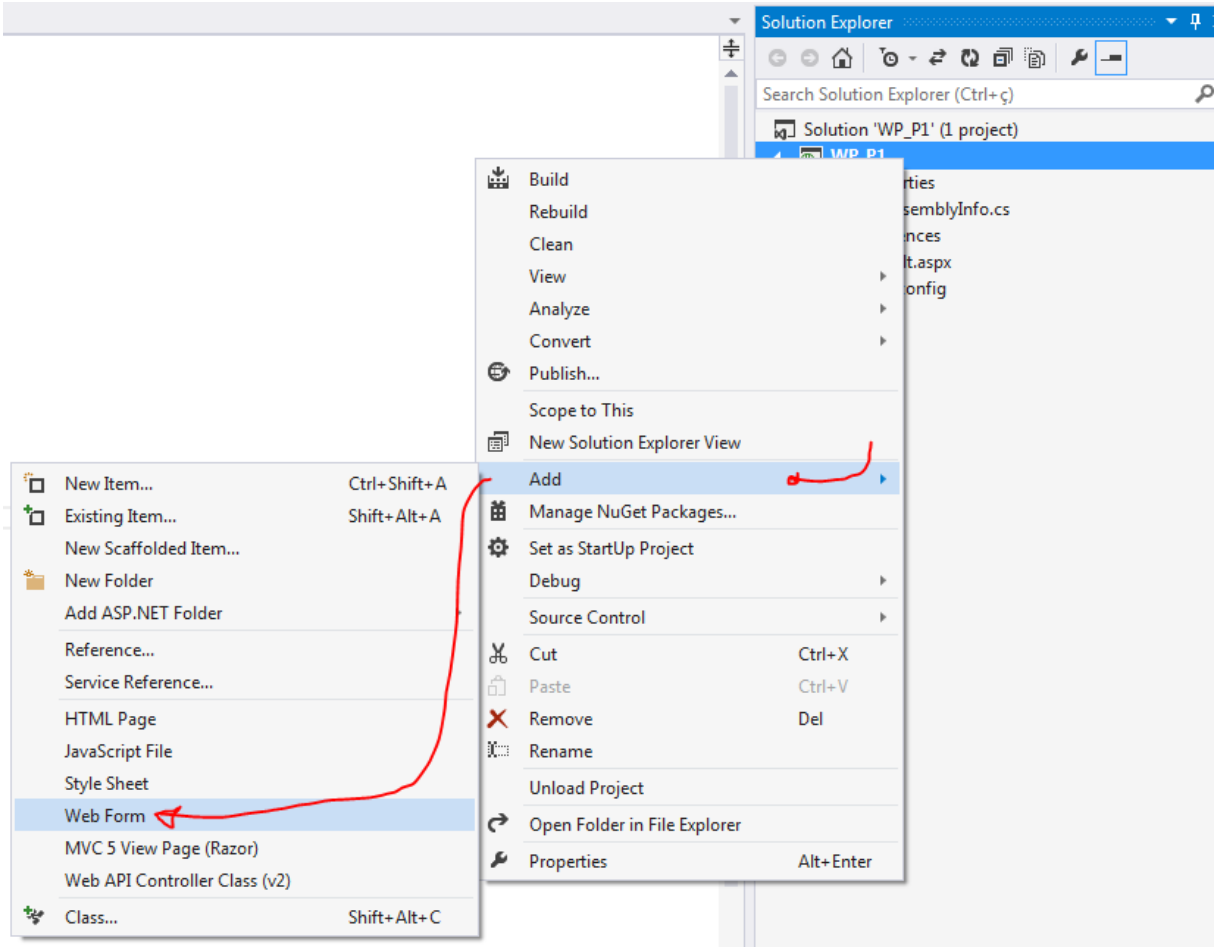
WEB FORMS


New Web Site
Recent
Installed
Templates
Visual Basic
Visual C#
Samples
Online
.NET Framework 4.5
Sort by: Default
Search Installed Templates
ASP.NET Empty Web Site
Visual C#
ASP.NET Web Forms Site
Visual C#
ASP.NET Web Site (Razor v2)
Visual C#
ASP.NET Web Site (Razor v3)
Visual C#
ASP.NET Dynamic Data Entities Web Site
Visual C#
WCF Service
Visual C#
ASP.NET Reports Web Site
Visual C#
Type: Visual C#
An empty Web site

WEB FORMS


New Web Site
Recent
Installed
Templates
Visual Basic
Visual C#
Samples
Online
.NET Framework 4.5
Sort by: Default
Search Installed Templates
ASP.NET Empty Web Site
Visual C#
ASP.NET Web Forms Site
Visual C#
ASP.NET Web Site (Razor v2)
Visual C#
ASP.NET Web Site (Razor v3)
Visual C#
ASP.NET Dynamic Data Entities Web Site
Visual C#
WCF Service
Visual C#
ASP.NET Reports Web Site
Visual C#
Type: Visual C#
An empty Web site

WEB FORMS



Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

WEB FORMS

Default.aspx

```
1   <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WP_P1.WebForm1" %>
2
3   <!DOCTYPE html>
4
5   <html xmlns="http://www.w3.org/1999/xhtml">
6   <head runat="server">
7       <title></title>
8   </head>
9   <body>
10      <form id="form1" runat="server">
11      <div>
12
13          <asp:TextBox ID="Txt_NomeUser" runat="server"></asp:TextBox>
14          <asp:Button ID="Bt_Salvar" runat="server" OnClick="Bt_Salvar_Click" Text="Button" />
```

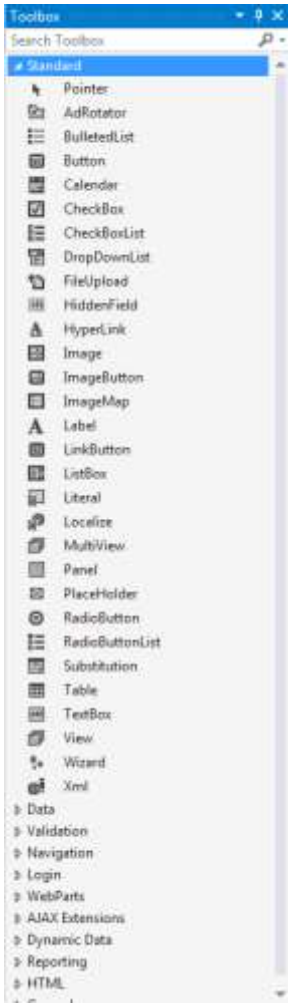
100 %

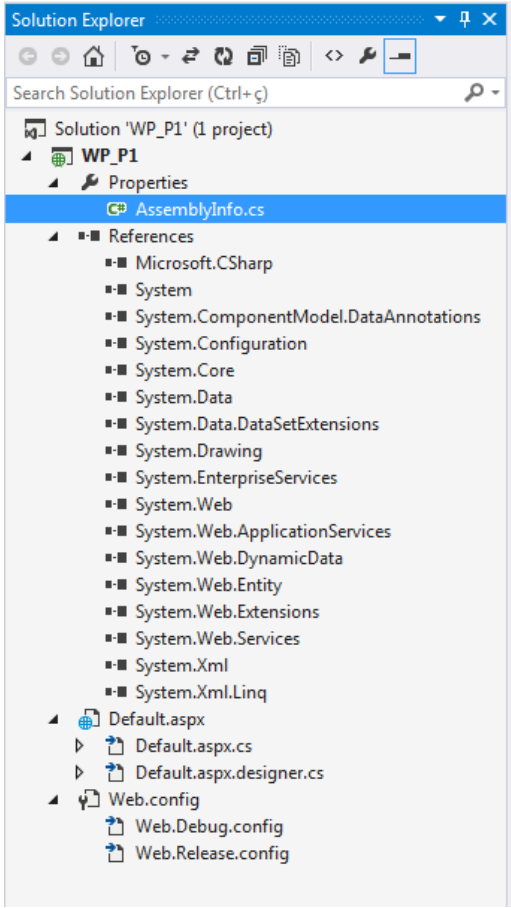
body

Button [Lb_Nome]

Design | Split | Source | <html> <body>

WEB FORMS

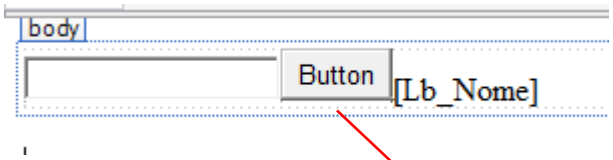




Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

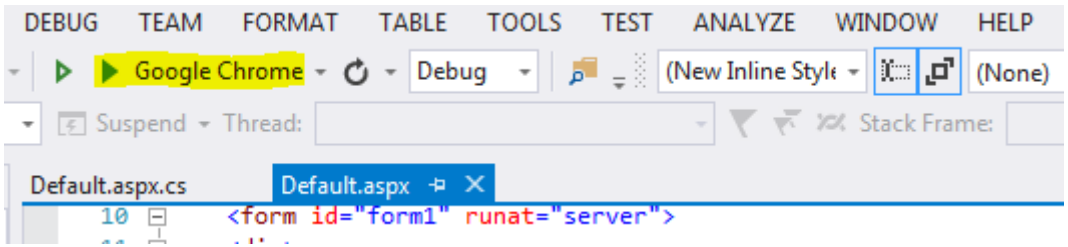
WEB FORMS



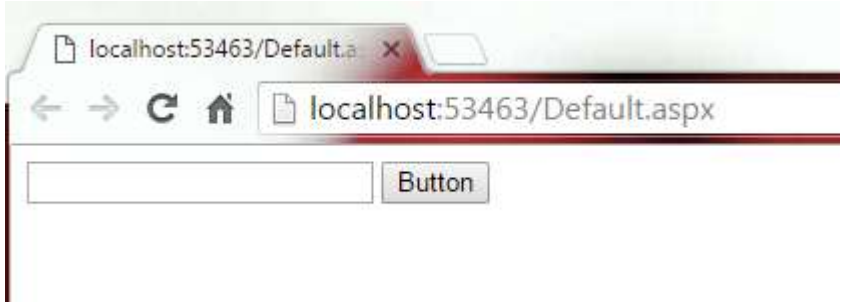

```
8   namespace WP_P1
9   {
10      public partial class WebForm1 : System.Web.UI.Page
11      {
12          protected void Page_Load(object sender, EventArgs e)
13          {
14
15          }
16
17          protected void Bt_Salvar_Click(object sender, EventArgs e)
18          {
19              Lb_Nome.Text = Txt_NomeUser.Text;
20          }
21      }
22  }
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

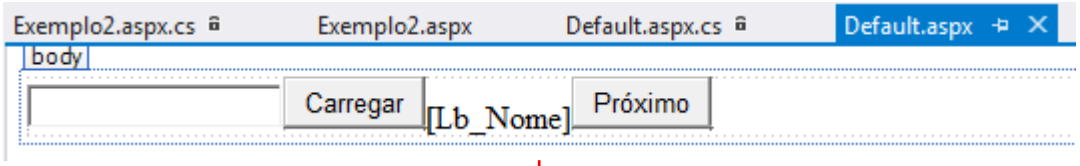
WEB FORMS

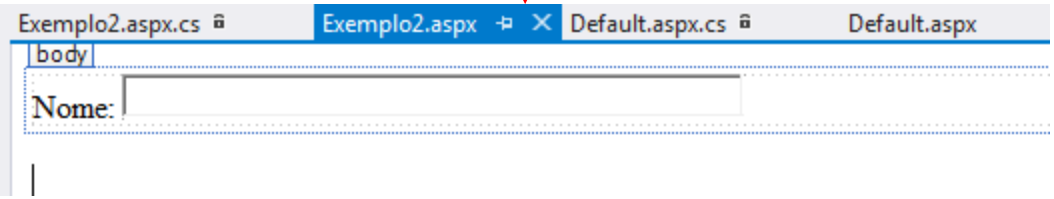

DEBUG TEAM FORMAT TABLE TOOLS TEST ANALYZE WINDOW HELP
Google Chrome
Debug
(New Inline Style
(None)
Suspend
Thread:
Stack Frame:
Default.aspx.cs
Default.aspx
10 <form id="form1" runat="server">
11

IIS Express
Personalizar...
106%
PT
22:47
05/09/2016


localhost:53463/Default.a
localhost:53463/Default.aspx
Button

WEB FORMS


Exemplo2.aspx.cs
Exemplo2.aspx
Default.aspx.cs
Default.aspx
body
Carregar
[Lb_Nome]
Próximo


Exemplo2.aspx.cs
Exemplo2.aspx
Default.aspx.cs
Default.aspx
body
Nome:

WEB FORMS

```
namespace WP_P1
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Bt_Salvar_Click(object sender, EventArgs e)
        {
            Lb_Nome.Text = Txt_NomeUser.Text;
        }

        protected void Bt_Proximo_Click(object sender, EventArgs e)
        {
            Response.Redirect("~/Exemplo2.aspx?Nome=" + Lb_Nome.Text);
        }
    }
}
```

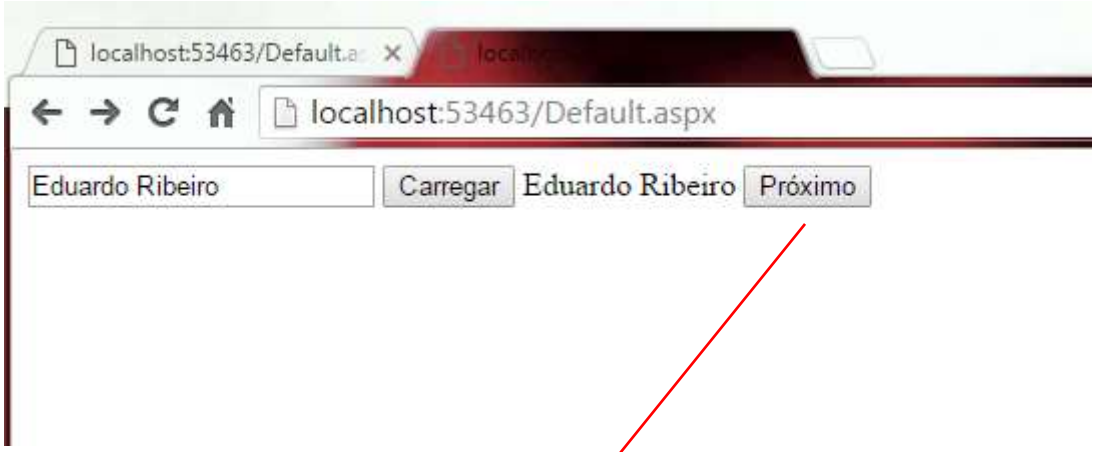
OBS.: ~ Diretório raiz

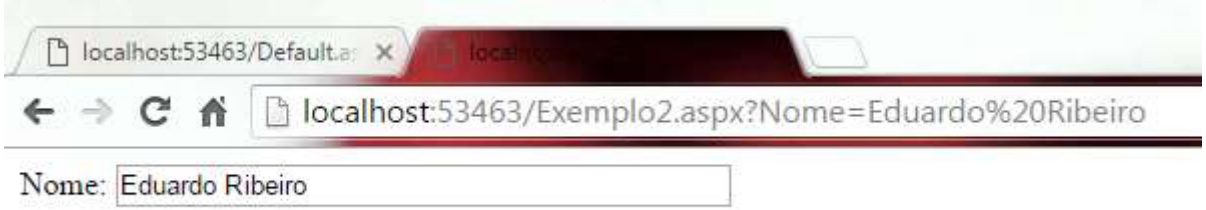
```
8   namespace WP_P1
9   {
10      public partial class Exemplo2 : System.Web.UI.Page
11      {
12          protected void Page_Load(object sender, EventArgs e)
13          {
14              Txt_Nome.Text = Request.QueryString["Nome"];
15          }
16      }
17  }
```

WEB FORMS

```
protected void Bt_Proximo_Click(object sender, EventArgs e)
{
    Response.Redirect("~/Exemplo2.aspx?Nome=" + Lb_Nome.Text+ "&Sobrenome=Ribeiro");
}
```

WEB FORMS


localhost:53463/Default.aspx
Eduardo Ribeiro
Carregar
Eduardo Ribeiro
Próximo


localhost:53463/Exemplo2.aspx?Nome=Eduardo%20Ribeiro
Nome:
Eduardo Ribeiro

WEB FORMS

Exemplo2.aspx.cs | Exemplo2.aspx | Default.aspx.cs | Default.aspx

body

Carregar [Lb_Nome]

Session

Próximo

```
protected void Bt_Session_Click(object sender, EventArgs e)
{
    Session["Nome"] = Txt_Email.Text;
}
```

Exemplo2.aspx.cs | Exemplo2.aspx | Default.aspx.cs

body

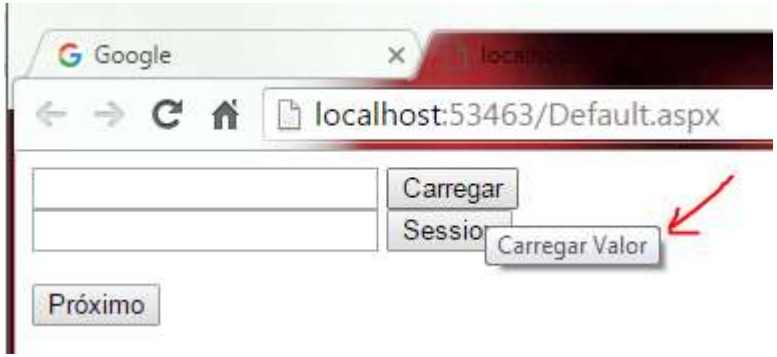
Nome:

Email:

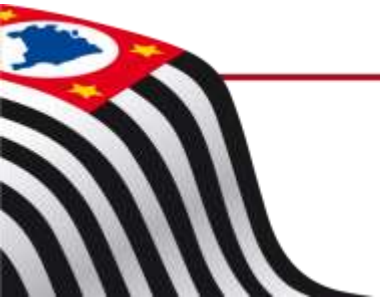
```
protected void Page_Load(object sender, EventArgs e)
{
    Txt_Nome.Text = Request.QueryString["Nome"];

    Txt_Email.Text = (string)(Session["Nome"]);
}
```

WEB FORMS

Height	
OnClientClick	
PostBackUrl	
SkinID	
TabIndex	0
Text	**Carregar**
ToolTip	**Carregar Valor**
UseSubmitBehavior	True
ValidateRequestMoc	Inherit
ValidationGroup	
ViewStateMode	Inherit
Visible	True
Width	



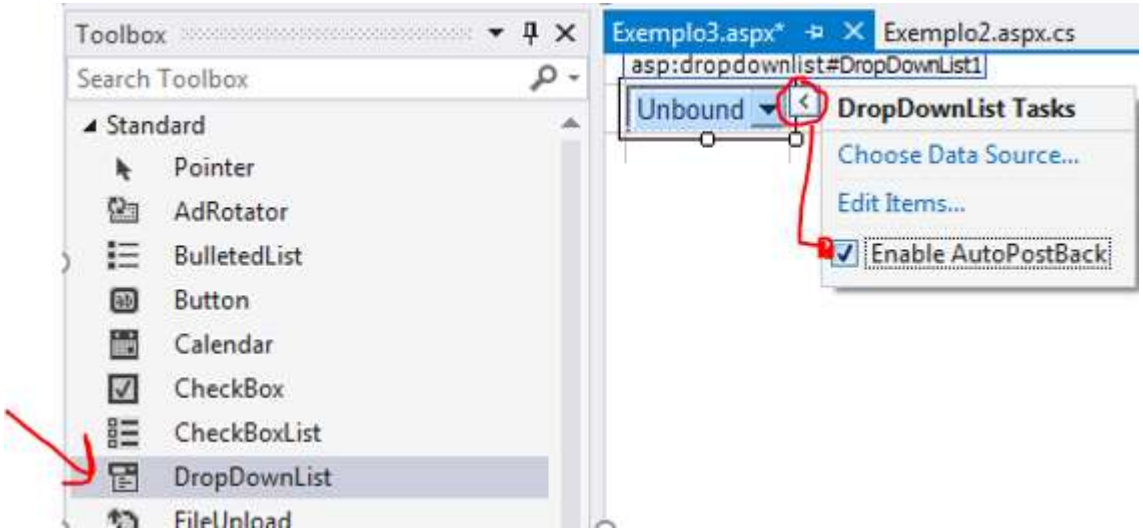
Capítulo 9: ASP.NET

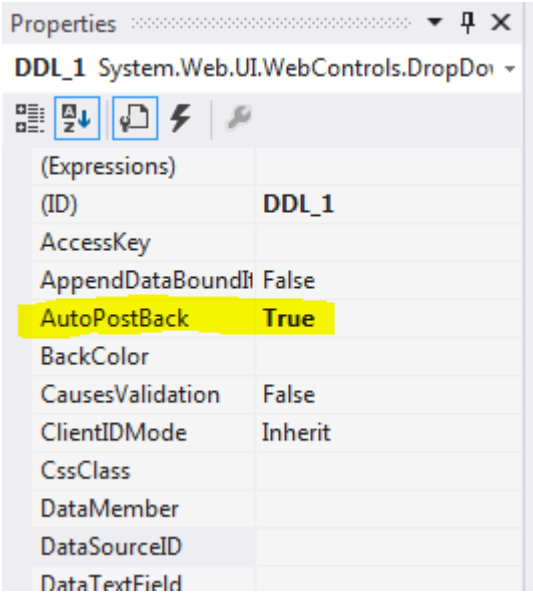


Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

WEB FORMS





postback é quando você executa algo e volta para a mesma página.

For WEB FORMS

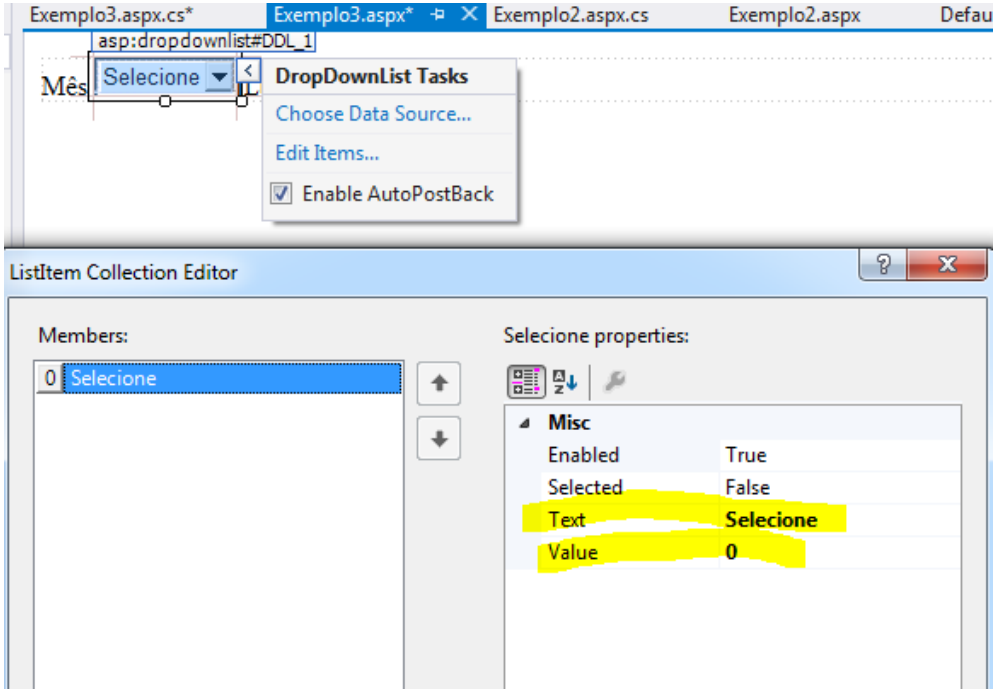
Exemplo3.aspx.cs | Exemplo3.aspx

body

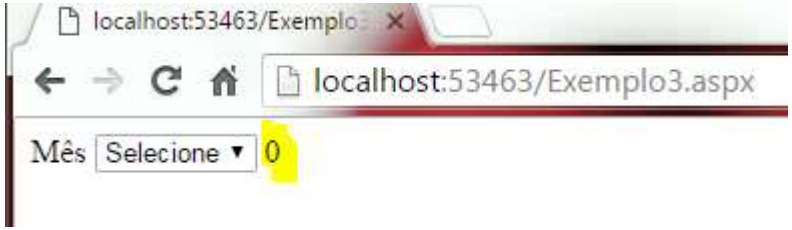
Mês Unbound [Lbl_Ddl]

```
public partial class Exemplo3 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack) //Se nao for postBack
        {
            for (int i = 1; i <=31; i++)
            {
                DDL_1.Items.Add(i.ToString());
            }
        }
    }

    protected void DDL_1_SelectedIndexChanged(object sender, EventArgs e)
    {
        Lbl_Ddl.Text = DDL_1.SelectedItem.ToString();
    }
}
```

WEB FORMS




```
Lbl_Ddl.Text = DDL_1.SelectedItem.Value.ToString();
```



While

WEB FORMS

Exemplo3.aspx.cs | Exemplo3.aspx

body

Mês Unbound [Lbl_Ddl]

```
public partial class dias_do_mes : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack) // se nao for postBack
        {
            int i = 1;
            while (i <= 31)
            {
                DropDownList1.Items.Add(i.ToString());
                i++;
            }
        }
    }
    protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
    {
        Label1.Text = DropDownList1.SelectedItem.ToString();
    }
}
```

Do.. While

WEB FORMS

Exemplo3.aspx.cs | Exemplo3.aspx

body

Mês Unbound [Lbl_Ddl]

```
public partial class dias_do_mes : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack) // se nao for postBack
        {
            int i = 1;
            do
            {
                DropDownList1.Items.Add(i.ToString());
                i++;
            }
            while (i <= 31);
        }
    }
    protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
    {
        Label1.Text = DropDownList1.SelectedItem.ToString();
    }
}
```

Exercício WEB FORMS

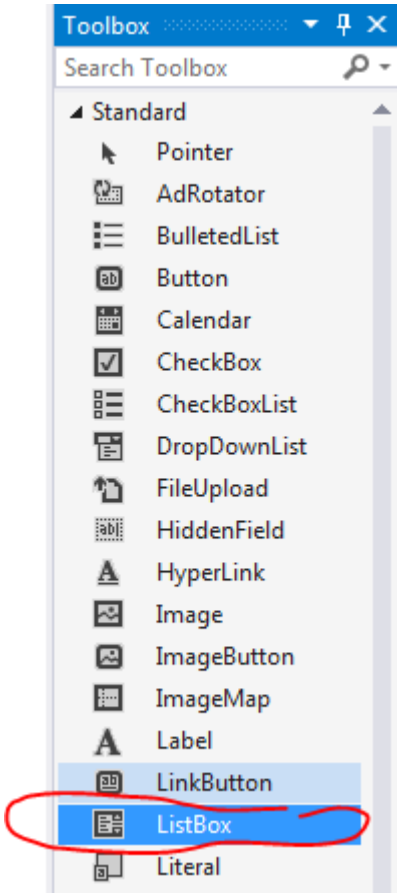
A. Construa uma página em ASPX contendo os seguintes dados:

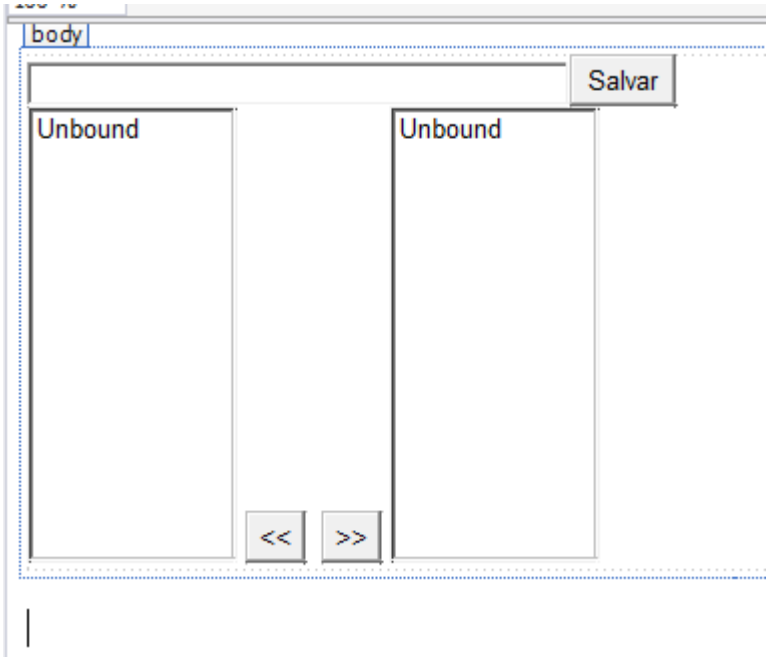
Nome:
Sobrenome:
Endereço:
Bairro:
Cidade:
CEP:
Telefone:
Celular:
Sexo:

B. Construir uma segunda página utilizando o componente label para receber os dados da página anterior.

1. Concatenar NOME + SOBRENOME.
2. Receber o valor (CEP) por Session.

ListBox WEB FORMS





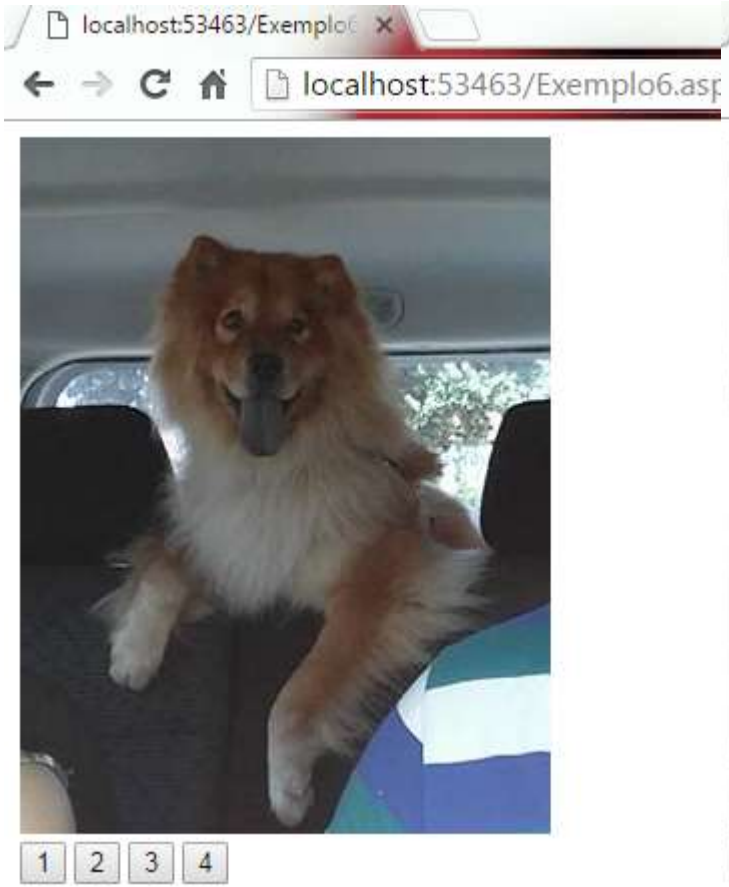
ListBox WEB FORMS

```
protected void Bt_Salvar_Click(object sender, EventArgs e)
{
    Lbx_Esq.Items.Add(TextBox1.Text);
    TextBox1.Text = null;
}

protected void Bt_Esquerdo_Click(object sender, EventArgs e)
{
    if (Lbx_Dir.SelectedIndex == -1)
    {
        Response.Write("Selecione um Item");
    }
    else
    {
        Lbx_Esq.Items.Add(Lbx_Dir.SelectedItem.ToString());
        Lbx_Dir.Items.Remove(Lbx_Dir.SelectedItem.ToString());
    }
}
```

```
protected void Bt_Direito_Click(object sender, EventArgs e)
{
    if (Lbx_Esq.SelectedIndex == -1)
    {
        Response.Write("Selecione um Item");
    }
    else
    {
        Lbx_Dir.Items.Add(Lbx_Esq.SelectedItem.ToString());
        Lbx_Esq.Items.Remove(Lbx_Esq.SelectedItem.ToString());
    }
}
```

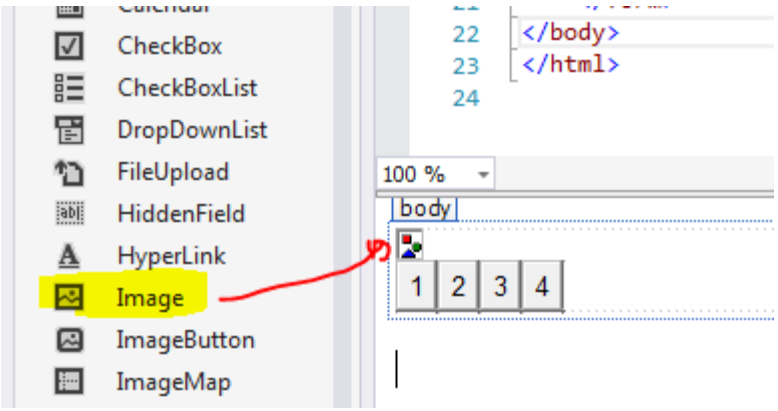
Image

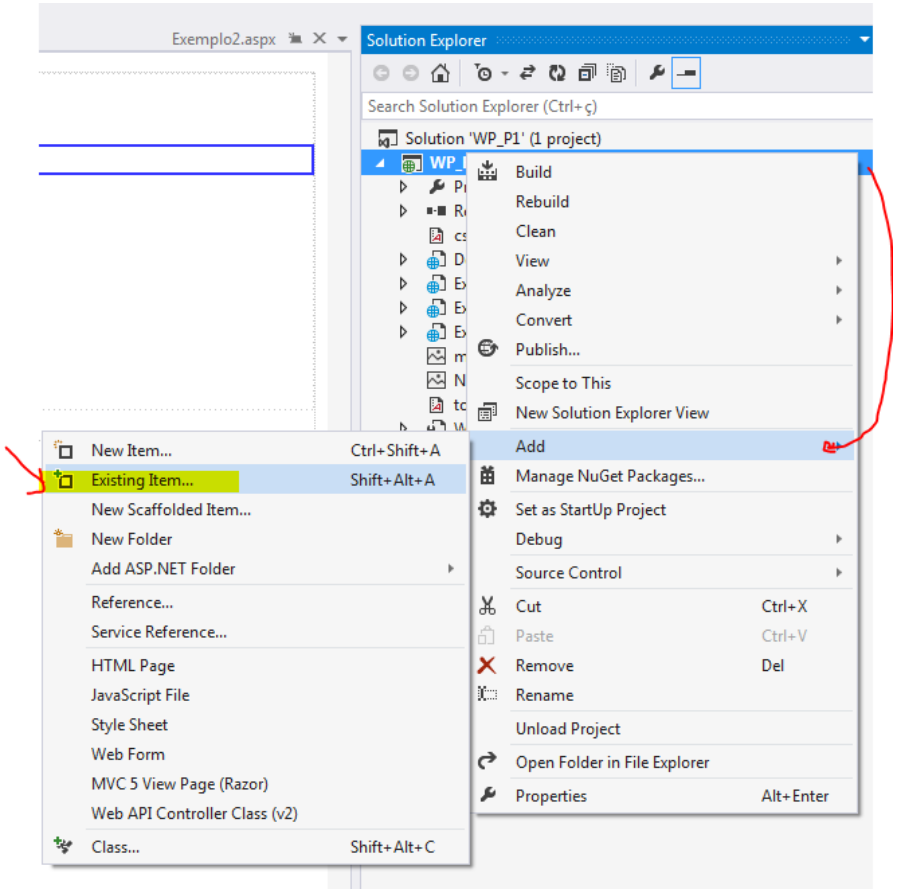
WEB FORMS



Image

WEB FORMS

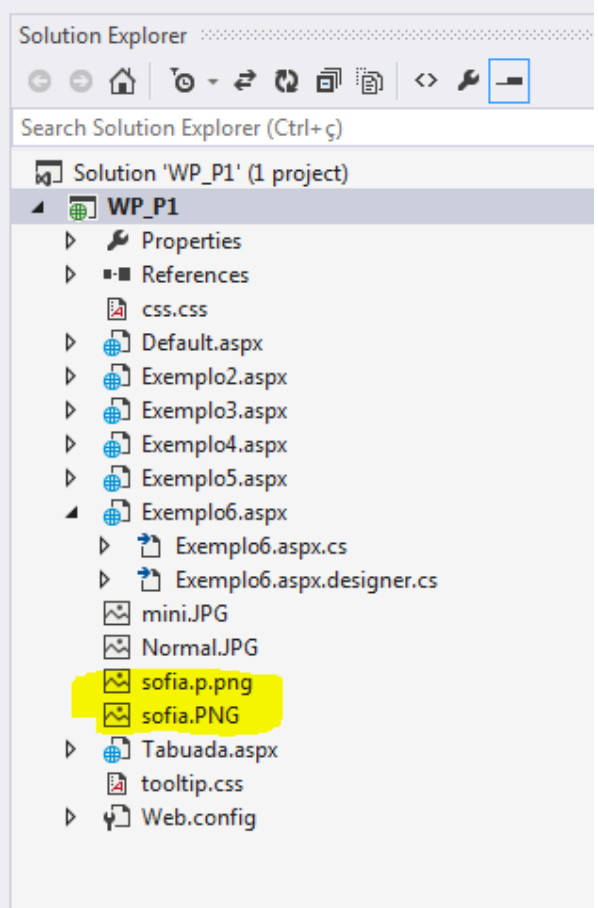




Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

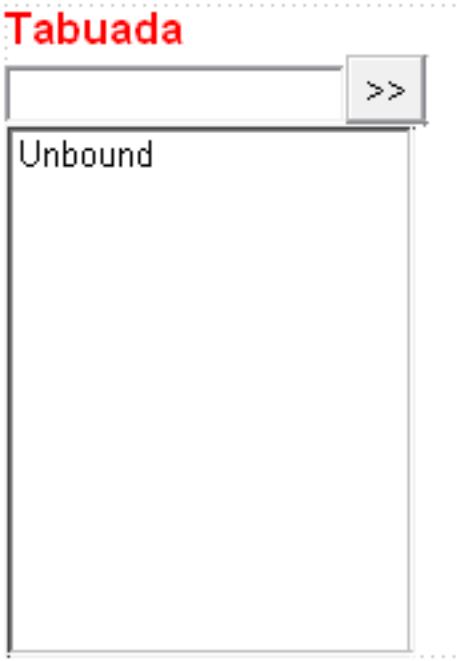
Image WEB FORMS



Exemplo6.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
{
    Img_principal.ImageUrl = "~/Imagens/sofia.PNG";
}

protected void Tb_1_Click(object sender, EventArgs e)
{
    Img_principal.ImageUrl = "~/Imagens/sofia.PNG";
}

protected void Bt_2_Click(object sender, EventArgs e)
{
    Img_principal.ImageUrl = "~/Imagens/sofia.p.PNG";
}

protected void Bt_3_Click(object sender, EventArgs e)
{
    Img_principal.Visible = false;
}

protected void Bt_4_Click(object sender, EventArgs e)
{
    Img_principal.Visible = true;
}
```

Exercício WEB FORMS



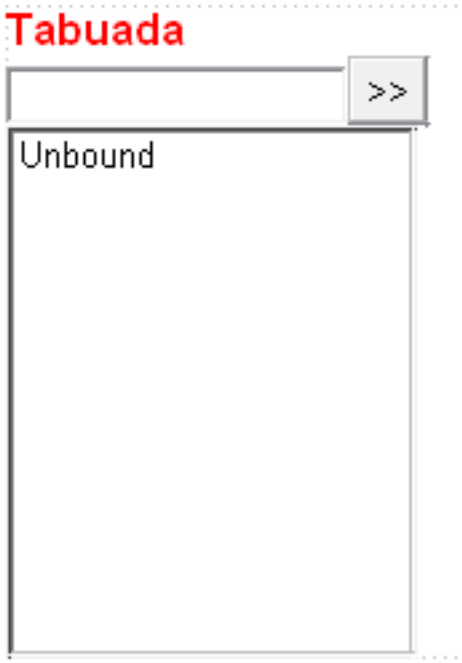
Exercício WEB FORMS

Dados 3 valores X, Y, Z, verificar se eles podem ser comprimentos de um triângulo e, se forem, verificar se é triângulo equilátero, isósceles e escaleno. Se não formarem, escrever uma mensagem. Dica: *Nenhum lado pode ser maior que a soma dos outros dois.*

"**Equilátero**. O que tem os três ângulos iguais."
"**Escaleno**. O que tem todos os ângulos e lados desiguais."
"**Isóscele**. O que tem dois lados e, portanto, dois ângulos iguais."

Dica: *Exercício resolvido em Console Application*

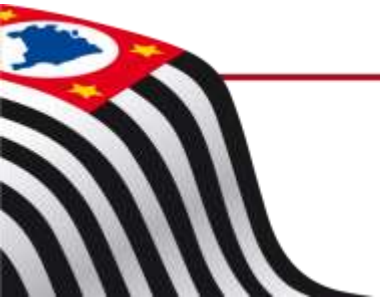
Exercício WEB FORMS




```
namespace WP_P1
{
    public partial class Tabuada : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Bt_Tabuada_Click(object sender, EventArgs e)
        {
            Lbx_Resultado.Items.Clear();
            for (int i = 1; i<= 10; i++)
            {
                Lbx_Resultado.Items.Add((int.Parse(Txt_Valor.Text) * i).ToString());
            }
        }
    }
}
```

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Capítulo 10: ASP.NET

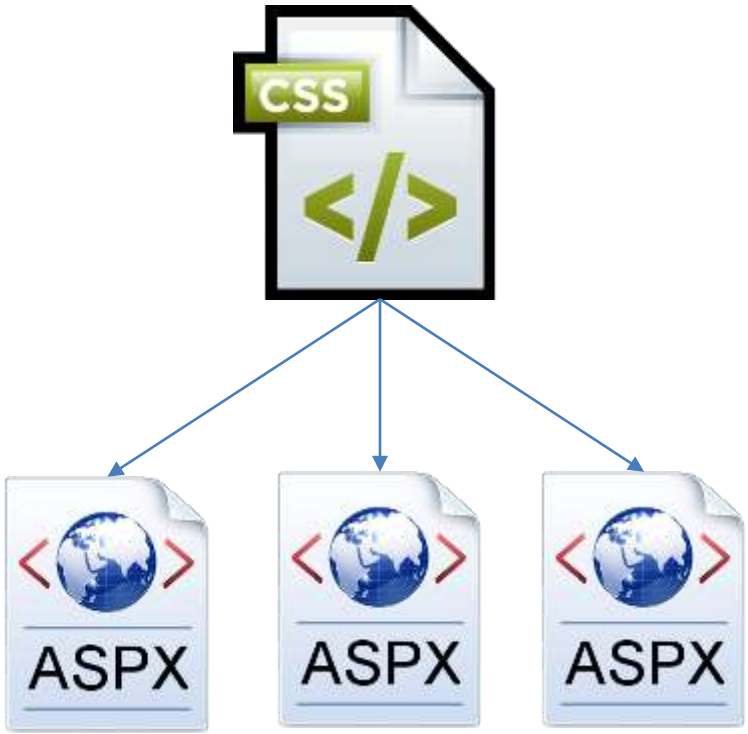


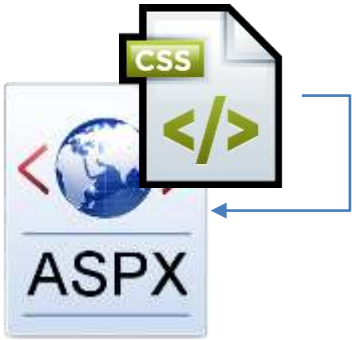
Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

WEB FORMS

- **Cascading Style Sheets (CSS)**
  - É uma linguagem de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento.

WEB FORMS

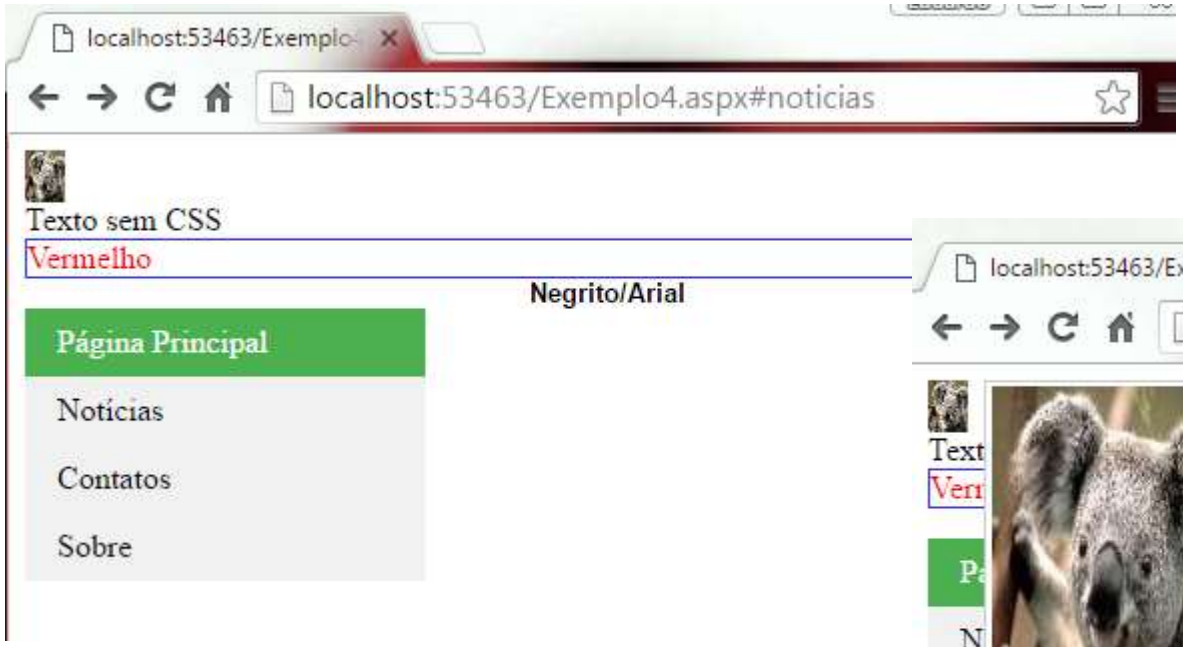


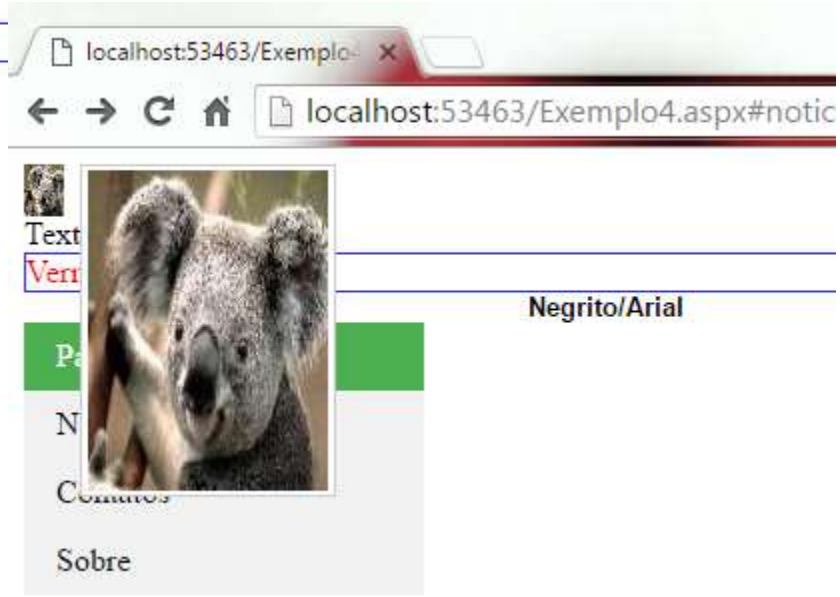


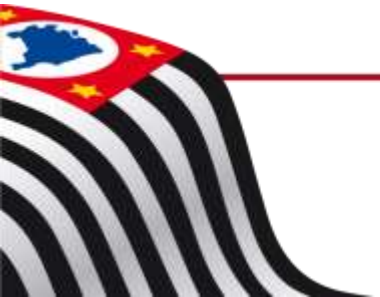
CSS

WEB FORMS

Microsoft
ASP.net
Microsoft
.net
Visual Studio
SQL Server
Microsoft
.net
AJAX


localhost:53463/Exemplo
localhost:53463/Exemplo4.aspx#noticias
Texto sem CSS
Vermelho
Negrito/Arial
Página Principal
Notícias
Contatos
Sobre

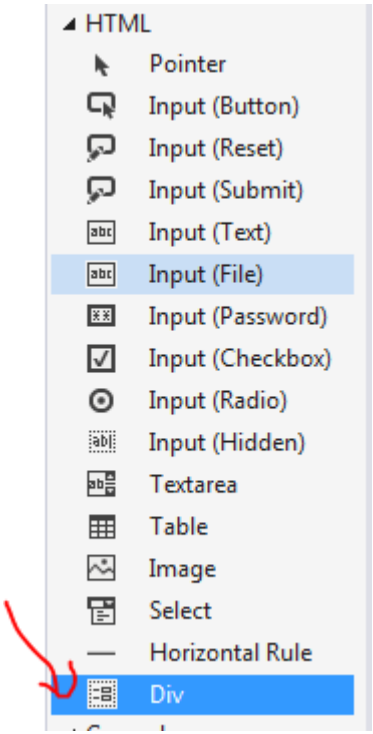

localhost:53463/Exemplo
localhost:53463/Exemplo4.aspx#notic
Negrito/Arial
Sobre

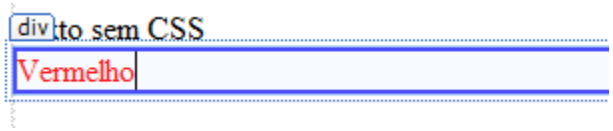


Centro
Paula Souza

GOVERNO DO ESTADO
SÃO PAULO

WEB FORMS

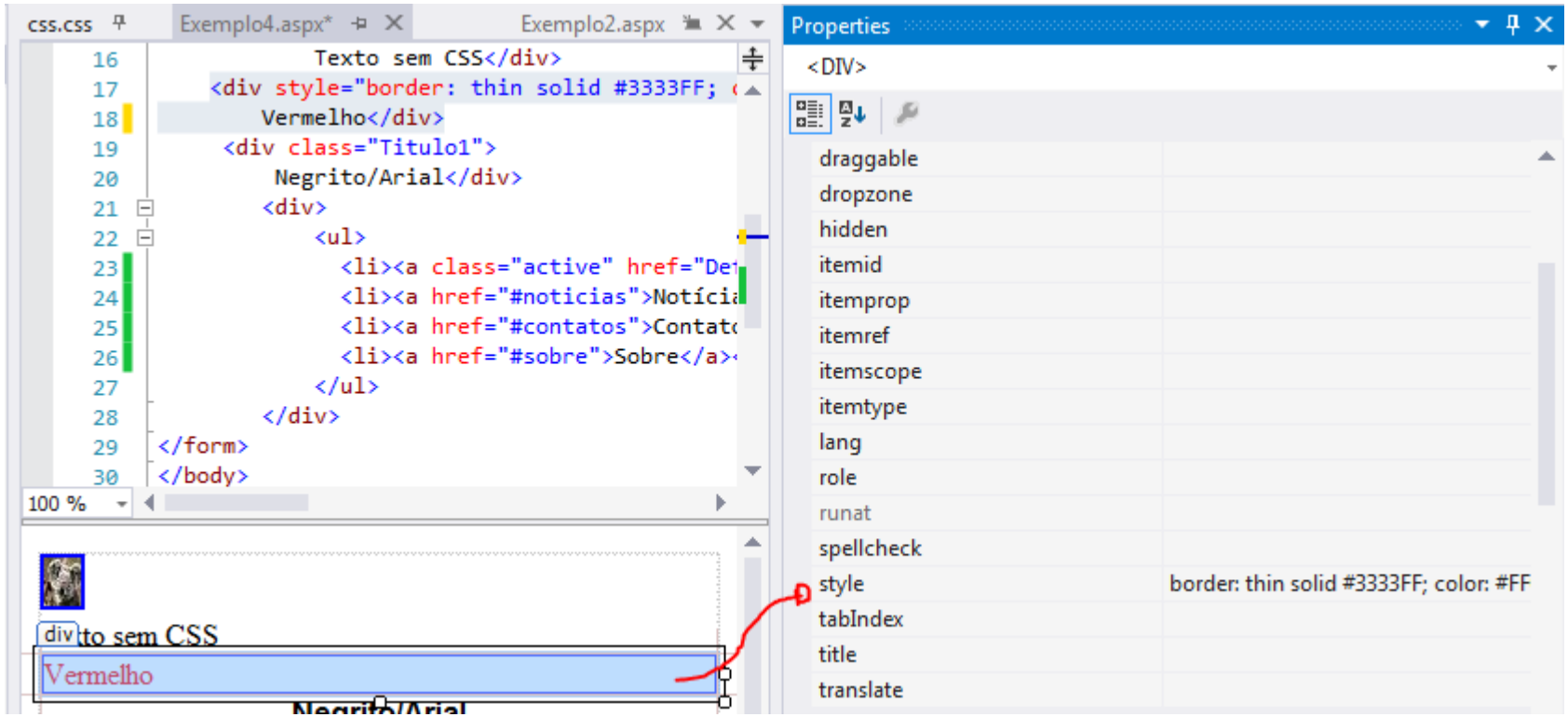

HTML
Pointer
Input (Button)
Input (Reset)
Input (Submit)
Input (Text)
Input (File)
Input (Password)
Input (Checkbox)
Input (Radio)
Input (Hidden)
Textarea
Table
Image
Select
Horizontal Rule
Div

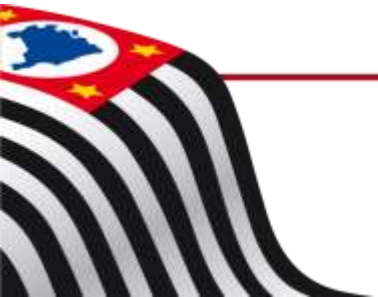

div
to sem CSS
Vermelho

CSS

WEB FORMS

Microsoft
ASP.net
Microsoft
.net
Visual Studio
SQL Server
Microsoft
.net
AJAX

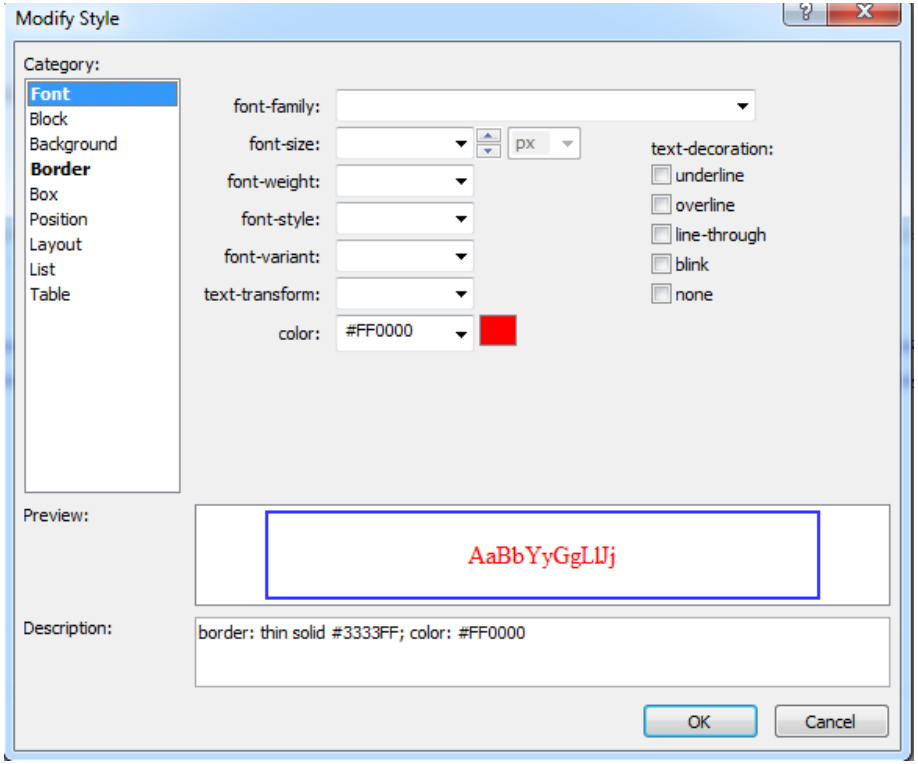

css.css
Exemplo4.aspx*
Exemplo2.aspx
16 Texto sem CSS
17 <div style="border: thin solid #3333FF;
18 Vermelho
19
20 Negrito/Arial
21
22 <ul>
23 <li><a class="active" href="Det
24 <li><a href="#noticias">Notícia
25 <li><a href="#contatos">Contato
26 <li><a href="#sobre">Sobre</a>
27 </ul>
28
29 </form>
30 </body>
100 %
div
to sem CSS
Vermelho
Properties
<DIV>
draggable
dropzone
hidden
itemid
itemprop
itemref
itemscope
itemtype
lang
role
runat
spellcheck
style
border: thin solid #3333FF; color: #FF
tabIndex
title
translate

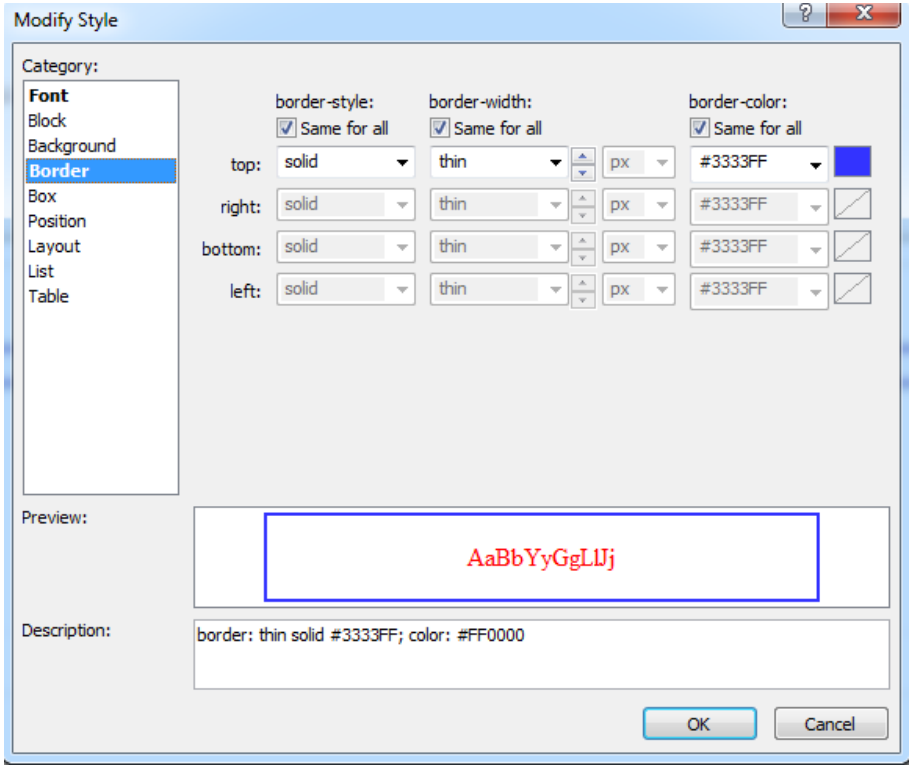


Centro
Paula Souza

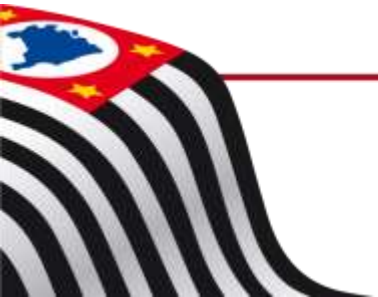
GOVERNO DO ESTADO
SÃO PAULO

WEB FORMS



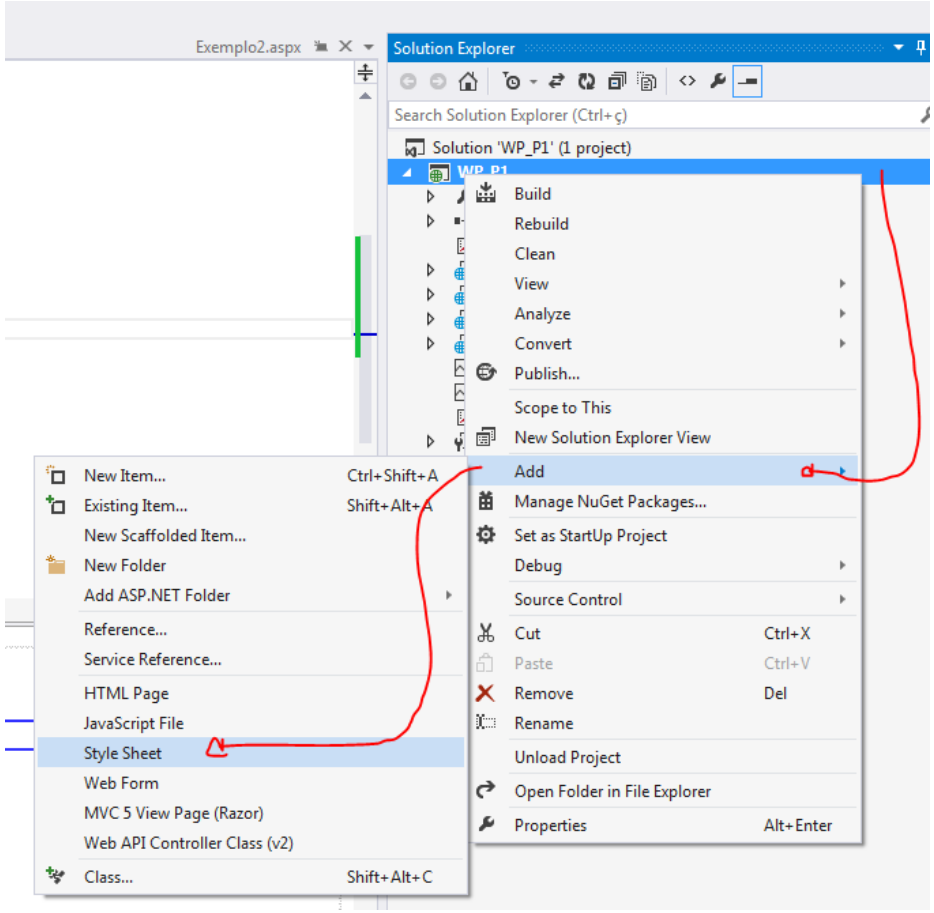


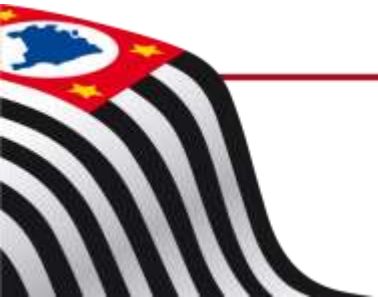

```
<div style="border: thin solid #3333FF; color: #FF0000; ">  Vermelho</div>
```



WEB FORMS

Negrito/Arial





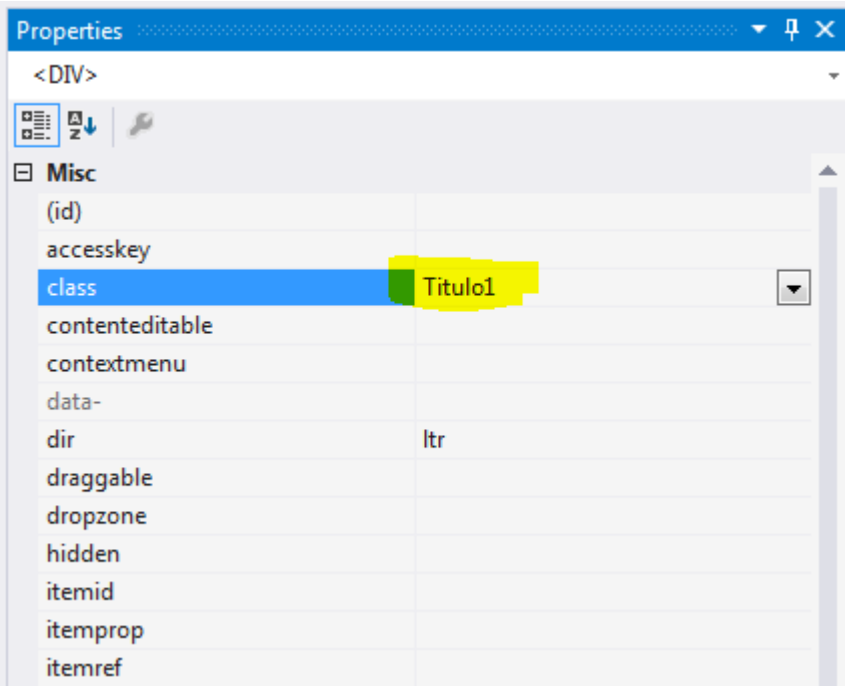
WEB FORMS

```
.Titulo1 {font-family: Arial; font-size: small; font-weight: bold; font-style: normal; text-align: center;}
```

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Exe
<link href="tooltip.css" rel="stylesheet" />
<link href="css.css" rel="stylesheet" />
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
```

WEB FORMS

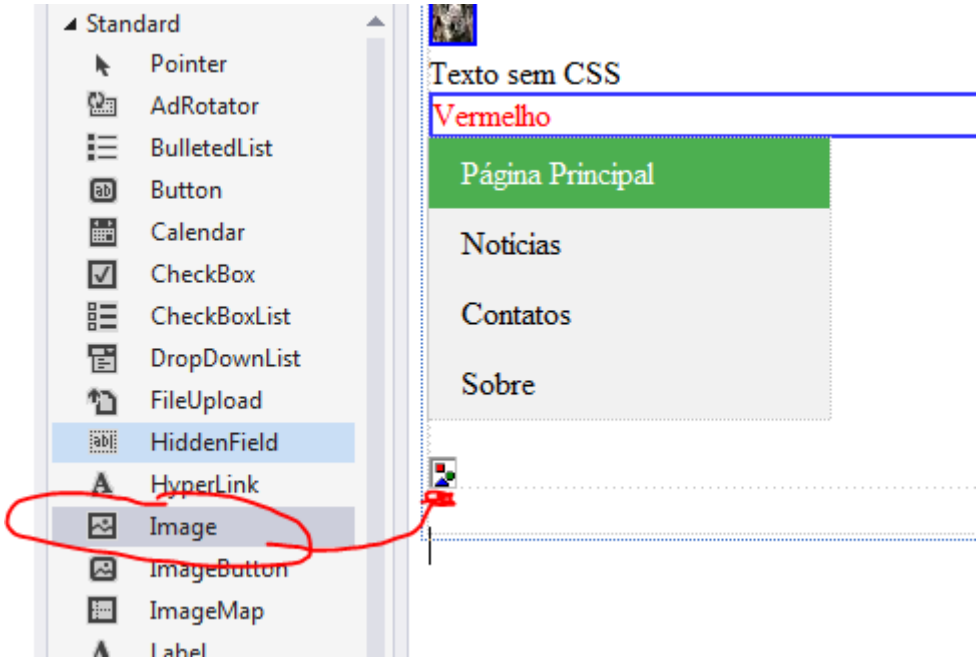
Negrito/Arial

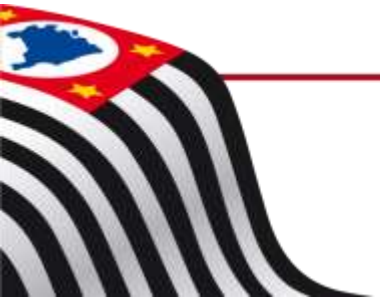


CSS

WEB FORMS

Microsoft
ASP.net
Microsoft
.net
Visual Studio
SQL Server
Microsoft
.net
AJAX

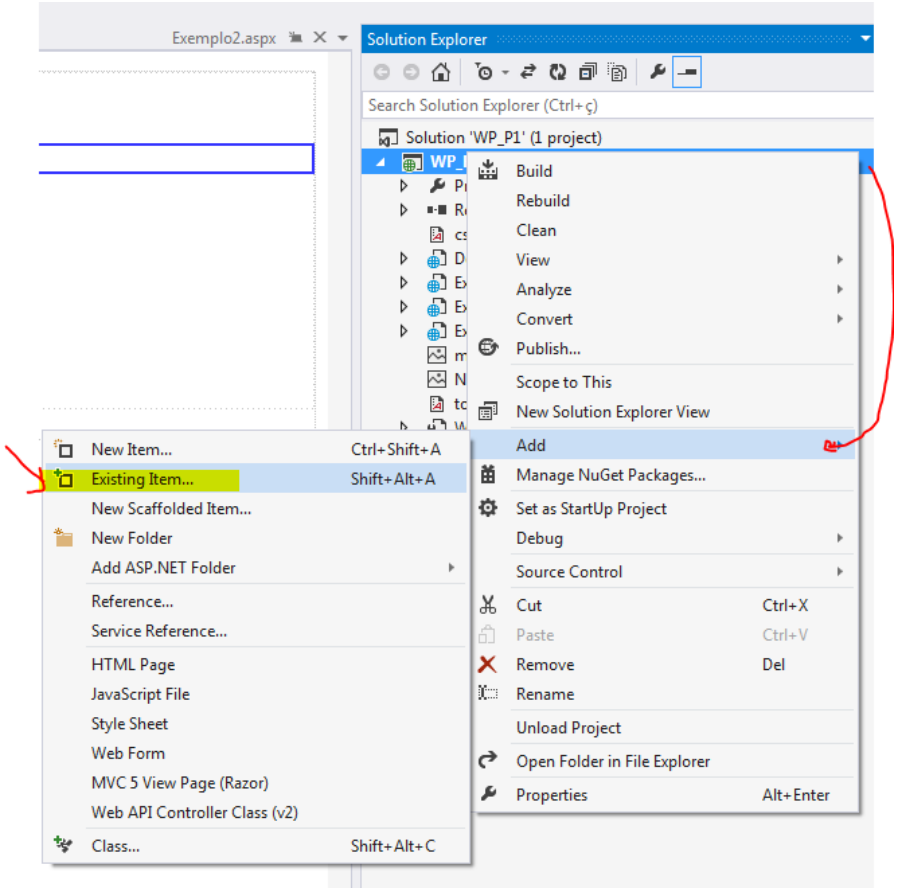

Standard
Pointer
AdRotator
BulletedList
Button
Calendar
CheckBox
CheckBoxList
DropDownList
FileUpload
HiddenField
HyperLink
Image
ImageButton
ImageMap
Label
Texto sem CSS
Vermelho
Página Principal
Notícias
Contatos
Sobre

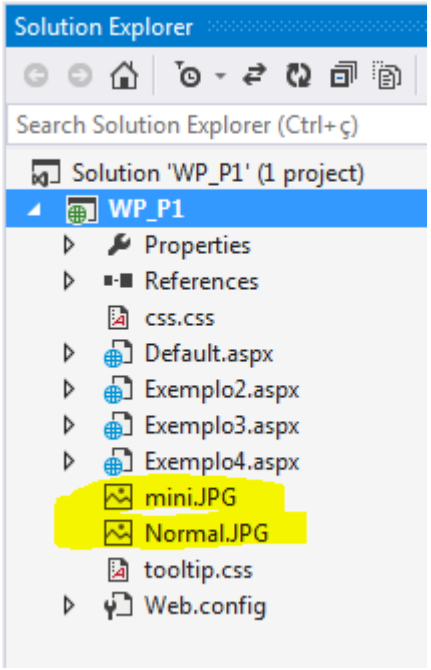


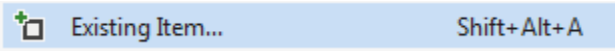
Centro
Paula Souza

GOVERNO DO ESTADO
SÃO PAULO

WEB FORMS







Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

WEB FORMS

ss* tooltip.css Exemplo4.aspx*

```
CSbody {font-family:Georgia, "Times New Roman", Times, serif}
a:hover {background:#ffffff; text-decoration:none;}
a.tooltip span {display:none; padding:2px 3px; margin-left:8px;}
a.tooltip:hover span{display:inline; position:absolute; background:#ffffff; border:1px solid #cccccc; color:#6c6c6c;}
.cssCorpo { font-family: Arial, Helvetica, sans-serif;  font-size: 11px;font-style: normal;}
.cssTitulo {font-family: Arial, Helvetica, sans-serif;font-size: 11px;font-style: inherit;}
```

WEB FORMS

asp:Image#Image1

Texto sem CSS

Vermelho

Página Principal

Notícias

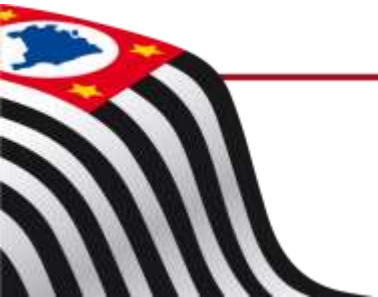
Contatos

Sobre

Negrito/Arial

Image1 System.Web.UI.WebControls.Image

Accessibility	
AccessKey	
DescriptionUrl	
GenerateEmptyAlternateText	False
TabIndex	0
Appearance	
AlternateText	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
CssClass	
ForeColor	
ImageUrl	~/mini.JPG
Behavior	
ClientIDMode	Inherit
EnableTheming	True
EnableViewState	True

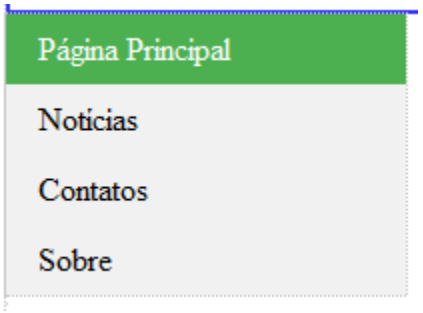


WEB FORMS

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Exemplo
<link href="tooltip.css" rel="stylesheet" />
<link href="css.css" rel="stylesheet" />
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body style="height: 273px">
    <form id="form1" runat="server">
    <div>
        <a href="#" class="tooltip"><asp:Image ID="Image1" runat=
    </div>
        <div>
            Texto sem CSS</div>
```

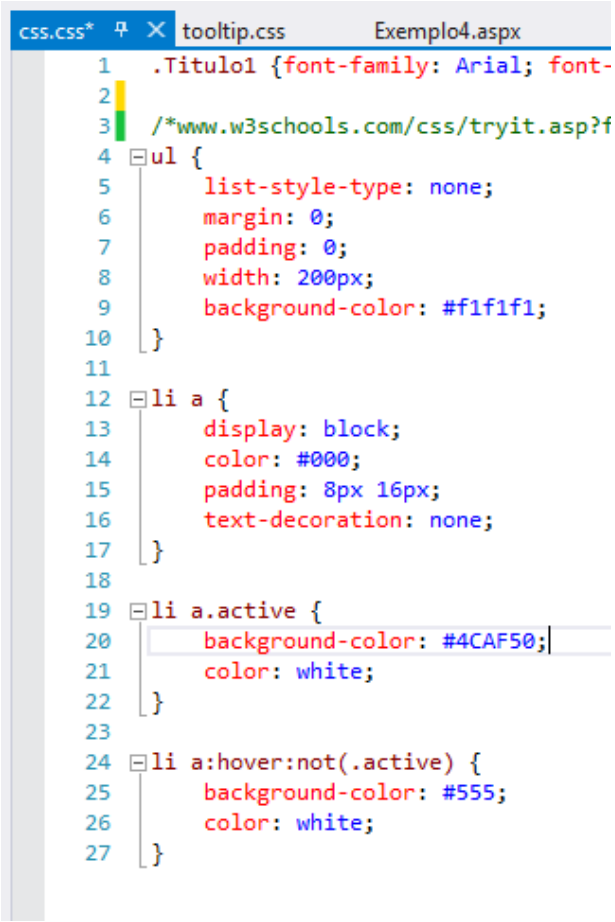
```
<div>
    <a href="#" class="tooltip"><asp:Image ID="Image1" runat="server"
        ImageUrl="~/mini.JPG" /><span><img src="Normal.JPG" /></span></a>
</div>
```

WEB FORMS



www.w3schools.com/css/tryit.asp?filename=trycss_navbar_vertical_active

WEB FORMS




```
.Titulo1 {font-family: Arial; font-

/*www.w3schools.com/css/tryit.asp?f
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    width: 200px;
    background-color: #f1f1f1;
}

li a {
    display: block;
    color: #000;
    padding: 8px 16px;
    text-decoration: none;
}

li a.active {
    background-color: #4CAF50;
    color: white;
}

li a:hover:not(.active) {
    background-color: #555;
    color: white;
}
```

WEB FORMS

```
7   <head runat="server">
8       <title></title>
9   </head>
10  <body style="height: 273px">
11      <form id="form1" runat="server">
12      <div>
13          <a href="#" class="tooltip"><asp:Image ID="Image1" runat="server" ImageUrl="~/mi
14      </div>
15          <div>
16              Texto sem CSS</div>
17
18          <div style="border: thin solid #3333FF; color: #FF0000; ">  Vermelho</div>
19
20          <div>
21              <ul>
22                <li><a class="active" href="Default.aspx">Página Principal</a></li>
23                <li><a href="#noticias">Notícias</a></li>
24                <li><a href="#contatos">Contatos</a></li>
25                <li><a href="#sobre">Sobre</a></li>
26              </ul>
27          </div>
28
29          <div class="Titulo1">Negrito/Arial</div>
30  </form>
31  </body>
```

Capítulo 11: ASP.NET

Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

WEB FORMS

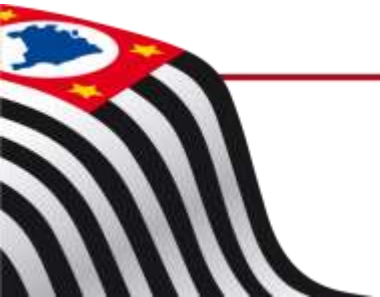
body

Nome		RequiredFieldValidator
Senha		
Confirma		CompareValidator
Salário		RangeValidator
E-mail		RegularExpressionValidator
Login		CustomValidator
Salvar		

- Error message 1.
- Error message 2.

Toolbox

- Validation
  - Pointer
  - CompareValidator
  - CustomValidator
  - RangeValidator
  - RegularExpressionValidator
  - RequiredFieldValidator
  - ValidationSummary



WEB FORMS

```
'Campo Requerido" ControlToValidate="Txt_Nome">Campo Nome Vazio</asp:RequiredFieldValidator>
```

asp:RequiredField...#RequiredField...

Campo Nome Vazio

Senha não confere

Salário Inválido

E-mail Inválido

Usuário Incorreto

RequiredFieldValidator1 System.Web.UI.WebControls.RequiredFieldValidatc

(Expressions)	
(ID)	**RequiredFieldValidator1**
AccessKey	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
ClientIDMode	Inherit
ControlToValidate	**Txt_Nome**
CssClass	
Display	Static
EnableClientScript	True
Enabled	True
EnableTheming	True
EnableViewState	True
ErrorMessage	**Campo Requerido**
Font	
ForeColor	
Height	
InitialValue	
SetFocusOnError	False
SkinID	
TabIndex	0
Text	**Campo Nome Vazio**
ToolTip	
ValidateRequestMode	Inherit
ValidationGroup	
ViewStateMode	Inherit
Visible	True
Width	

Validation

WEB FORMS

Microsoft
ASP.net
Microsoft
.net
Visual Studio
SQL Server
Microsoft
.net
AJAX

localhost:56348/Default.
localhost:56348/Default.aspx
Nome
Senha
Confirma
Salário
E-mail
Login
Salvar
Campo Nome Vazio
Campo Requerido

WEB FORMS

```
e="Txt_Senha" ControlToValidate="Txt_Confirma">Senha não confere</asp:CompareValidator>

'Txt_Salario" MaximumValue="9999" MinimumValue="1">Salário Inválido</asp:RangeValidator>
```

Campo Nome Vazio

asp:CompareValidator#CompareValida...

Senha não confere

Salário Inválido

E-mail Inválido

Usuário Incorreto

Properties

CompareValidator1 System.Web.UI.WebControls.CompareValidator

(Expressions)	
(ID)	**CompareValidator1**
AccessKey	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
ClientIDMode	Inherit
ControlToCompare	**Txt_Senha**
ControlToValidate	**Txt_Confirma**
CssClass	
CultureInvariantValues	False
Display	Static
EnableClientScript	True
Enabled	True
EnableTheming	True
EnableViewState	True
ErrorMessage	**Senha não confere**
Font	
ForeColor	
Height	
Operator	Equal
SetFocusOnError	False
SkinID	
TabIndex	0
Text	**Senha não confere**
ToolTip	
Type	String
ValidateRequestMode	Inherit
ValidationGroup	
ValueToCompare	
ViewStateMode	Inherit
Visible	True
Width	

Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

WEB FORMS

```
are="Txt_Senha" ControlToValidate="Txt_Confirma">Senha não confere</asp:CompareValidator>

="Txt_Salario" MaximumValue="9999" MinimumValue="1">Salário Inválido</asp:RangeValidator>
```

Campo Nome Vazio

asp:RangeValidator#RangeValidator1

Salário Inválido

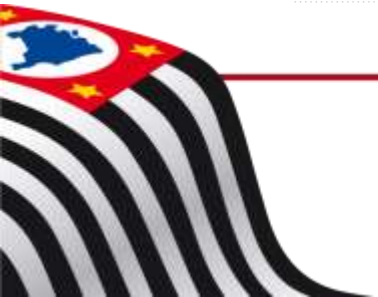
E-mail Inválido

Usuário Incorreto

Properties

RangeValidator1 System.Web.UI.WebControls.RangeValidator

(Expressions)	
(ID)	**RangeValidator1**
AccessKey	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
ClientIDMode	Inherit
ControlToValidate	**Txt_Salario**
CssClass	
CultureInvariantValues	False
Display	Static
EnableClientScript	True
Enabled	True
EnableTheming	True
EnableViewState	True
ErrorMessage	**Salário Incorreto**
Font	
ForeColor	
Height	
MaximumValue	**9999**
MinimumValue	**1**
SetFocusOnError	False
SkinID	
TabIndex	0
Text	**Salário Inválido**
ToolTip	
Type	String
ValidateRequestMode	Inherit
ValidationGroup	
ViewStateMode	Inherit
Visible	True
Width	



WEB FORMS

Validation

```
p:RangeValidator>

[-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*">E-mail Inválido</asp:RegularExpressionValidator>

ator>
```

Campo Nome Vazio

Senha não confere

asp:RegularExpres...#RegularExpres...

E-mail Inválido

Usuário Incorreto

Properties

RegularExpressionValidator1 System.Web.UI.WebControls.RegularExpressionVa

(Expressions)	
(ID)	RegularExpressionValidator1
AccessKey	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
ClientIDMode	Inherit
ControlToValidate	Txt_Email
CssClass	
Display	Static
EnableClientScript	True
Enabled	True
EnableTheming	True
EnableViewState	True
ErrorMessage	E-mail Digitado é inválido
Font	
ForeColor	
Height	
SetFocusOnError	False
SkinID	
TabIndex	0
Text	E-mail Inválido
ToolTip	
ValidateRequestMode	Inherit
ValidationExpression	\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\v
ValidationGroup	
ViewStateMode	Inherit
Visible	True
Width	

Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

WEB FORMS

Validation

p:RangeValidator>

[-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]

ator>

Campo Nome Vazio

Senha não confere

asp:RegularExpres...#RegularExpres...

E-mail Inválido

Usuário Incorreto

Regular Expression Editor

Standard expressions:

- French phone number
- French postal code
- German phone number
- German postal code
- Internet e-mail address

Validation expression:

\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*

OK | Cancel

Properties

RegularExpressionValidator1 System.Web.UI.WebControls.RegularExpressionVa

(Expressions)	
(ID)	RegularExpressionValidator1
AccessKey	
BackColor	
BorderColor	
	NotSet
de	Inherit
alidate	Txt_Email
	Static
tScript	True
	True
ning	True
State	True
ge	E-mail Digitado é inválido
Error	False
	0
	E-mail Inválido
ToolTip	
ValidateRequestMode	Inherit
ValidationExpression	\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w
ValidationGroup	
ViewStateMode	Inherit
Visible	True
Width	

Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

WEB FORMS

```
ControlToValidate="Txt_Salario" MaximumValue="9999" MinimumValue="1">Salário Inválido</asp:Rang

sage="E-mail Digitado é inválido" ControlToValidate="Txt_Email" ValidationExpression="\w+([-+.'

ere" OnServerValidate="CustomValidator1_ServerValidate">Usuário Incorreto</asp:CustomValidator>
```

Campo Nome Vazio

Senha não confere

Salário Inválido

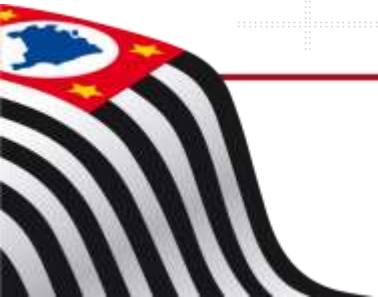
asp:CustomValidator#CustomValidator1

Usuário Incorreto

Properties

CustomValidator1 System.Web.UI.WebControls.CustomValidator

(Expressions)	
(ID)	**CustomValidator1**
AccessKey	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
ClientIDMode	Inherit
ClientValidationFunction	
ControlToValidate	
CssClass	
Display	Static
EnableClientScript	True
Enabled	True
EnableTheming	True
EnableViewState	True
ErrorMessage	**Usuário não confere**
Font	
ForeColor	
Height	
SetFocusOnError	False
SkinID	
TabIndex	0
Text	**Usuário Incorreto**
ToolTip	
ValidateEmptyText	False
ValidateRequestMode	Inherit
ValidationGroup	
ViewStateMode	Inherit
Visible	True
Width	



WEB FORMS

```
ControlToValidate="Txt_Salario" MaximumValue="9999" MinimumValue="1">Salário Inválido</asp:Rang
```

Properties

CustomValidator1 System.Web.UI.WebControls.CustomValidator

Web.config | Default.aspx.cs | Default.aspx

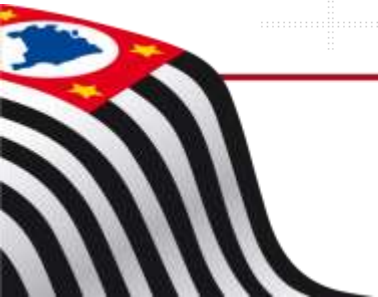
_Default | CustomValidator1_ServerV

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.UI;
6  using System.Web.UI.WebControls;
7
8  public partial class _Default : System.Web.UI.Page
9  {
10     protected void Page_Load(object sender, EventArgs e)
11     {
12
13     }
14     protected void CustomValidator1_ServerValidate(object source, ServerValidateEventArgs args)
15     {
16         args.IsValid = (Txt_Login.Text == "eduardo");
17     }
18 }
```

asp:CustomValidator#CustomValidator1

Usuário Incorreto

ValidateRequestMode	Inherit
ValidationGroup	
ViewStateMode	Inherit
Visible	True
Width	



WEB FORMS

Erro de Servidor no Aplicativo '/'.

WebForms UnobtrusiveValidationMode requer um ScriptResourceMapping para 'jquery'. Adicione um jquery nomeado de ScriptResourceMapping (diferencia maiúsculas de minúsculas).

WEB FORMS

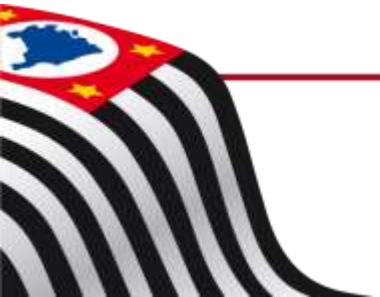
Web.config | Default.aspx.cs | Default.aspx

```xml
<?xml version="1.0"?>

<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
  -->

<configuration>

    <system.web>
      <compilation debug="true" targetFramework="4.5" />
      <httpRuntime targetFramework="4.5" />
    </system.web>

  <appSettings>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>
  </appSettings>


</configuration>
```

RESOLVIDO

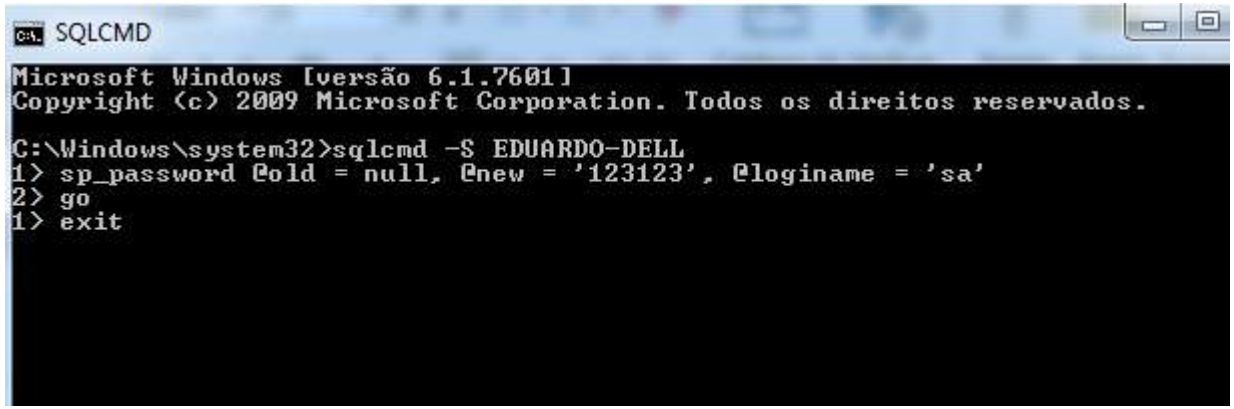
```xml
<appSettings>
   <add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>
 </appSettings>
```

Capítulo 12a: ASP.NET

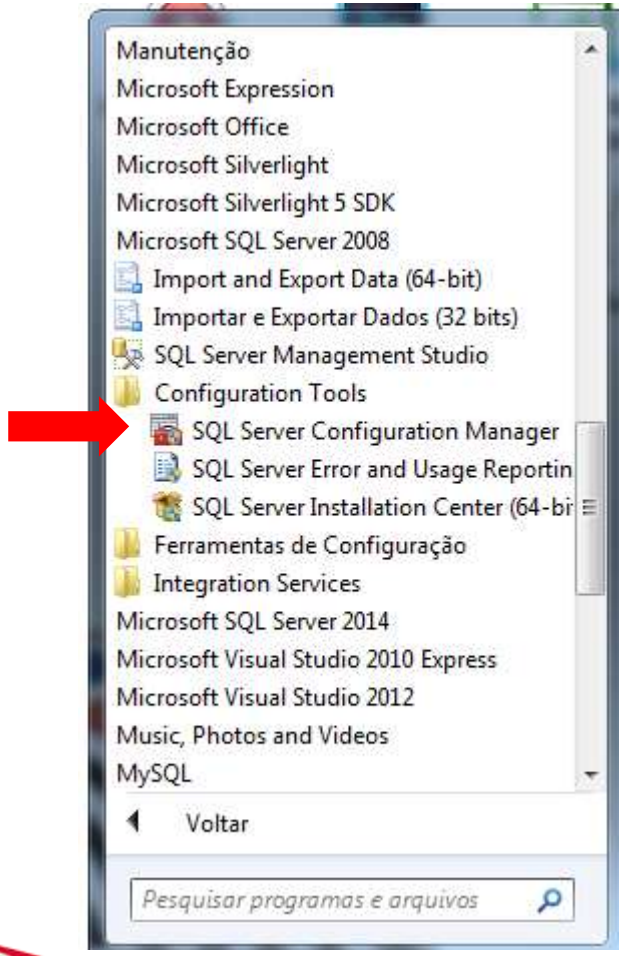


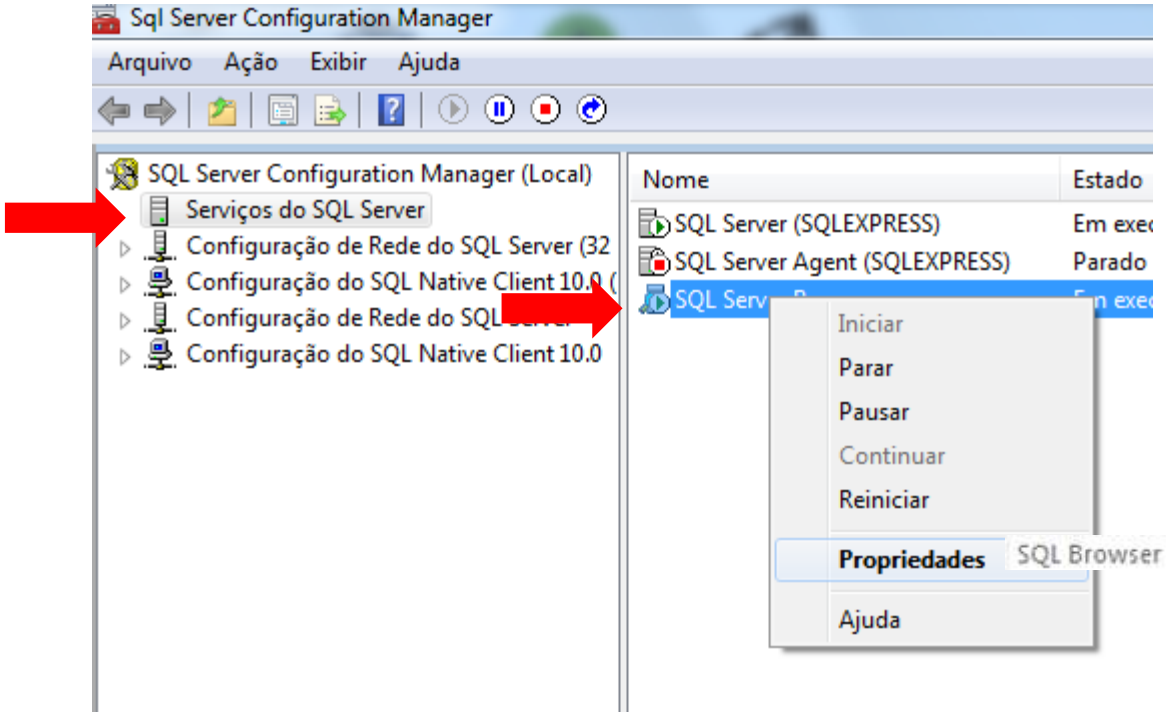
Configurando Usuário SA

```
sqlcmd -S EDUARDO-DELL
sp_password @old = null, @new = '123123', @loginame = 'sa'
go
exit
```

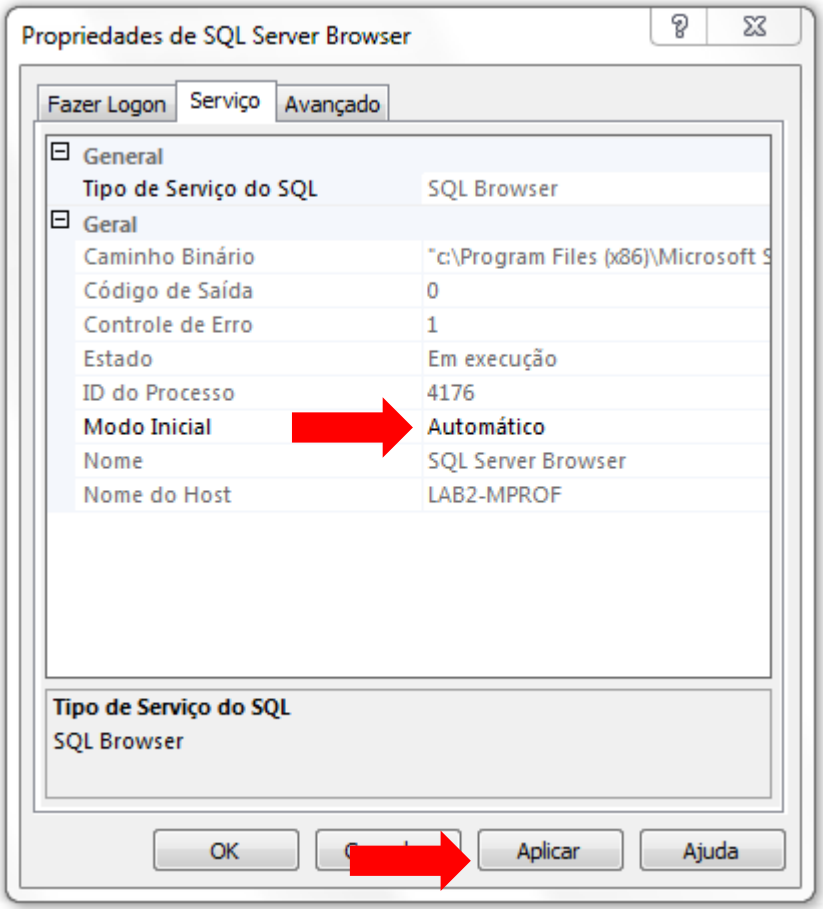


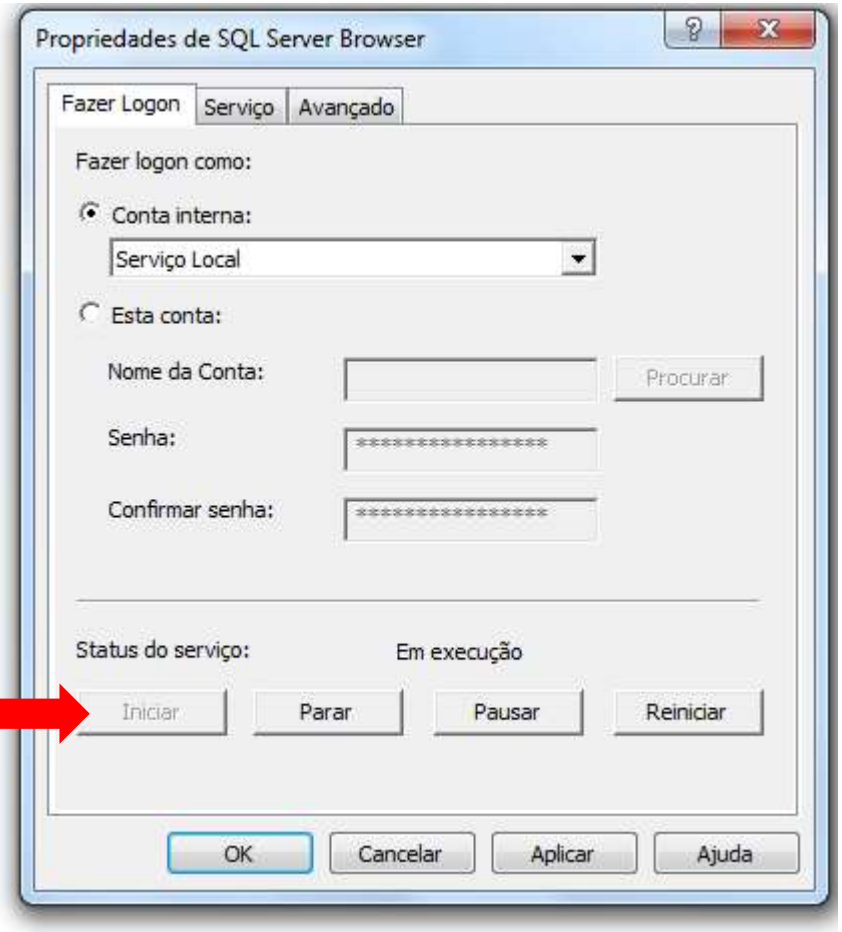
Configurando Usuário SA





Configurando Usuário SA

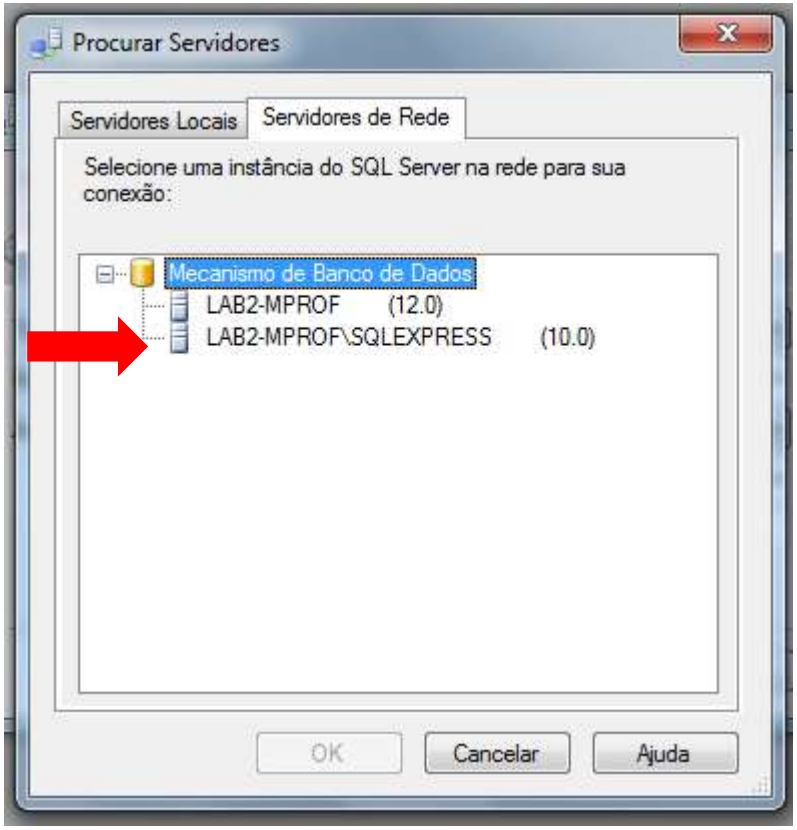




Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

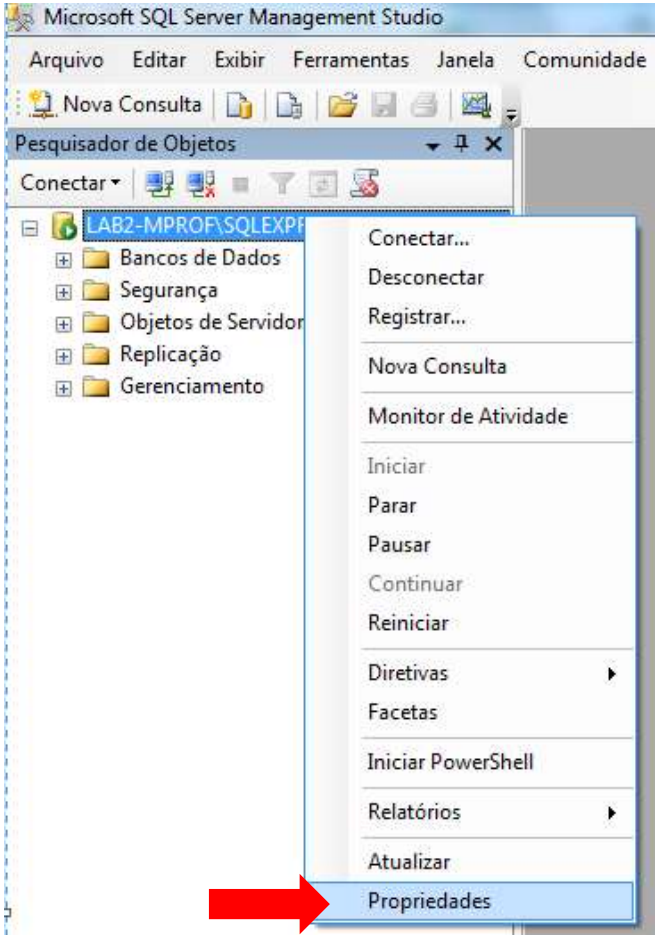
Configurando Usuário SA



Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

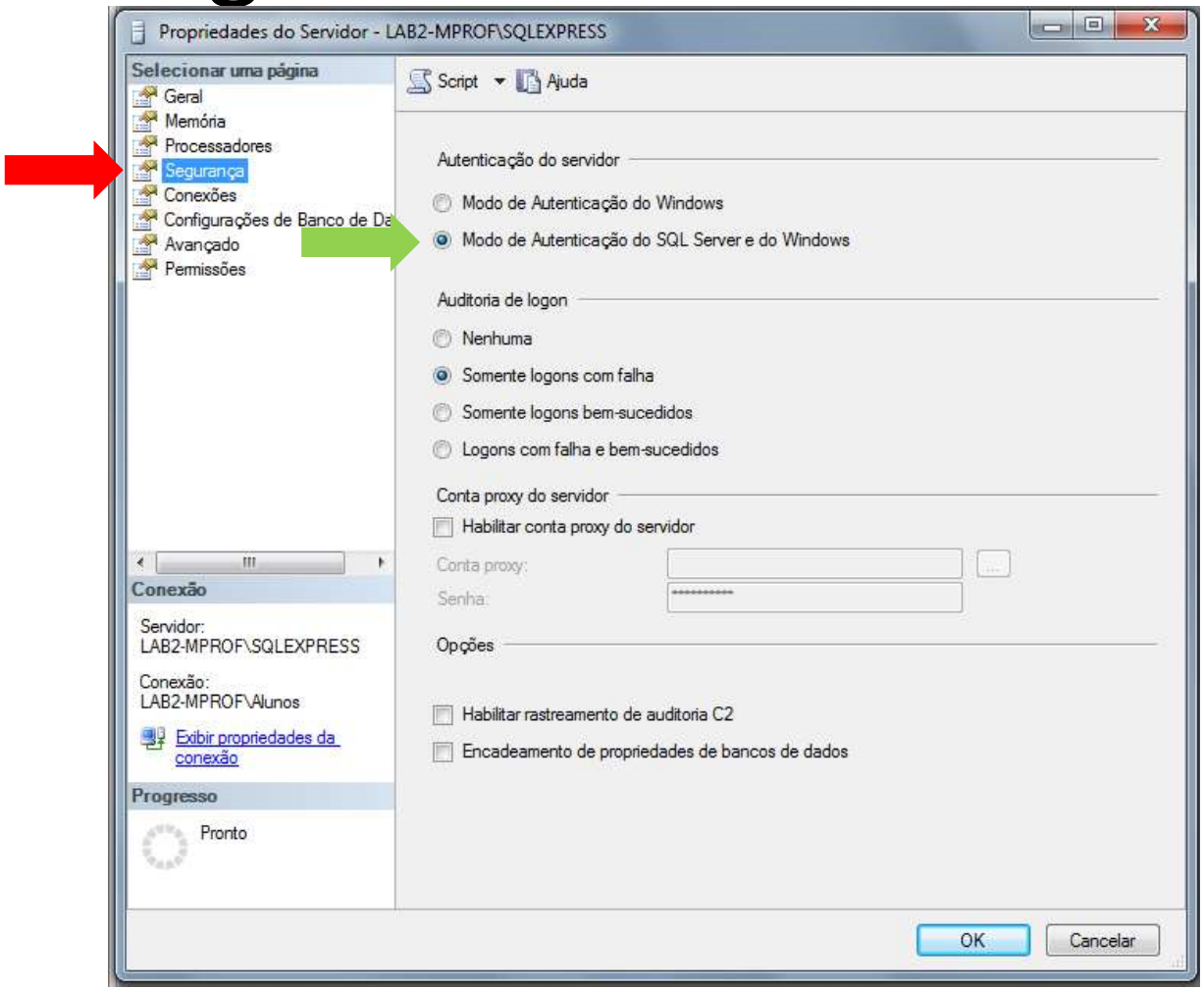
Configurando Usuário SA



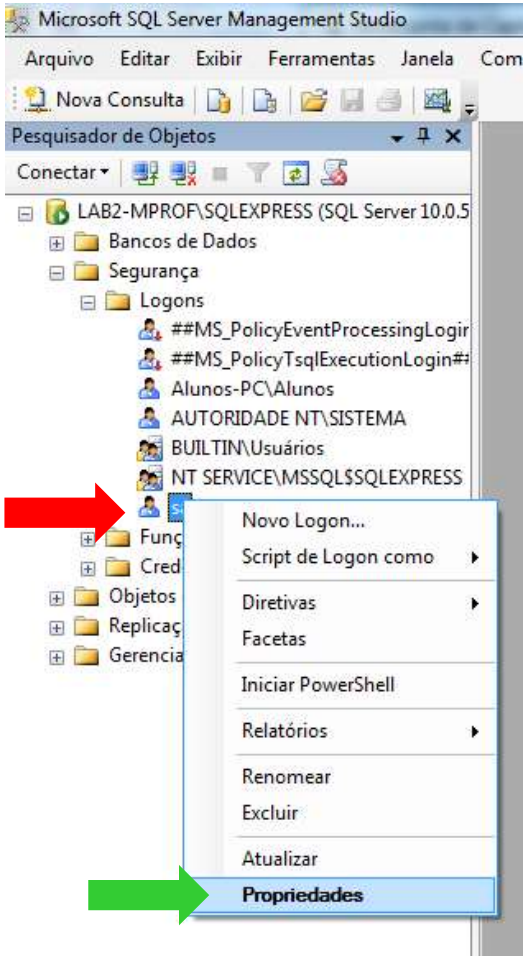
Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

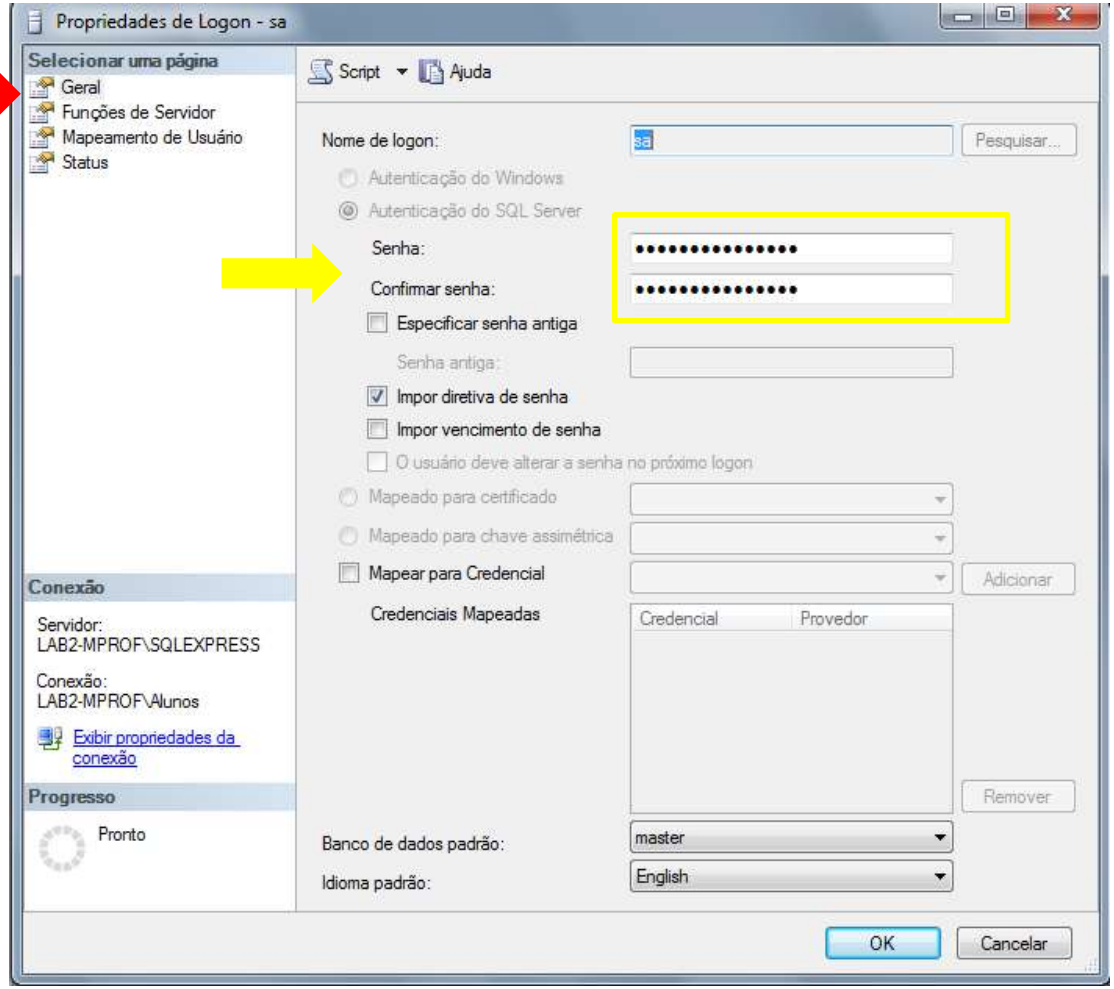
Configurando Usuário SA



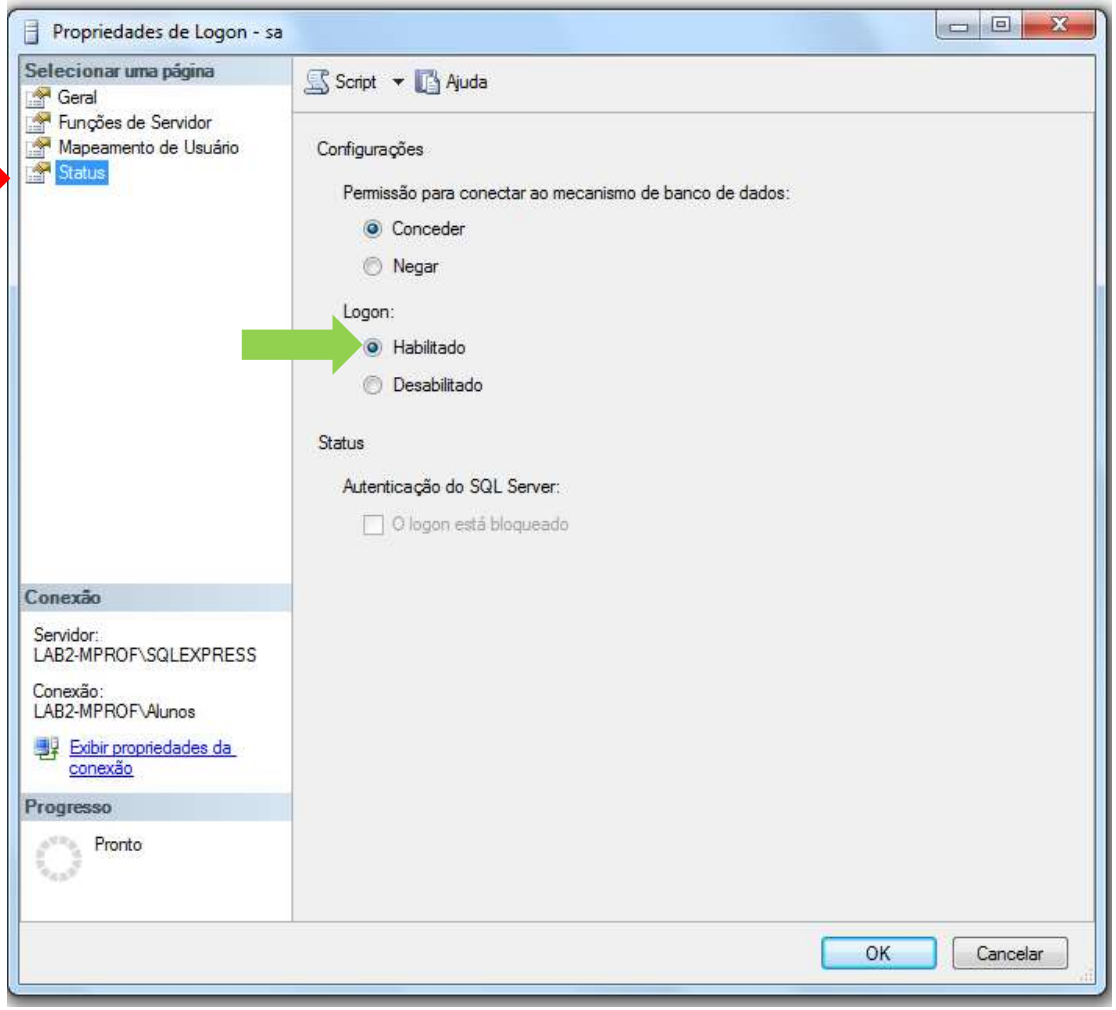
Configurando Usuário SA



Configurando Usuário SA

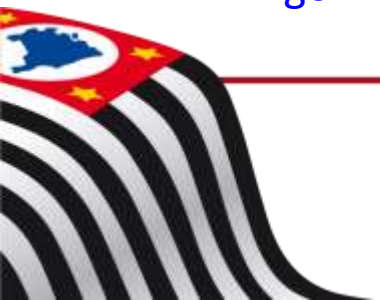


Configurando Usuário SA

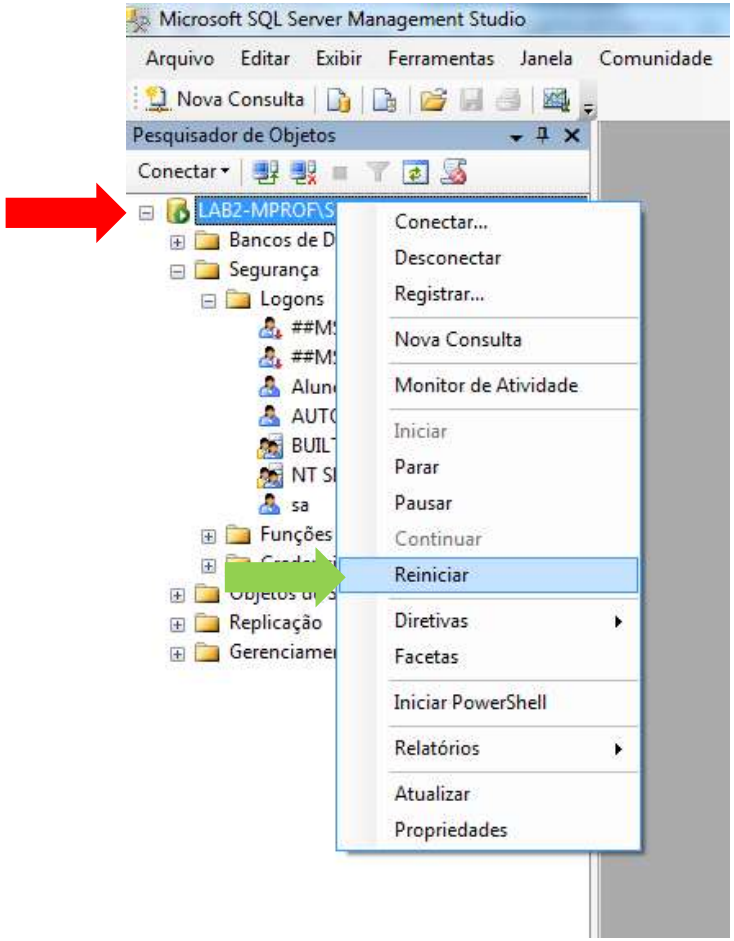


Configurando Usuário SA

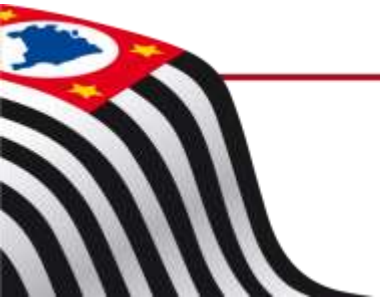
```
use [master]
go

--altera o modo de autenticação do banco de dados para "Mixed"
exec xp_instance_regwrite 'hkey_local_machine',
                    'software\microsoft\mssqlserver\mssqlserver',
                    'loginmode', reg_dword, 2
go

--Altera a senha do SA
alter login sa with password='123123'
go

--Habilita o usuário sa
alter login sa enable
go
```



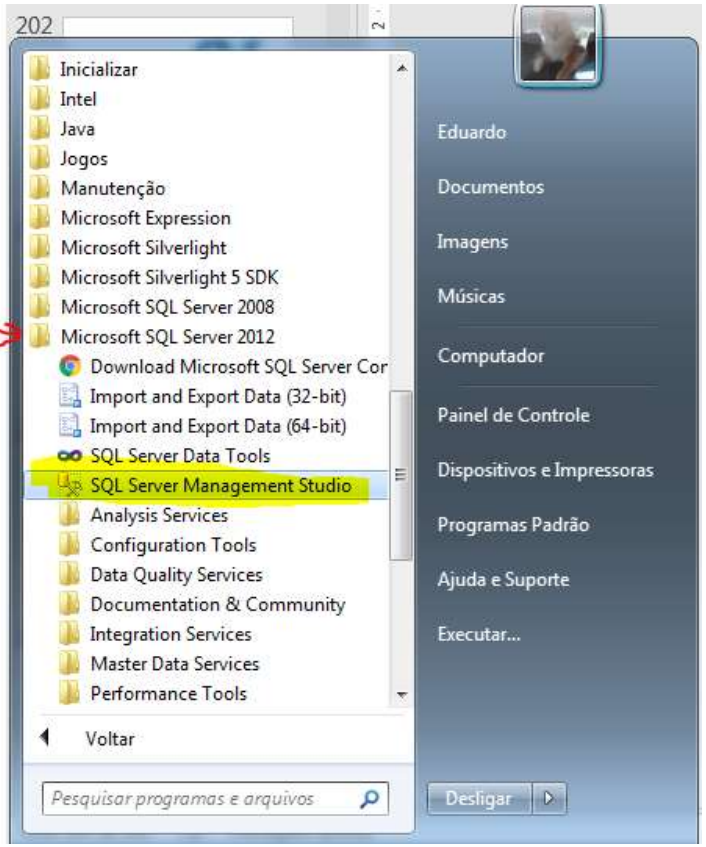
Configurando Usuário SA



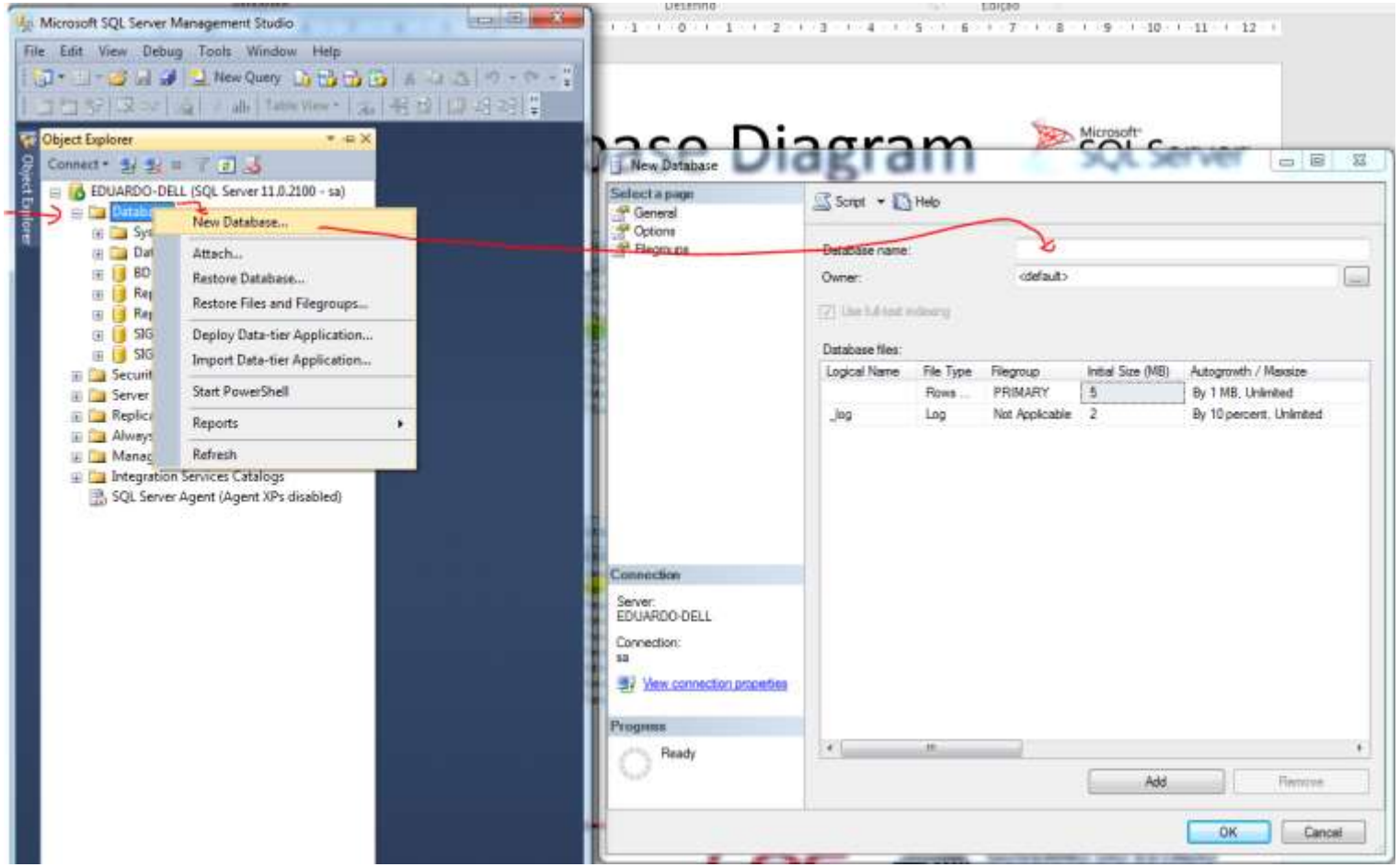
Capítulo 12b: ASP.NET



Database Diagram



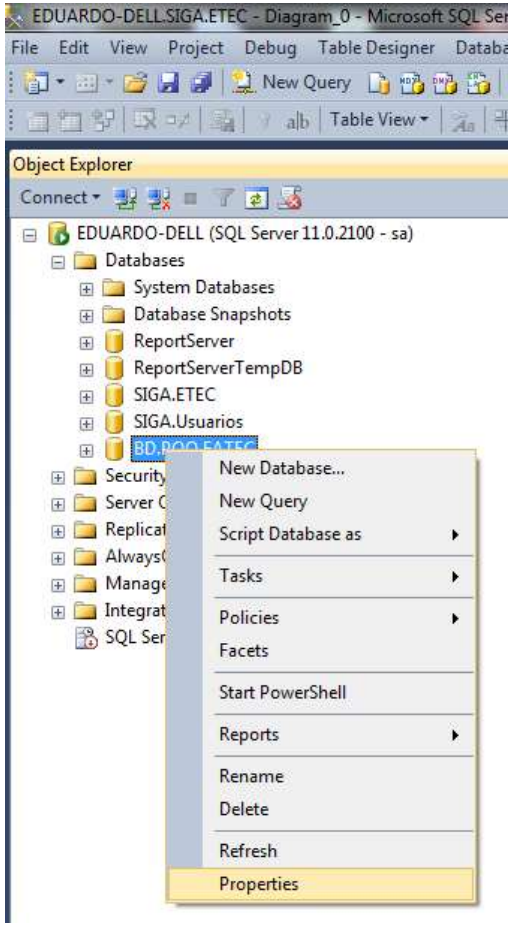
Database Diagram

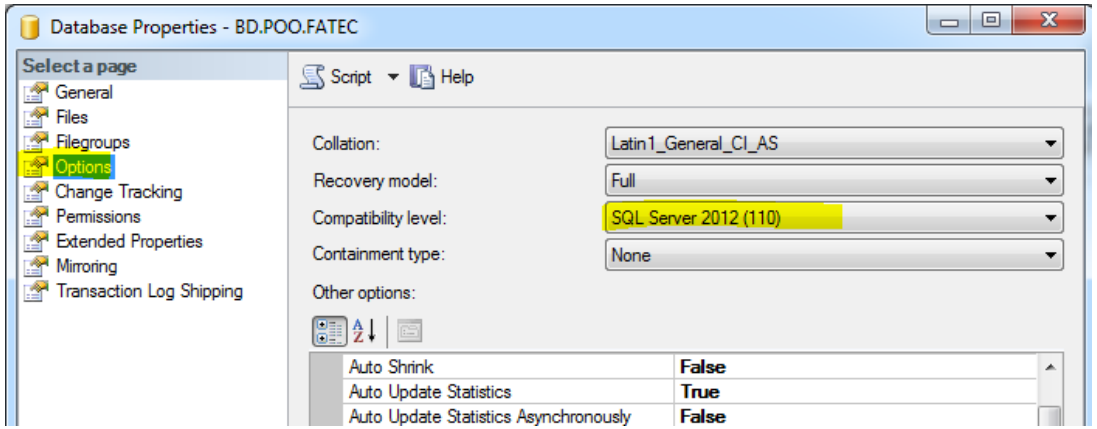


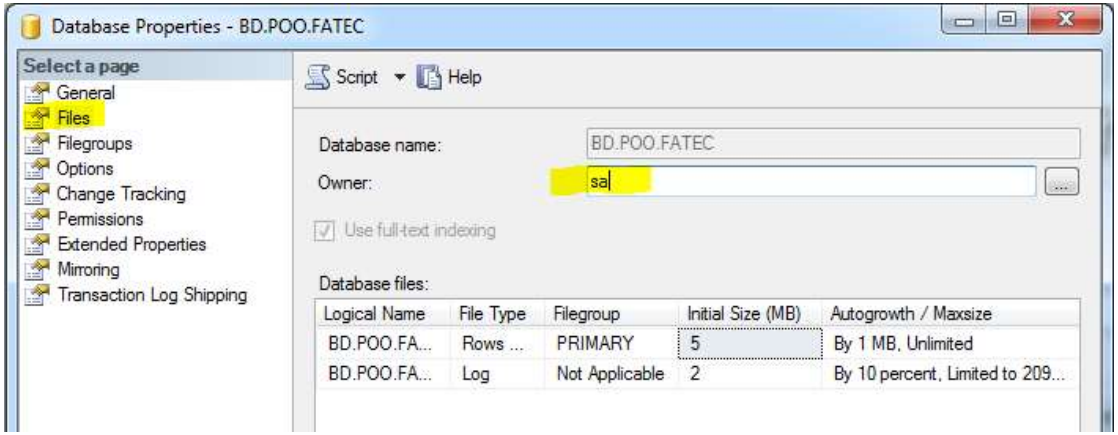
Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

Database Diagram



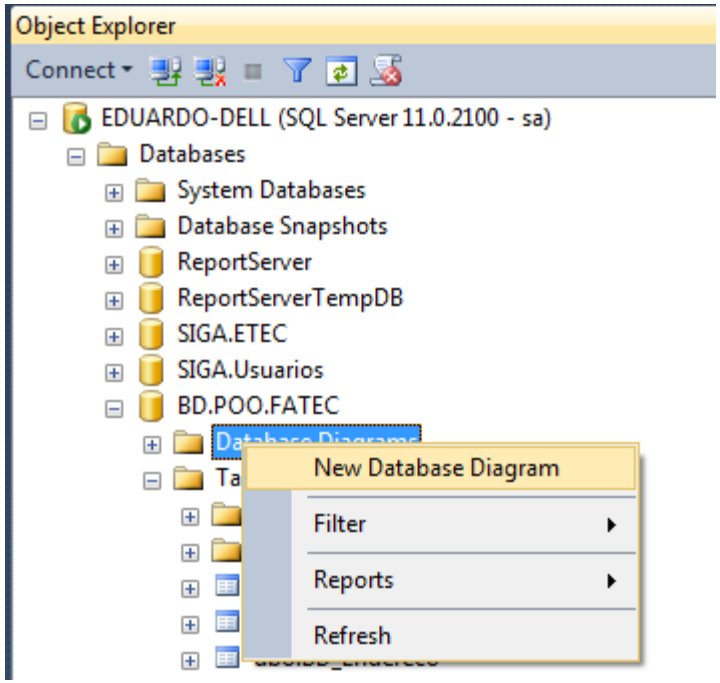


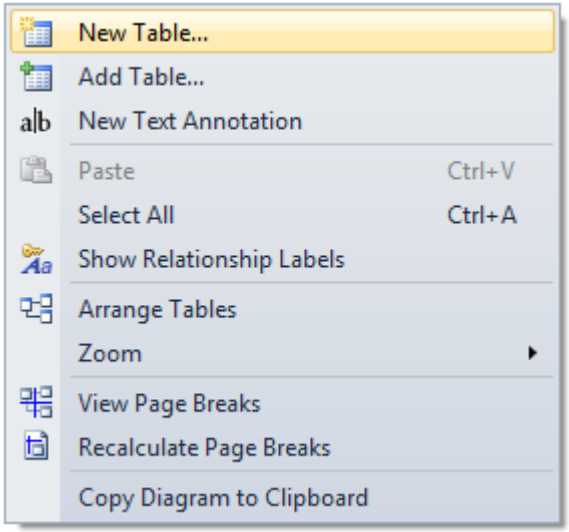


Centro Paula Souza

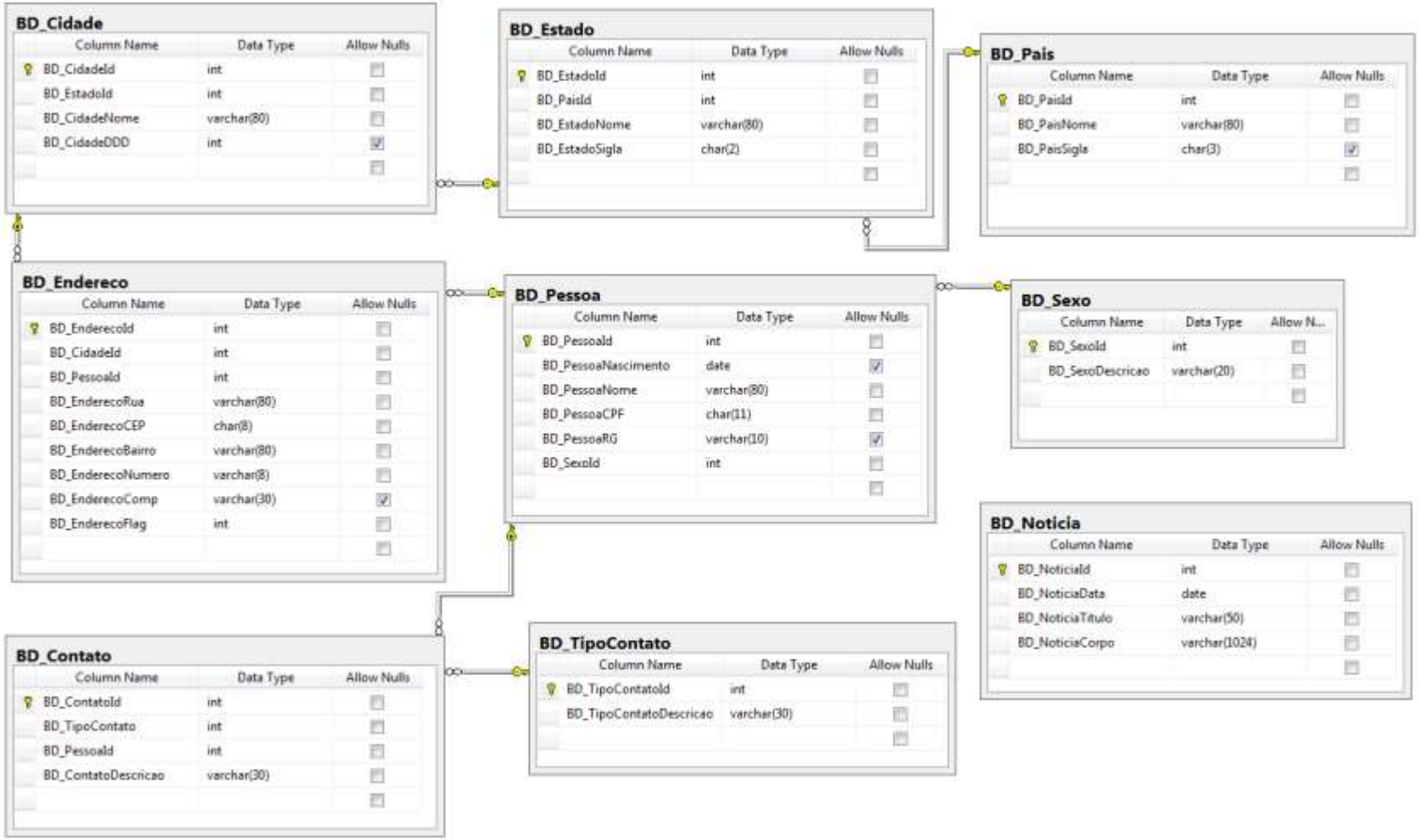
GOVERNO DO ESTADO SÃO PAULO

Database Diagram





Database Diagram

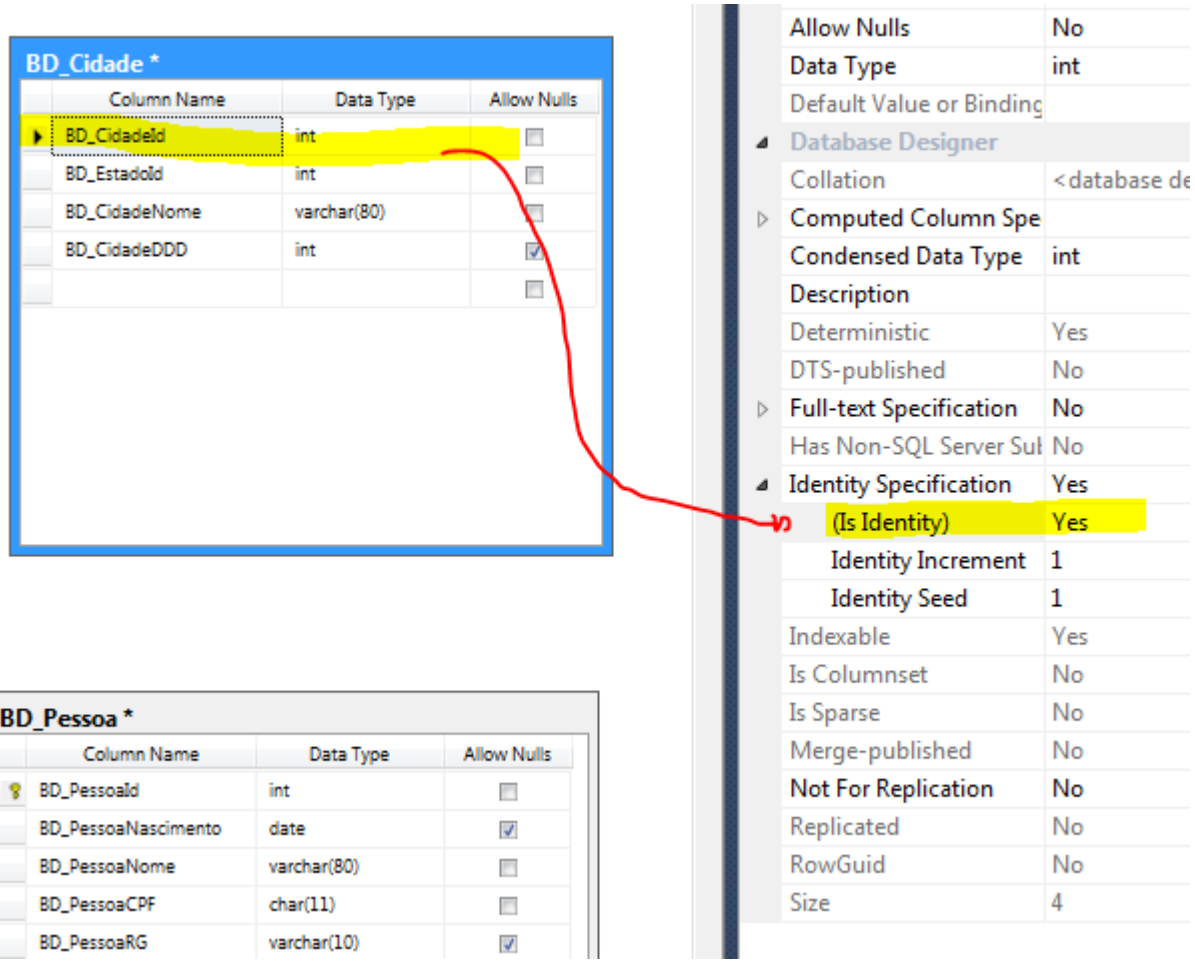


Microsoft SQL Server

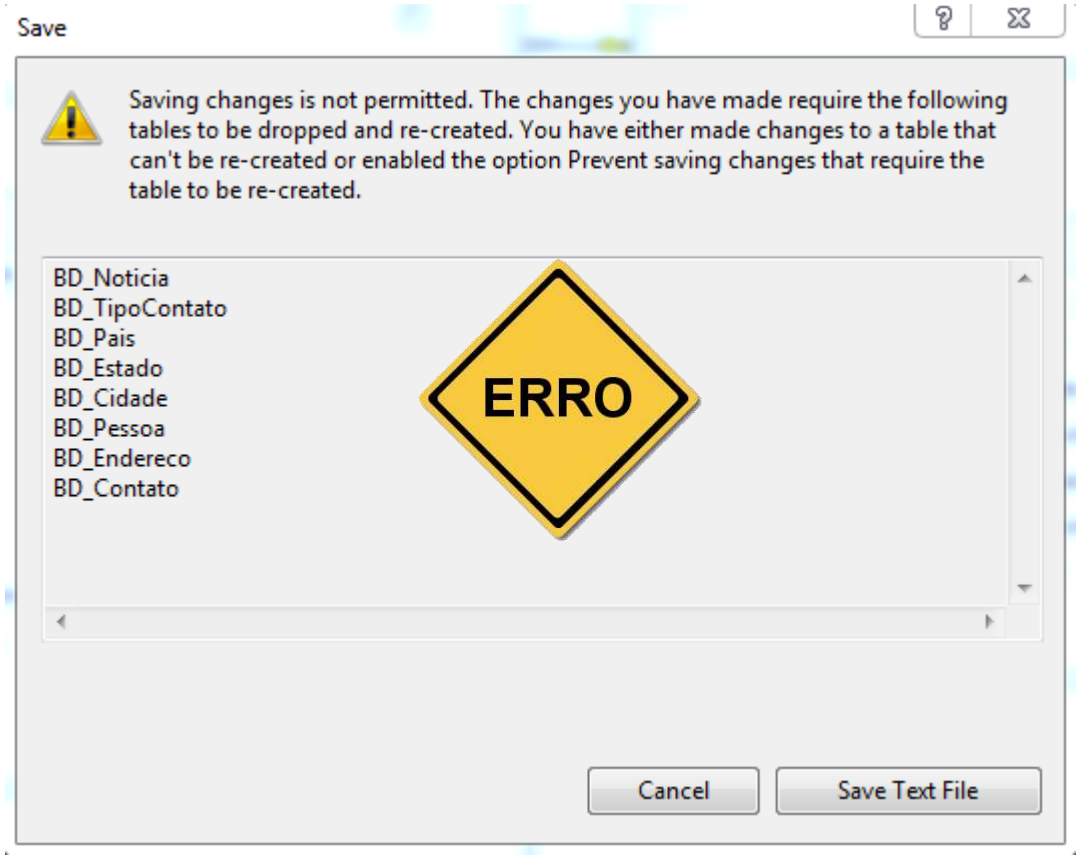
Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

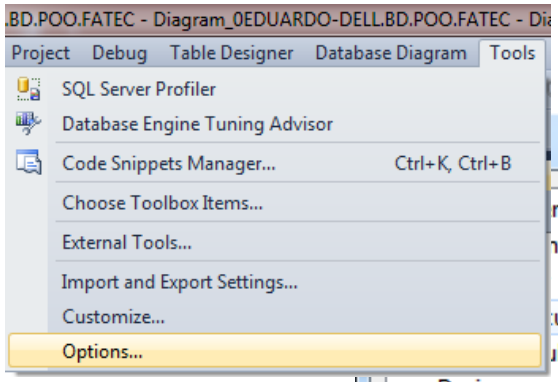
Database Diagram

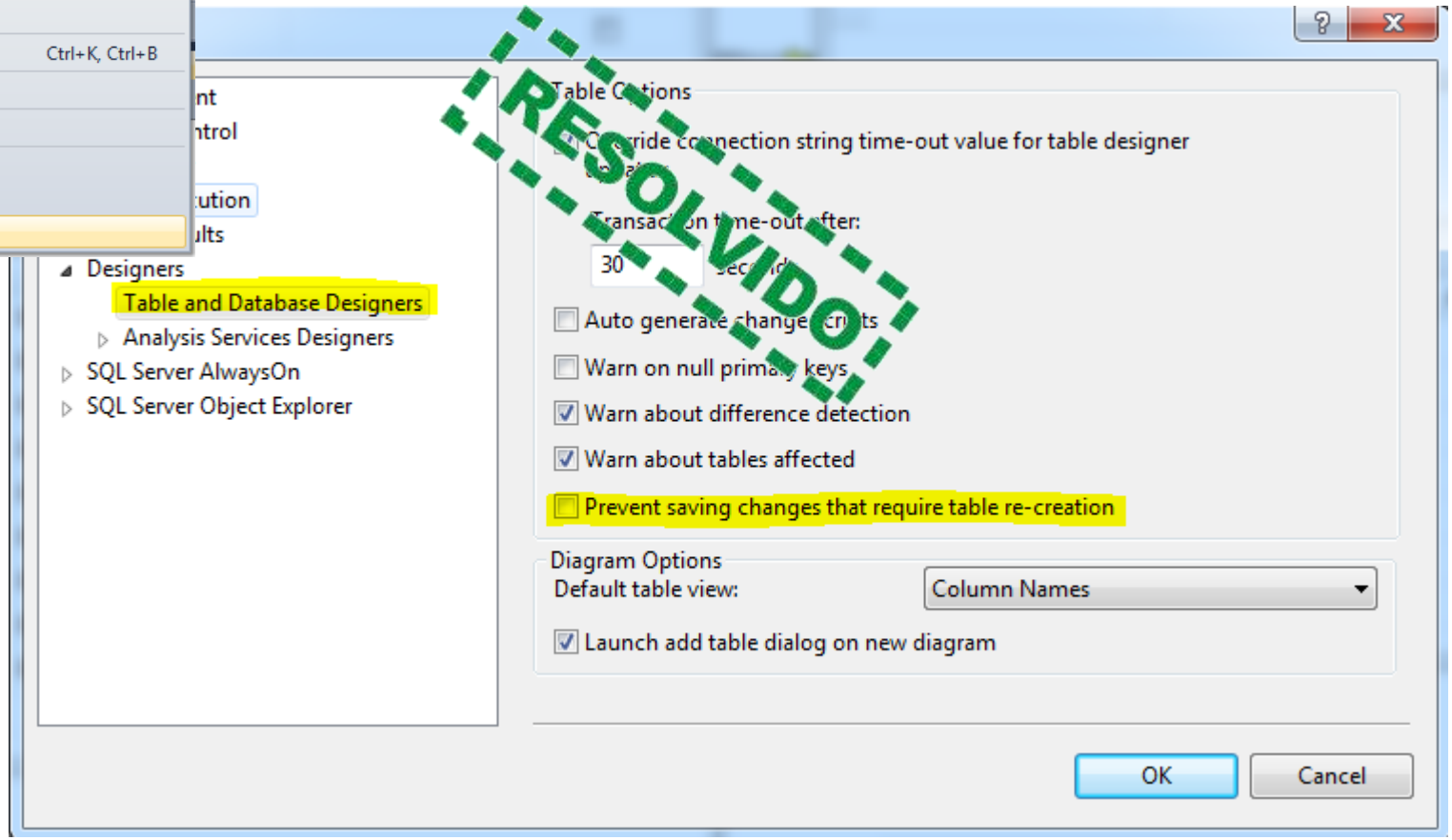


Database Diagram

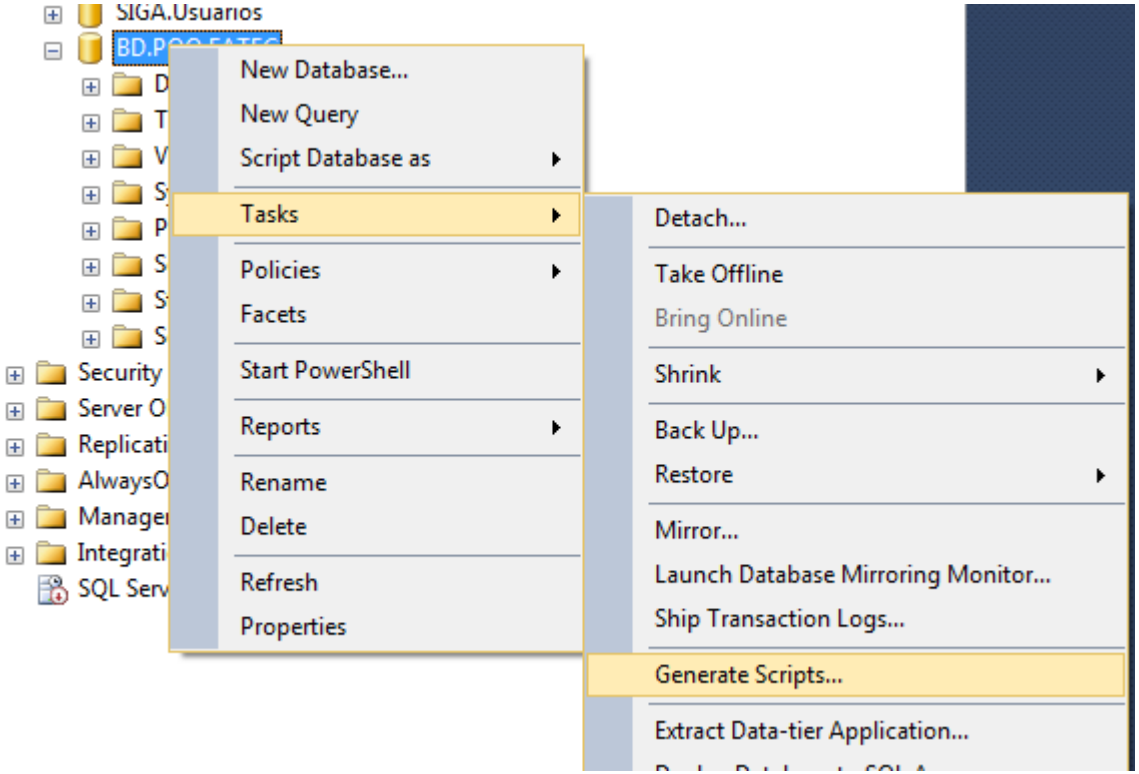


Database Diagram

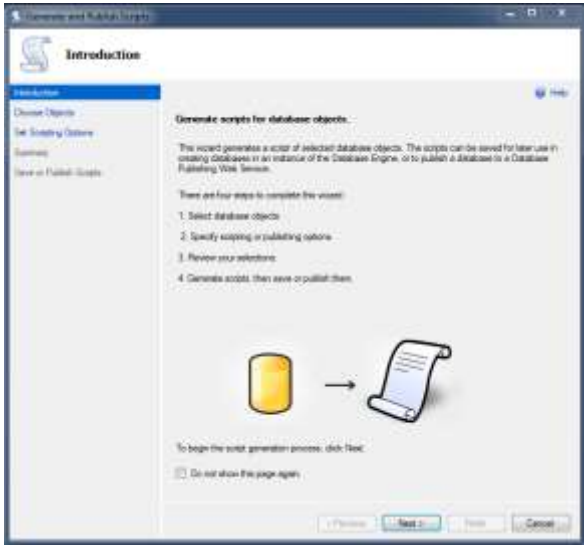


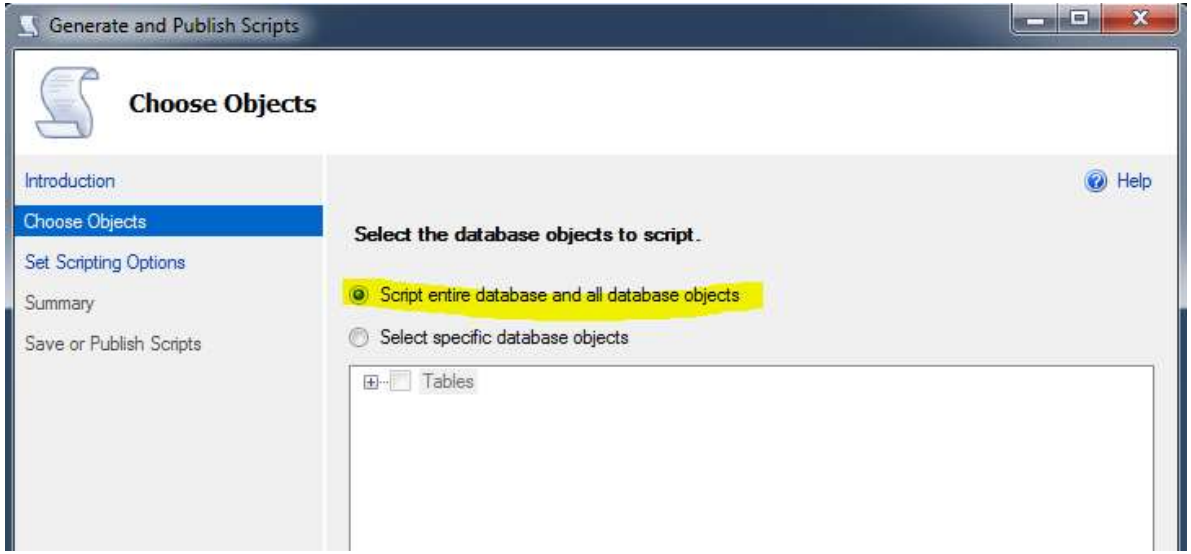


Database Diagram

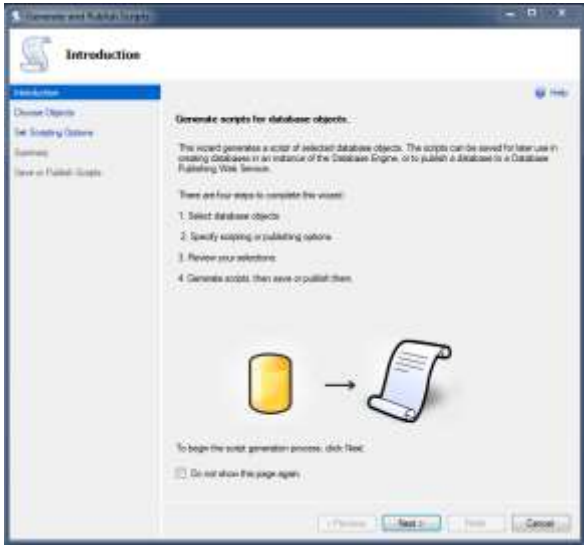


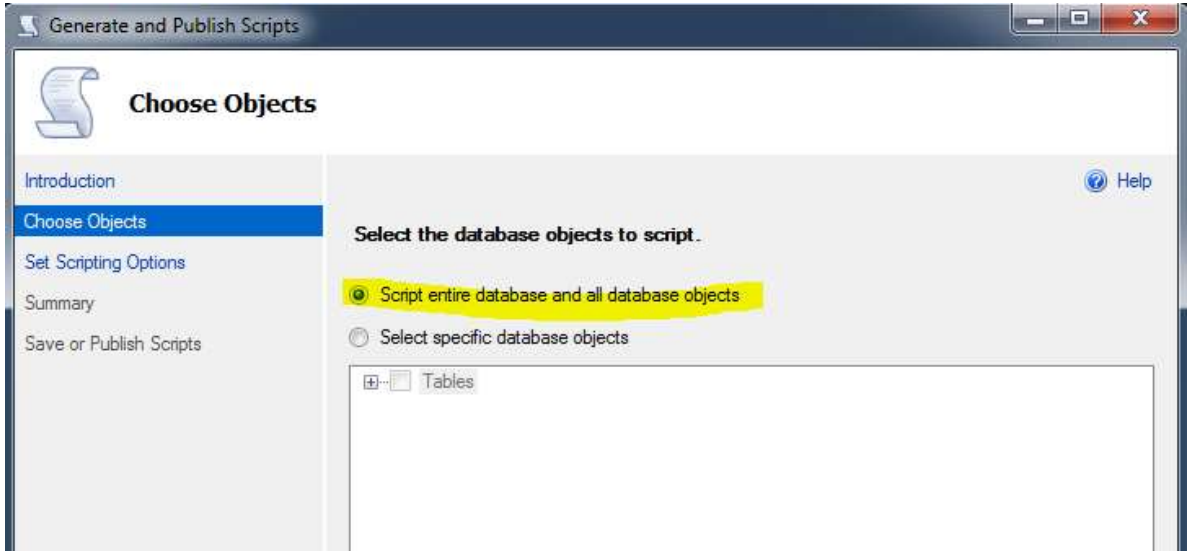
Database Diagram



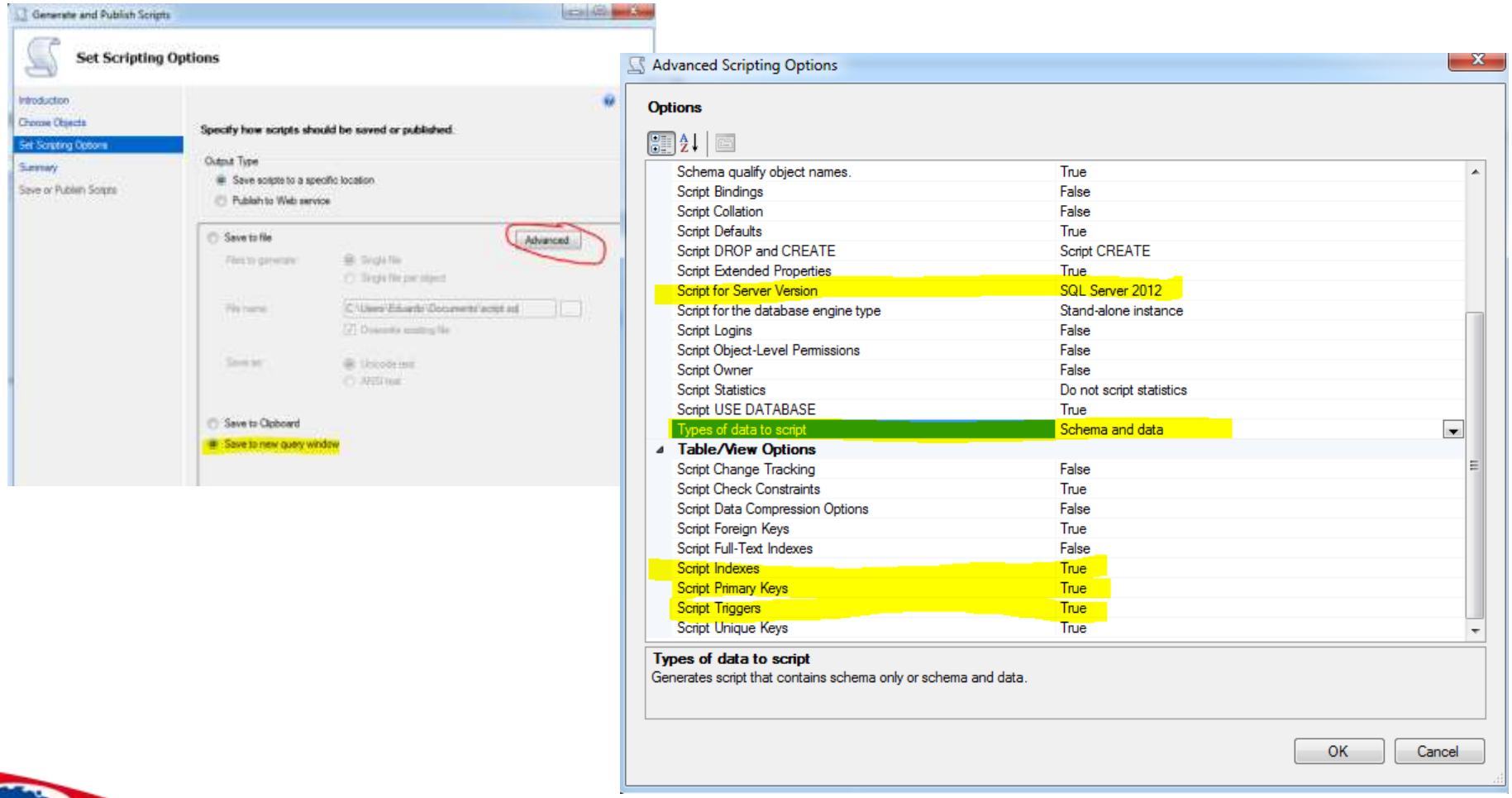


Database Diagram

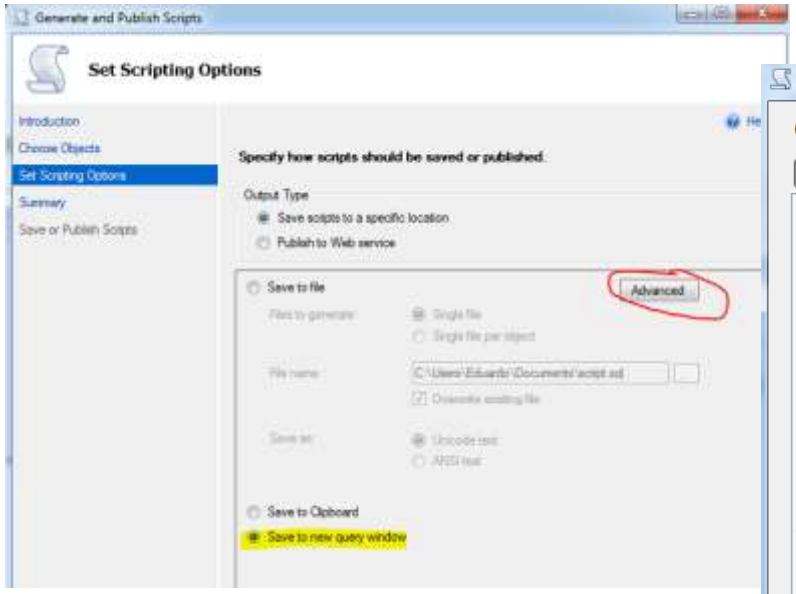


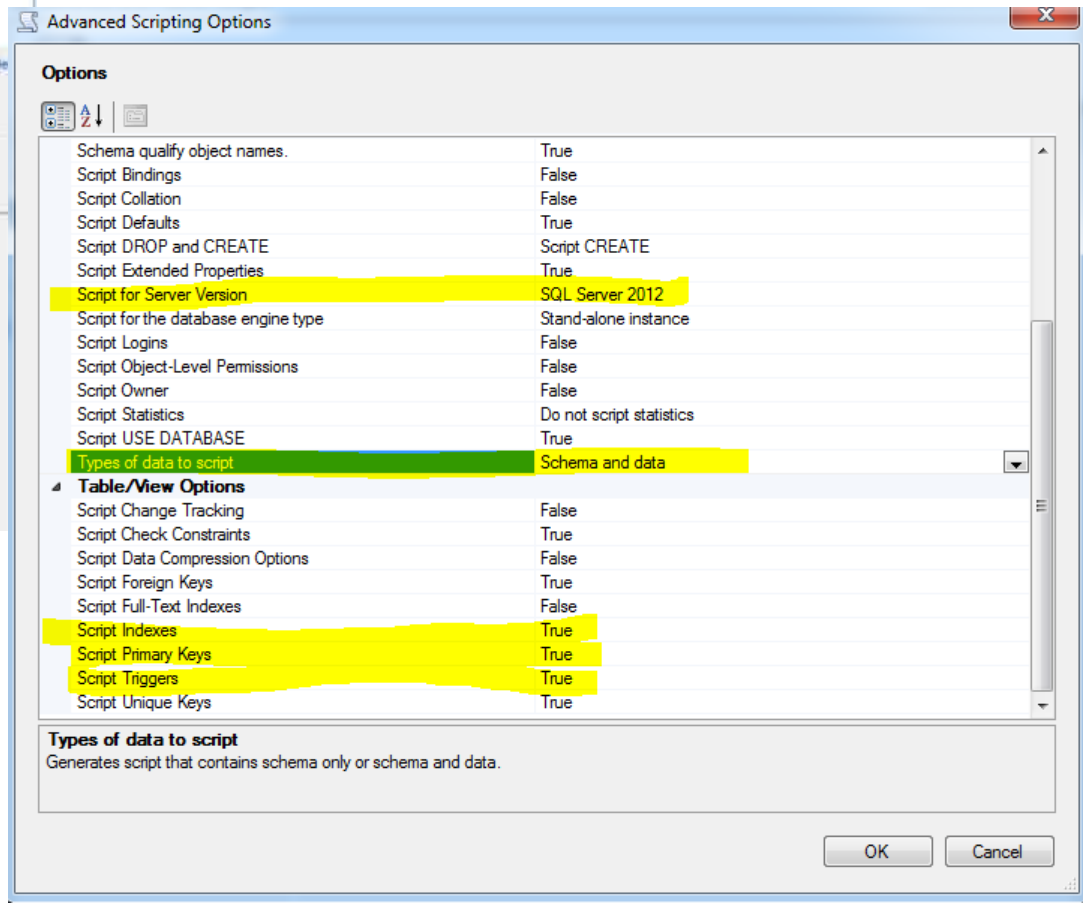


Database Diagram



Database Diagram

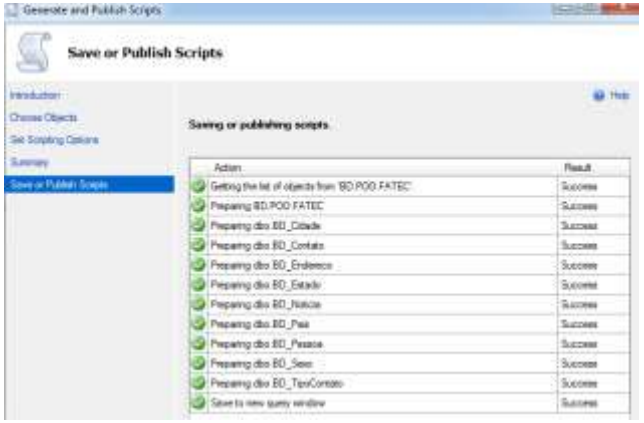




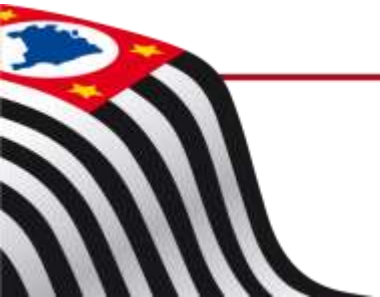
Database Diagram

```
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
SET IDENTITY_INSERT [dbo].[BD_Pais] ON

GO
INSERT [dbo].[BD_Pais] ([BD_PaisId], [BD_PaisNome], [BD_PaisSigla]) VALUES (1, N'Brasil', N'BRA')
GO
SET IDENTITY_INSERT [dbo].[BD_Pais] OFF
GO
ALTER TABLE [dbo].[BD_Cidade]  WITH CHECK ADD  CONSTRAINT [FK_BD_Cidade_BD_Estado] FOREIGN KEY([BD_Estadc
REFERENCES [dbo].[BD_Estado] ([BD_EstadoId])
GO
```

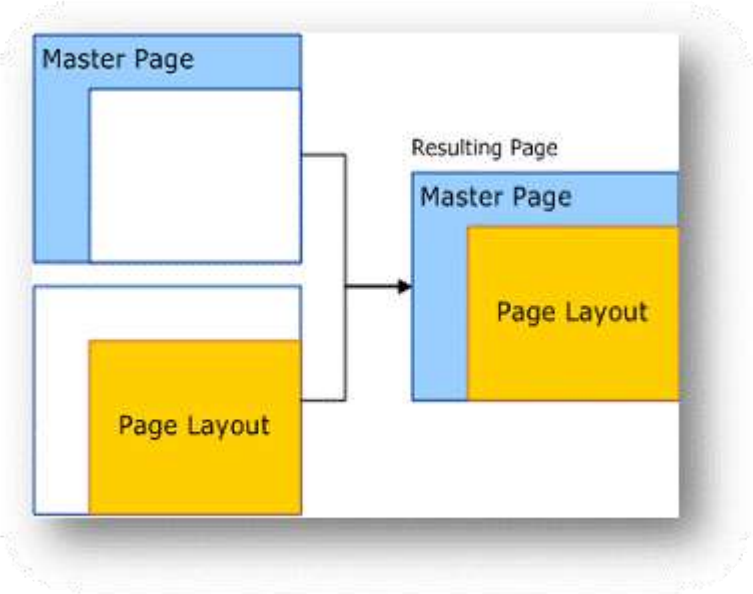
Database Diagram



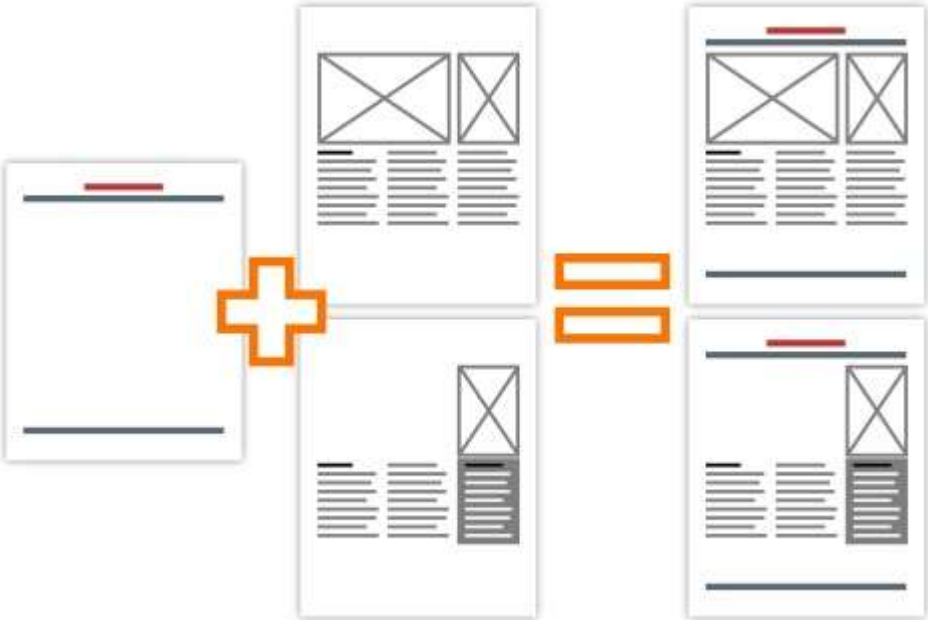
Capítulo 13: ASP.NET



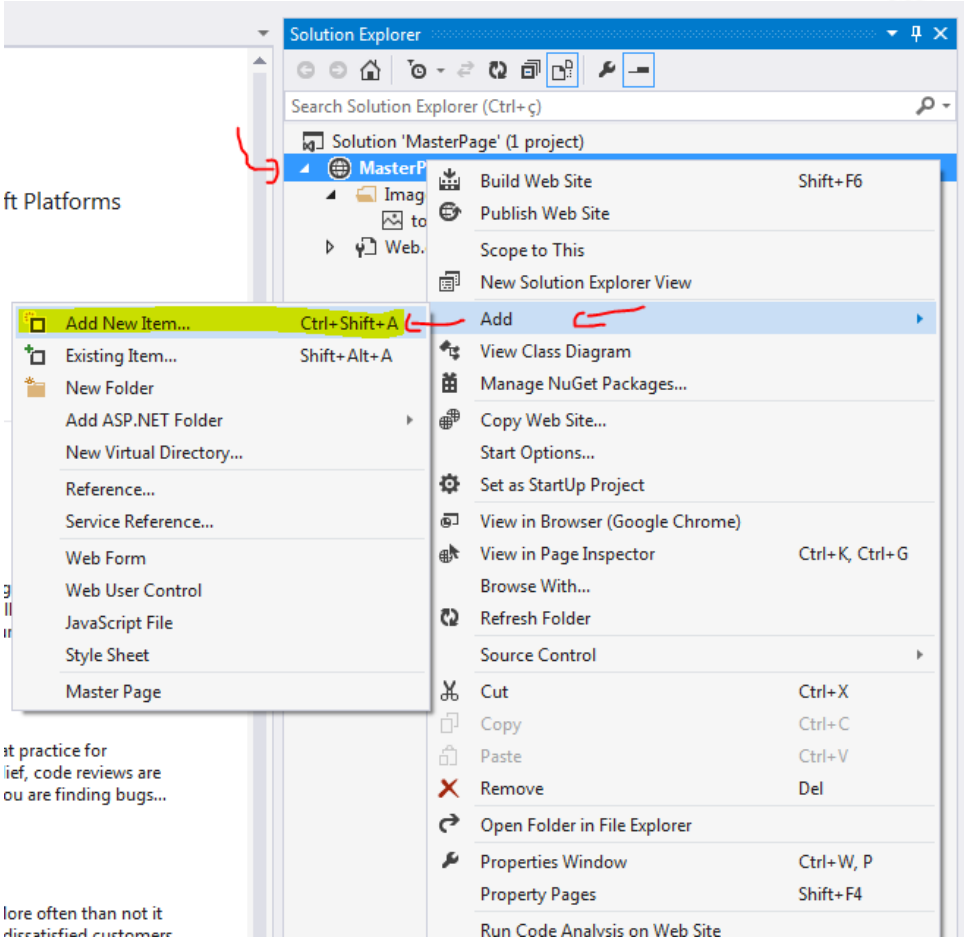
Master Page


Master Page
Page Layout
Resulting Page
Master Page
Page Layout

Master Page

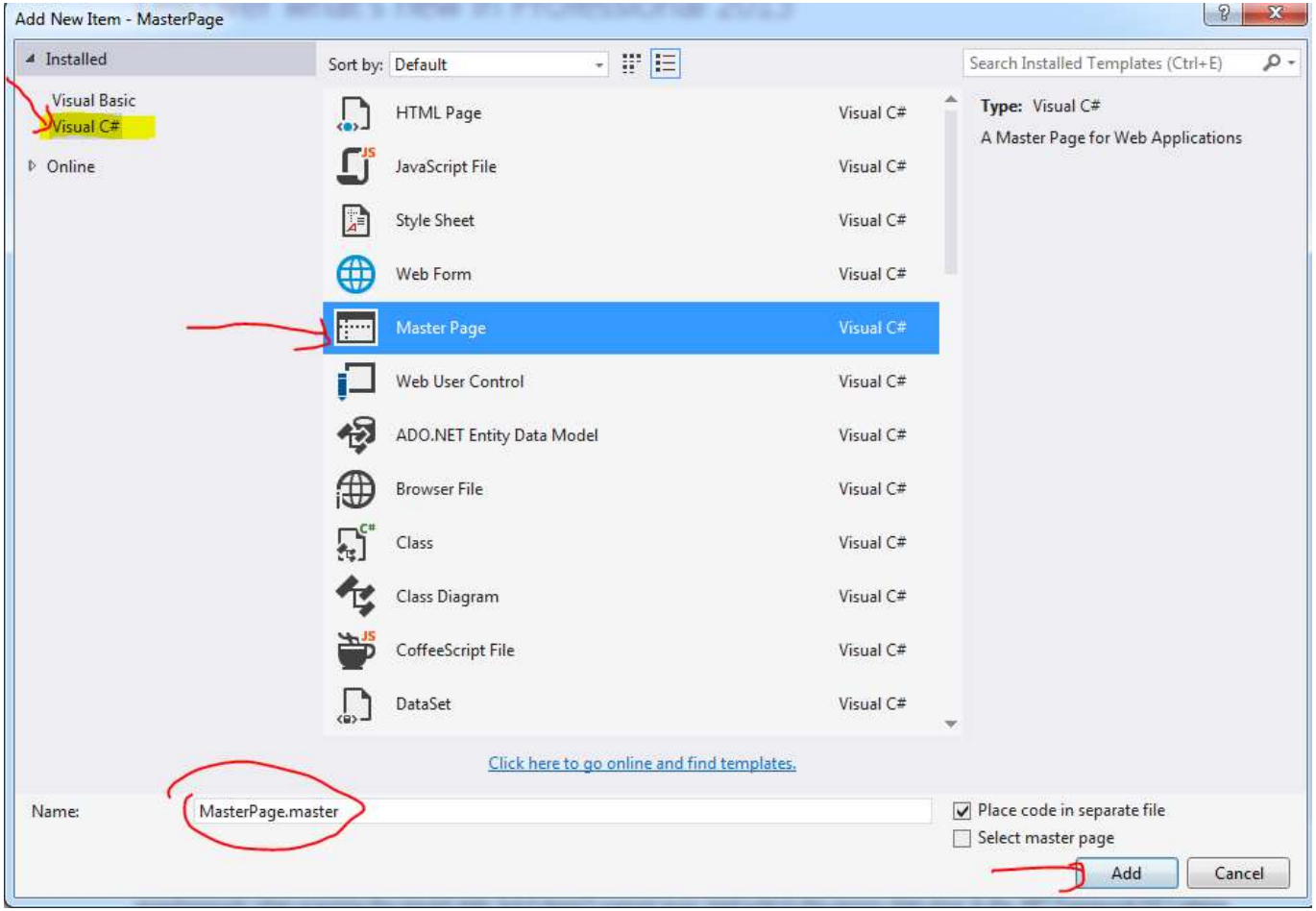


Master Page

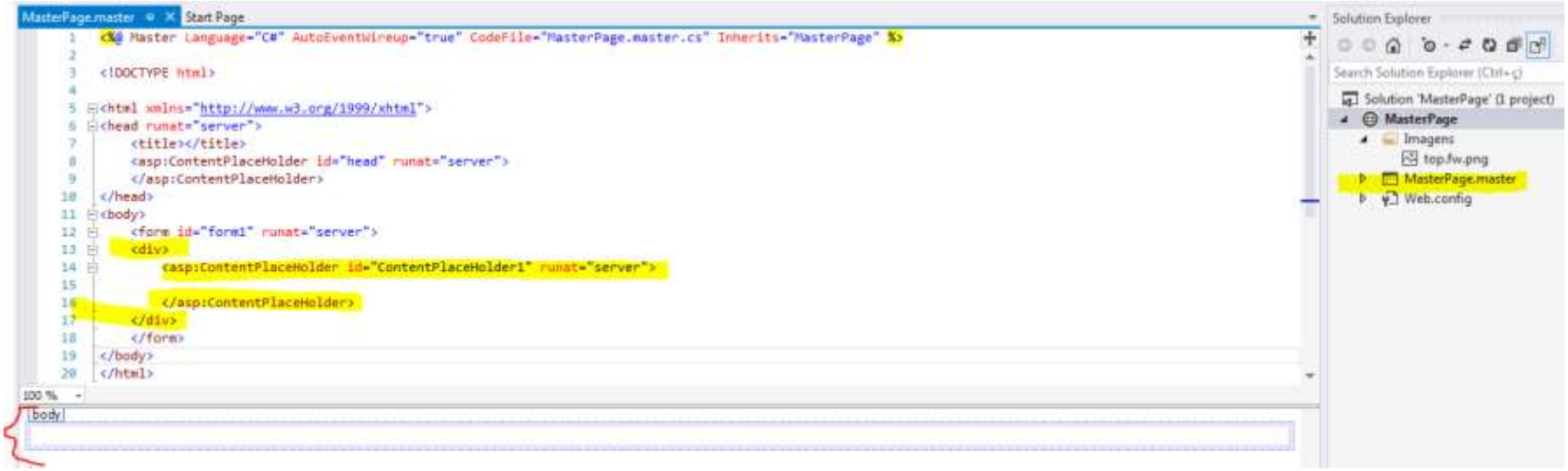


Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Master Page


Add New Item - MasterPage
Installed
Visual Basic
Visual C#
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
HTML Page Visual C#
JavaScript File Visual C#
Style Sheet Visual C#
Web Form Visual C#
Master Page Visual C#
Web User Control Visual C#
ADO.NET Entity Data Model Visual C#
Browser File Visual C#
Class Visual C#
Class Diagram Visual C#
CoffeeScript File Visual C#
DataSet Visual C#
Type: Visual C#
A Master Page for Web Applications
Click here to go online and find templates.
Name:
MasterPage.master
Place code in separate file
Select master page
Add
Cancel

Master Page


MasterPage.master
Start Page
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<asp:ContentPlaceHolder id="head" runat="server">
</asp:ContentPlaceHolder>
</head>
<body>
<form id="form1" runat="server">

<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
</asp:ContentPlaceHolder>

</form>
</body>
</html>
100 %
body
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'MasterPage' (1 project)
MasterPage
Imagens
top.fw.png
MasterPage.master
Web.config

Master Page

MasterPage.master* Start Page

```
9       </asp:ContentPlaceHolder>
10  </head>
11  <body>
12      <form id="form1" runat="server">
13      <div>
14          <table style="width: 100%;">
15              <tr>
16                  <td> </td>
17              </tr>
18              <tr>
19                  <td>
20                      <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
21
22                      </asp:ContentPlaceHolder>
23                  </td>
24              </tr>
25              <tr>
26                  <td> </td>
27              </tr>
28          </table>
29      </div>
30      </form>
31  </body>
32  </html>
33
```

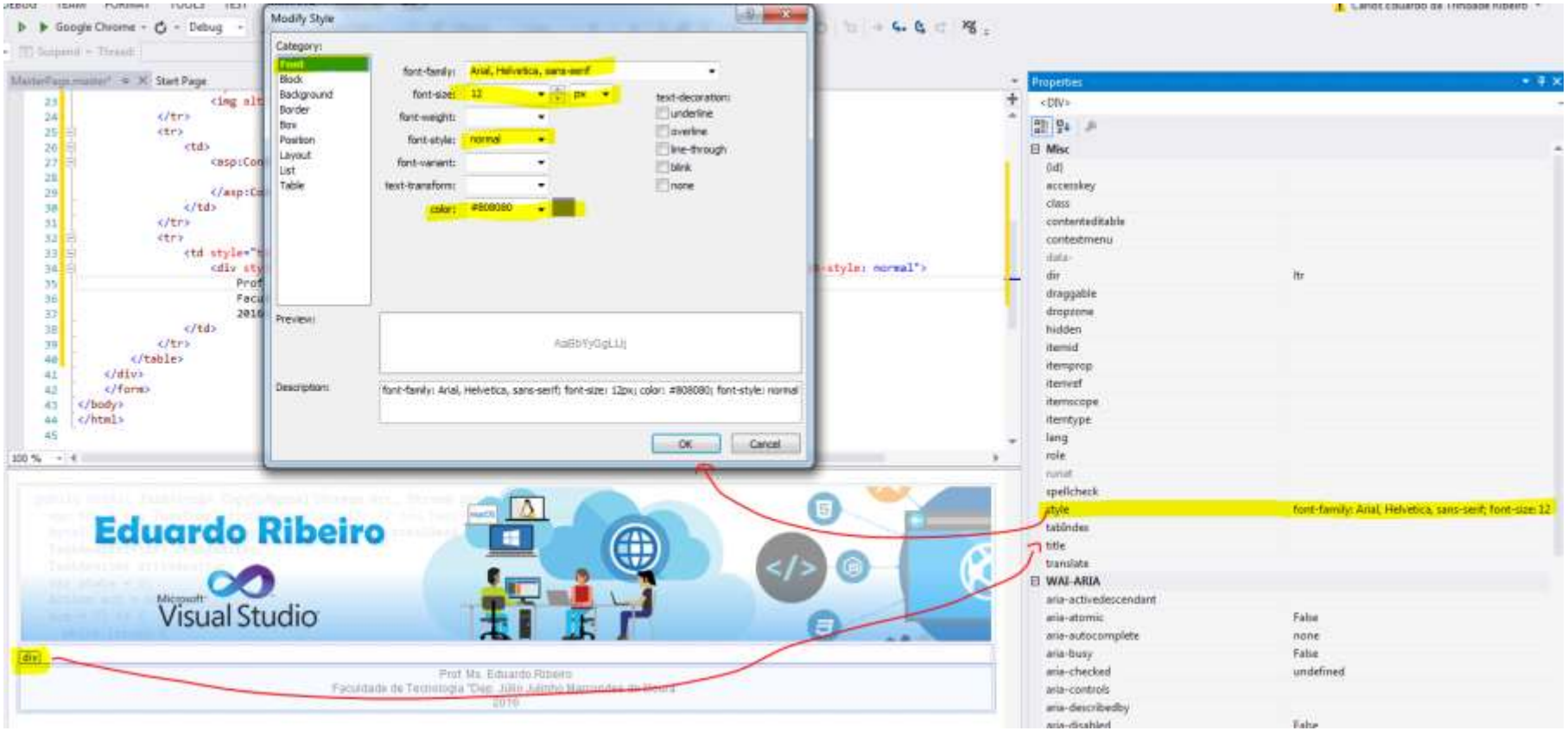
100 %

body

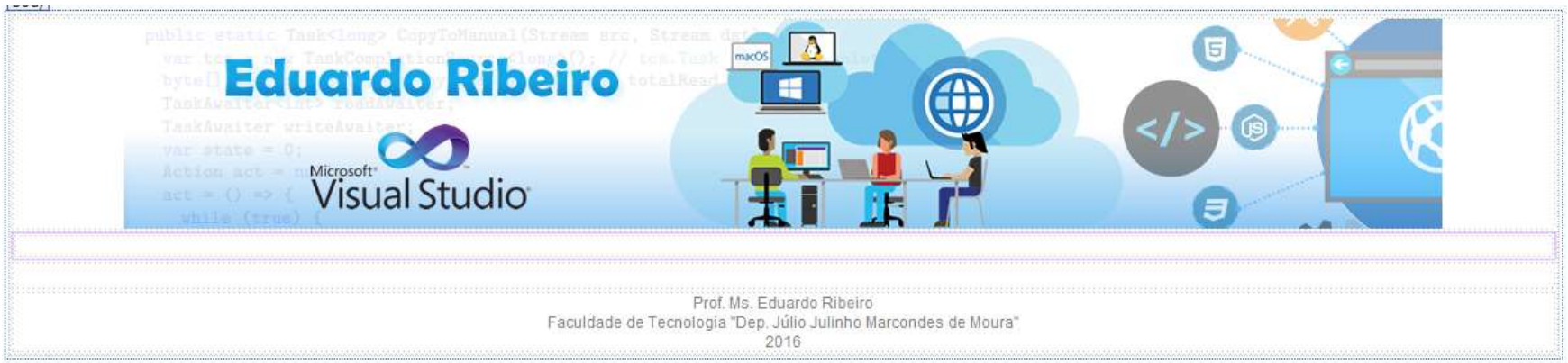
Master Page

MasterPage.master*
Start Page
16 </head>
17 <body>
18 <form id="form1" runat="server">
19
20 <table style="width: 100%;">
21 <tr>
22 <td style="text-align: center">
23 <img alt="ASPNET" class="auto-style1" src="Imagens/top.fw.png" /></td>
24 </tr>
25 <tr>
26 <td>
27 <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
29 </asp:ContentPlaceHolder>
30 </td>
31 </tr>
32 <tr>
33 <td> </td>
34 </tr>
35 </table>
36
37 </form>
38 </body>
39 </html>
100 %
body
Eduardo Ribeiro
Microsoft
Visual Studio
Solution Explorer
Search Solution Explorer (Ctrl+ç)
Solution 'MasterPage' (1 proj
MasterPage
Imagens
top.fw.png
MasterPage.master
Web.config

Master Page


Modify Style
Category:
Font
Block
Background
Border
Box
Position
Layout
List
Table
font-family: Arial, Helvetica, sans-serif
font-size: 12 px
font-weight:
font-style: normal
font-variant:
text-transform:
color: #808080
text-decoration:
underline
overline
line-through
blink
none
Preview:
Description:
font-family: Arial, Helvetica, sans-serif; font-size: 12px; color: #808080; font-style: normal
OK
Cancel
Properties
Misc
style
font-family: Arial, Helvetica, sans-serif; font-size: 12
title
WAI-ARIA
Eduardo Ribeiro
Microsoft Visual Studio
Prof. Ms. Eduardo Ribeiro

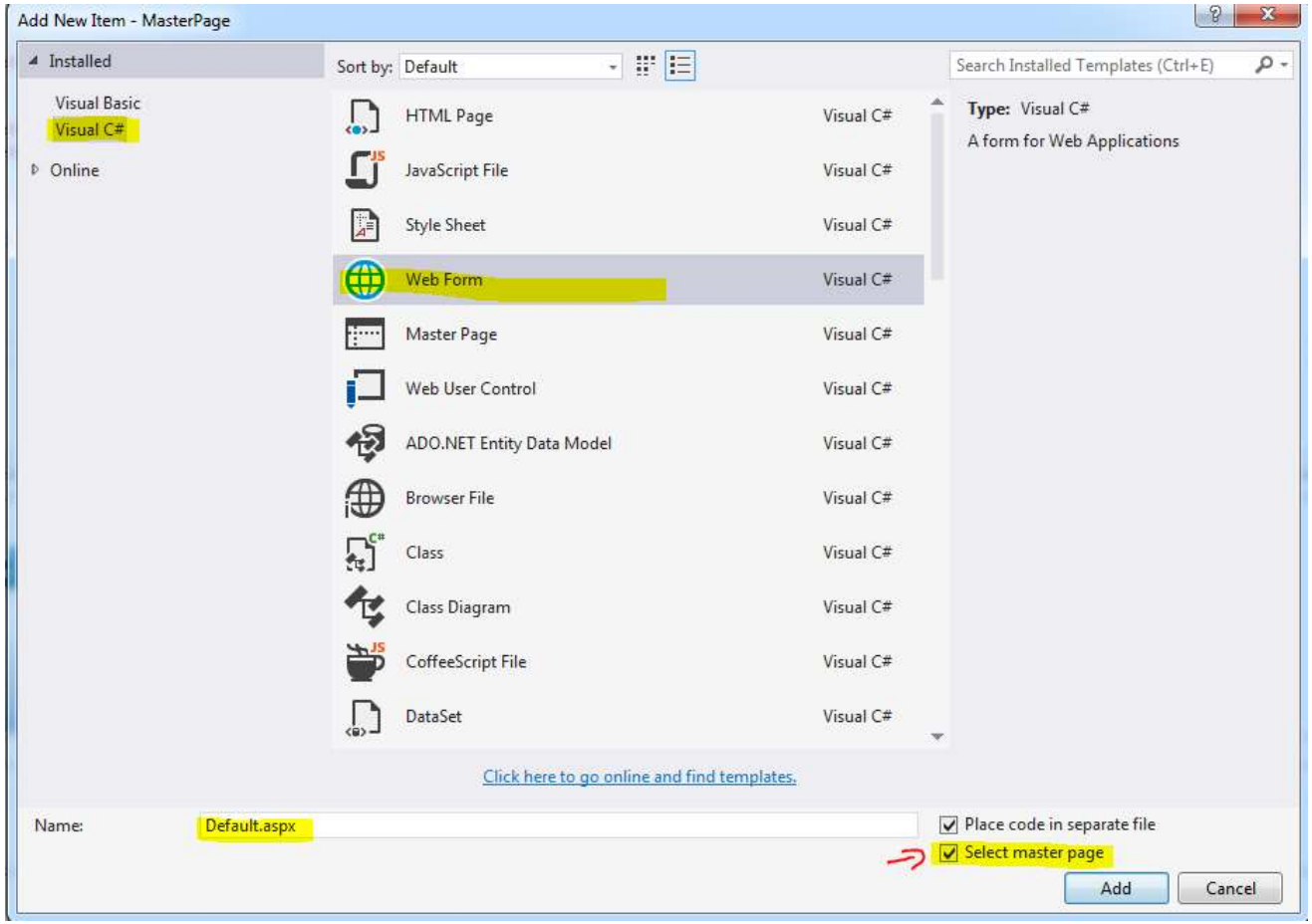
Master Page



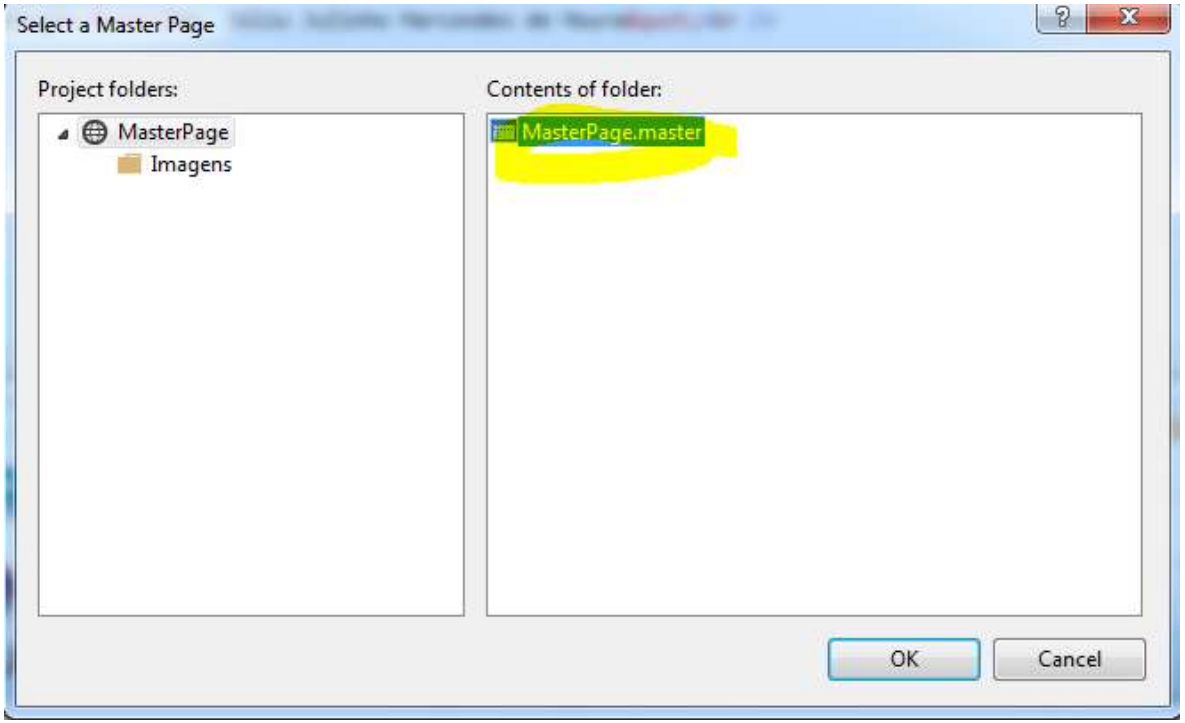
Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

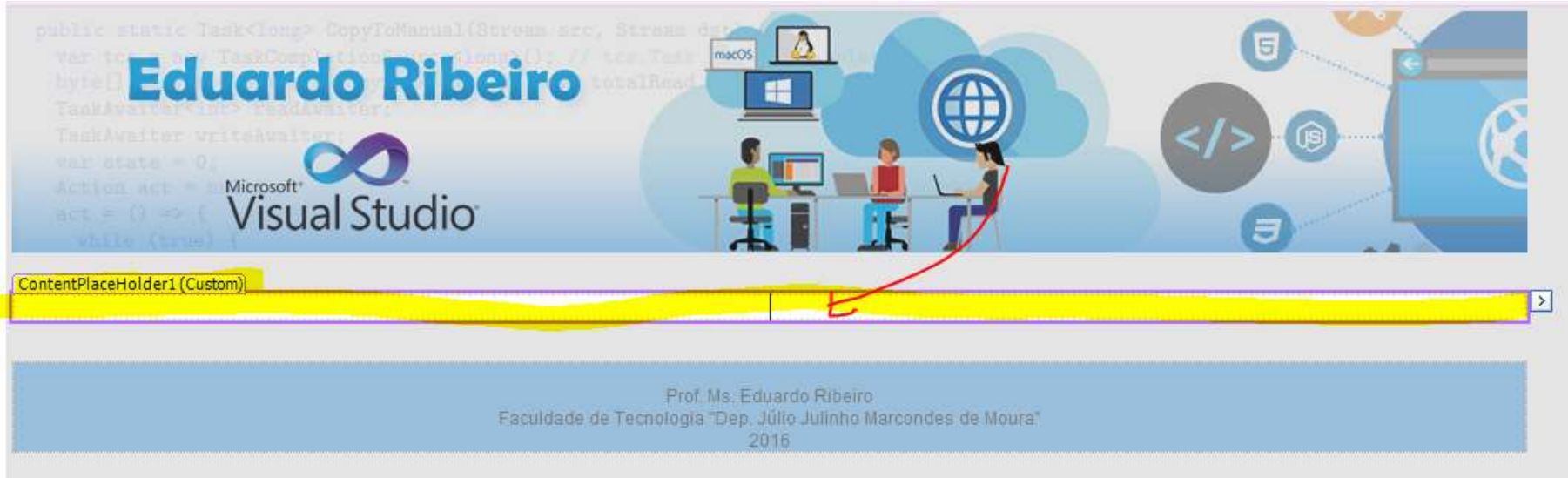
Master Page


Add New Item - MasterPage
Installed
Visual Basic
Visual C#
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
HTML Page Visual C#
JavaScript File Visual C#
Style Sheet Visual C#
Web Form Visual C#
Master Page Visual C#
Web User Control Visual C#
ADO.NET Entity Data Model Visual C#
Browser File Visual C#
Class Visual C#
Class Diagram Visual C#
CoffeeScript File Visual C#
DataSet Visual C#
Type: Visual C#
A form for Web Applications
Click here to go online and find templates.
Name: Default.aspx
Place code in separate file
Select master page
Add
Cancel

Master Page


Select a Master Page
Project folders:
Contents of folder:
MasterPage
Imagens
MasterPage.master
OK
Cancel

Master Page


Eduardo Ribeiro
Microsoft
Visual Studio
macOS
ContentPlaceHolder1 (Custom)
Prof. Ms. Eduardo Ribeiro
Faculdade de Tecnologia "Dep. Júlio Julinho Marcondes de Moura"
2016

Master Page

Centro Paula Souza

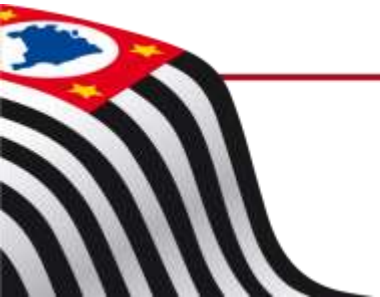
GOVERNO DO ESTADO SÃO PAULO

Master Page

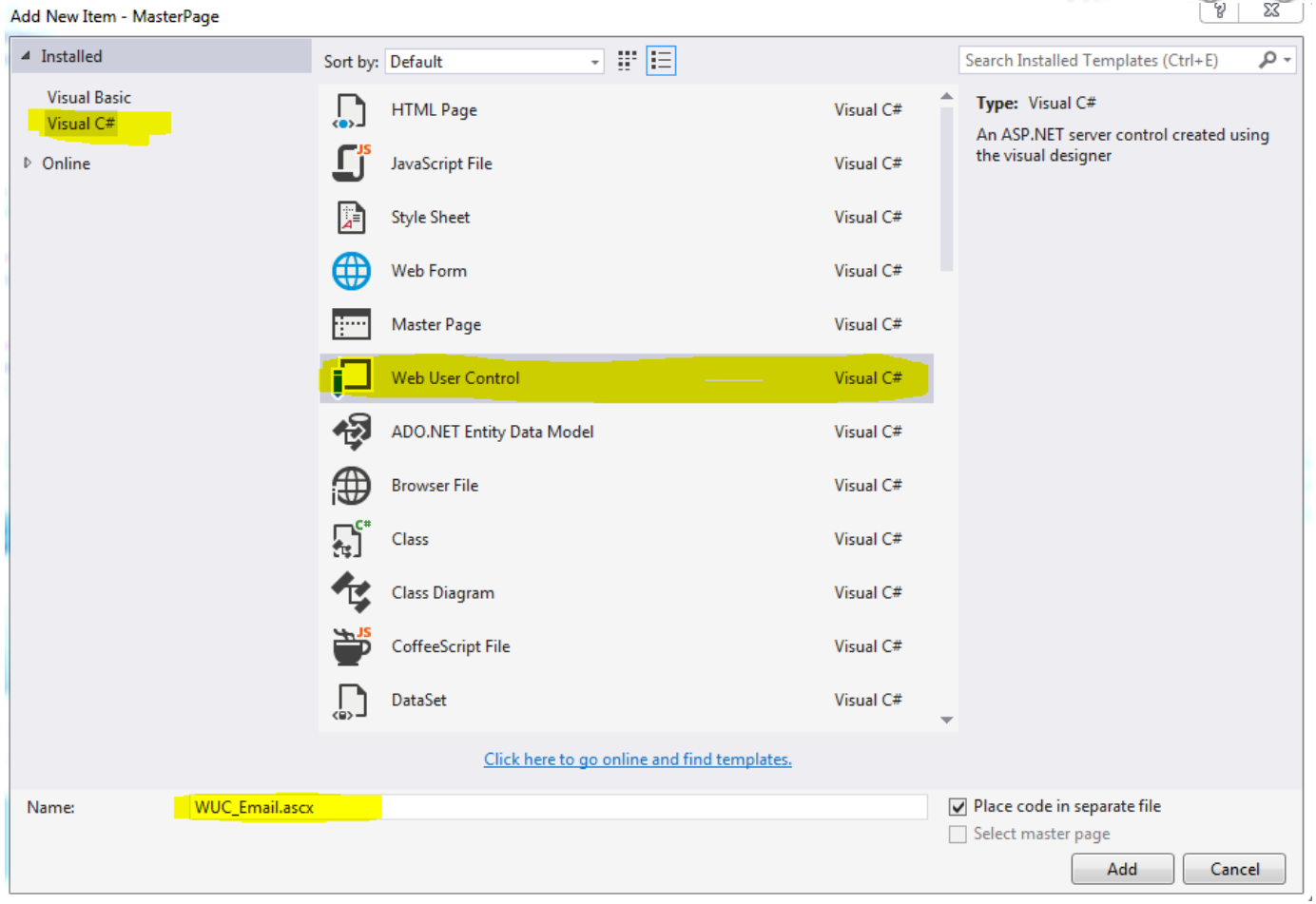
Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

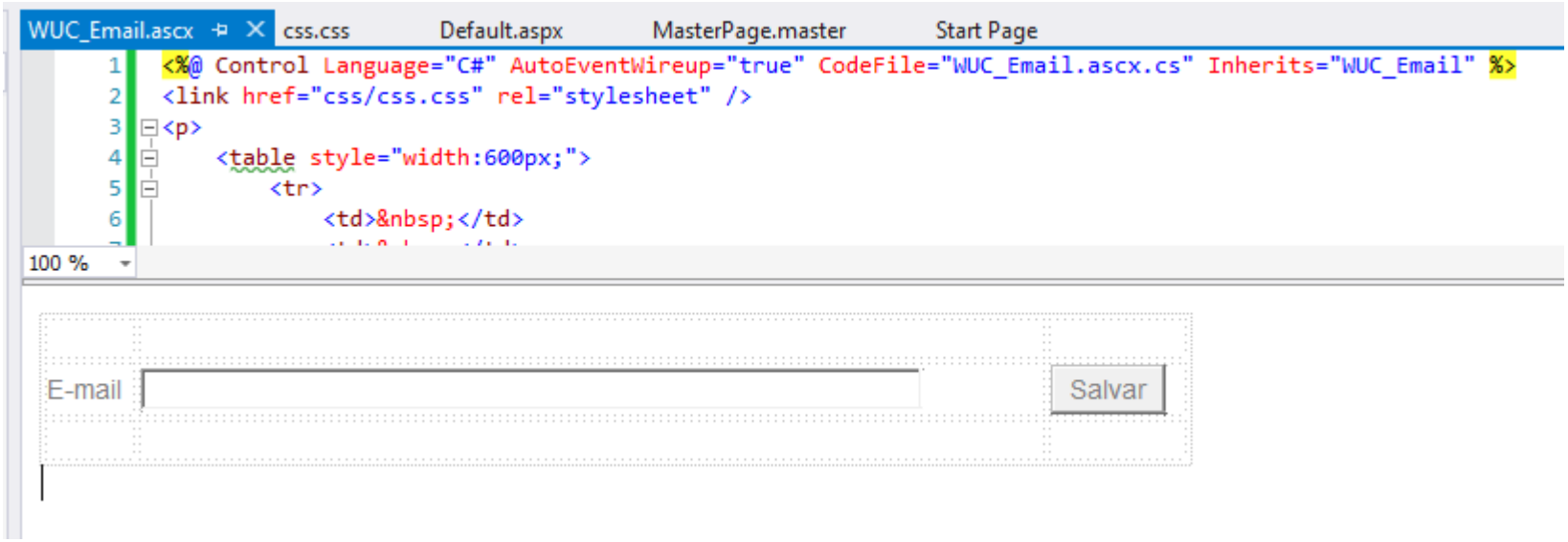
Capítulo 14: ASP.NET



Web User Control



Web User Control

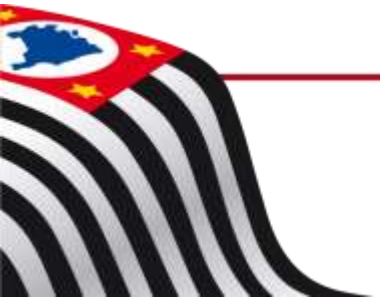


Web User Control

Centro Paula Souza

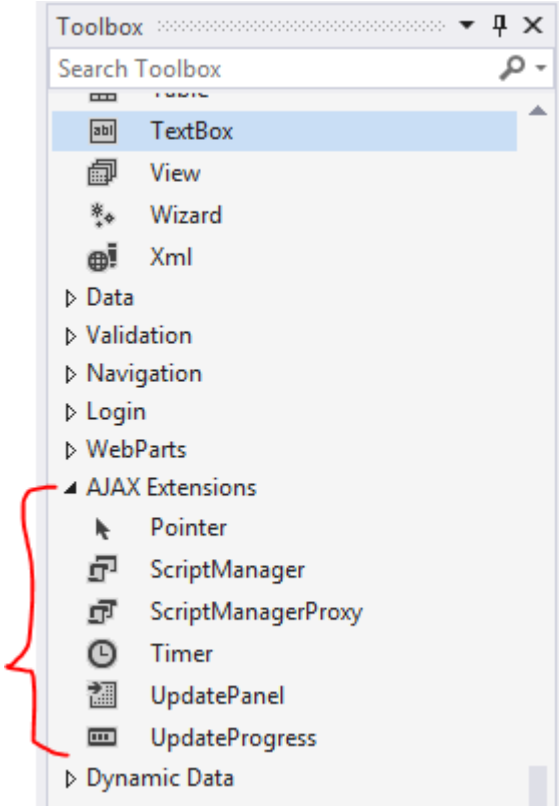
GOVERNO DO ESTADO SÃO PAULO

Capítulo 15: ASP.NET



Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

AJAX


Toolbox
Search Toolbox
TextBox
View
Wizard
Xml
Data
Validation
Navigation
Login
WebParts
AJAX Extensions
Pointer
ScriptManager
ScriptManagerProxy
Timer
UpdatePanel
UpdateProgress
Dynamic Data

AJAX

AJAX Extensions
Pointer
ScriptManager
ScriptManagerProxy
Timer
UpdatePanel
UpdateProgress
Dynamic Data
Pointer
DynamicControl
DynamicDataManager
DynamicEntity
DynamicFilter
DynamicHyperLink
DynamicValidator
Reporting
HTML
Microsoft Visual Studio
ContentPlaceHolder1 (Custom)
td
ScriptManager - ScriptManager1
Prof. Ms. Eduardo Ribeiro
Faculdade de Tecnologia "Dep. Júlio Julinho Marcondes de Moura"
2016

AJAX

Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

AJAX

MasterPage.master | css.css | Ajax.aspx.cs | Ajax.aspx

Ajax

```
7
8   public partial class Ajax : System.Web.UI.Page
9   {
10      protected void Page_Load(object sender, EventArgs e)
11      {
12          if (!IsPostBack)
13          {
14              DDL_Estado.Items.Add("AC");
15              DDL_Estado.Items.Add("AL");
16              DDL_Estado.Items.Add("AP");
17              DDL_Estado.Items.Add("AM");
18              DDL_Estado.Items.Add("BA");
19              DDL_Estado.Items.Add("CE");
20              DDL_Estado.Items.Add("DF");
21              DDL_Estado.Items.Add("ES");
22              DDL_Estado.Items.Add("GO");
23              DDL_Estado.Items.Add("MA");
24              DDL_Estado.Items.Add("MS");
25              DDL_Estado.Items.Add("MG");
26              DDL_Estado.Items.Add("SC");
27              DDL_Estado.Items.Add("SP");
28              DDL_Estado.Items.Add("RJ");
29          }
30      }
```

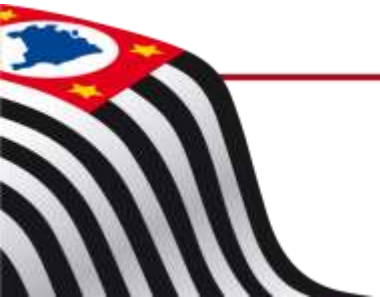
AJAX

```
protected void DDL_Estado_SelectedIndexChanged(object sender, EventArgs e)
{
    if (DDL_Estado.SelectedValue == "SP")
    {
        DDL_Cidades.Items.Clear();
        DDL_Cidades.Items.Add("Selecione");
        DDL_Cidades.Items.Add("Tupã");
        DDL_Cidades.Items.Add("Marília");
        DDL_Cidades.Items.Add("Garça");
        DDL_Cidades.Items.Add("Bauru");
        DDL_Cidades.Items.Add("Campinas");
        DDL_Cidades.Items.Add("São Paulo");
    }
    if (DDL_Estado.SelectedValue == "RJ")
    {
        DDL_Cidades.Items.Clear();
        DDL_Cidades.Items.Add("Selecione");
        DDL_Cidades.Items.Add("Rio de Janeiro");
        DDL_Cidades.Items.Add("Paraty");
        DDL_Cidades.Items.Add("Angra dos Reis");
        DDL_Cidades.Items.Add("Niterói");
        DDL_Cidades.Items.Add("Armação dos Búzios");
        DDL_Cidades.Items.Add("São João da Barra");
    }
    if (DDL_Estado.SelectedIndex == 0)
    {
        DDL_Cidades.Items.Clear();
    }
}
protected void Bt_Carregar_Click(object sender, EventArgs e)
{
    System.Threading.Thread.Sleep(3000);
    lbl_ver.Text = "Estado: " + DDL_Estado.SelectedValue.ToString() + " - Cidade: " + DDL_Cidades.SelectedValue.ToString();
}
```

AJAX

```
<asp:UpdateProgress ID="UpdateProgress1" runat="server">
    <ProgressTemplate>
       <img src="Imagens/loading.png"
                style="width:160px; height:160px;
                position:absolute;  top:50%;
                margin-top:-100px;  left:50%;
                margin-left:-100px; background-color: #FFFFFF;"
        />
    </ProgressTemplate>
</asp:UpdateProgress>
```

Capítulo 16a: ASP.NET



GridView

Cadastro de Países

Sigla

Nome

Cadastrar

	Código	Nome	Sigla
	4	Argentina	ARG
	2	Brasil	BRA
	3	Estados Unidos	USA
	5	México	MEX

1 2

SQLQuery1.sql - (l...POO.FATEC (sa (53))*

```
select * from BD_Pais
```

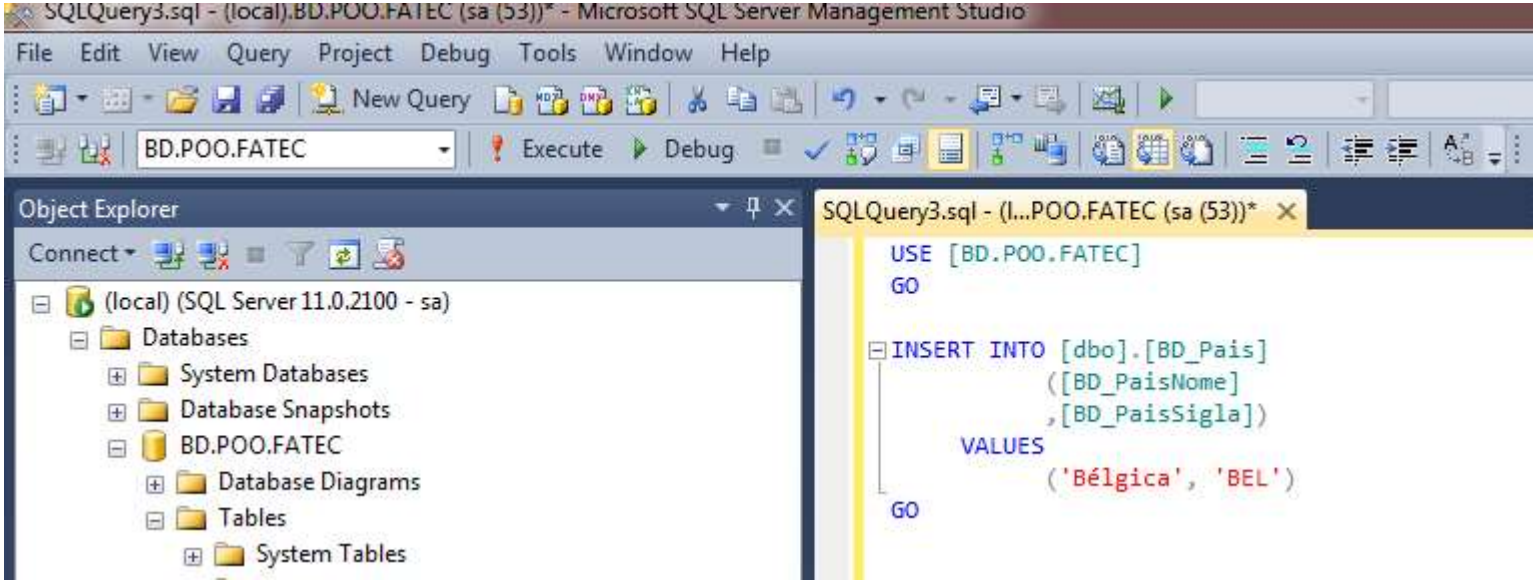
100 %

Results | Messages

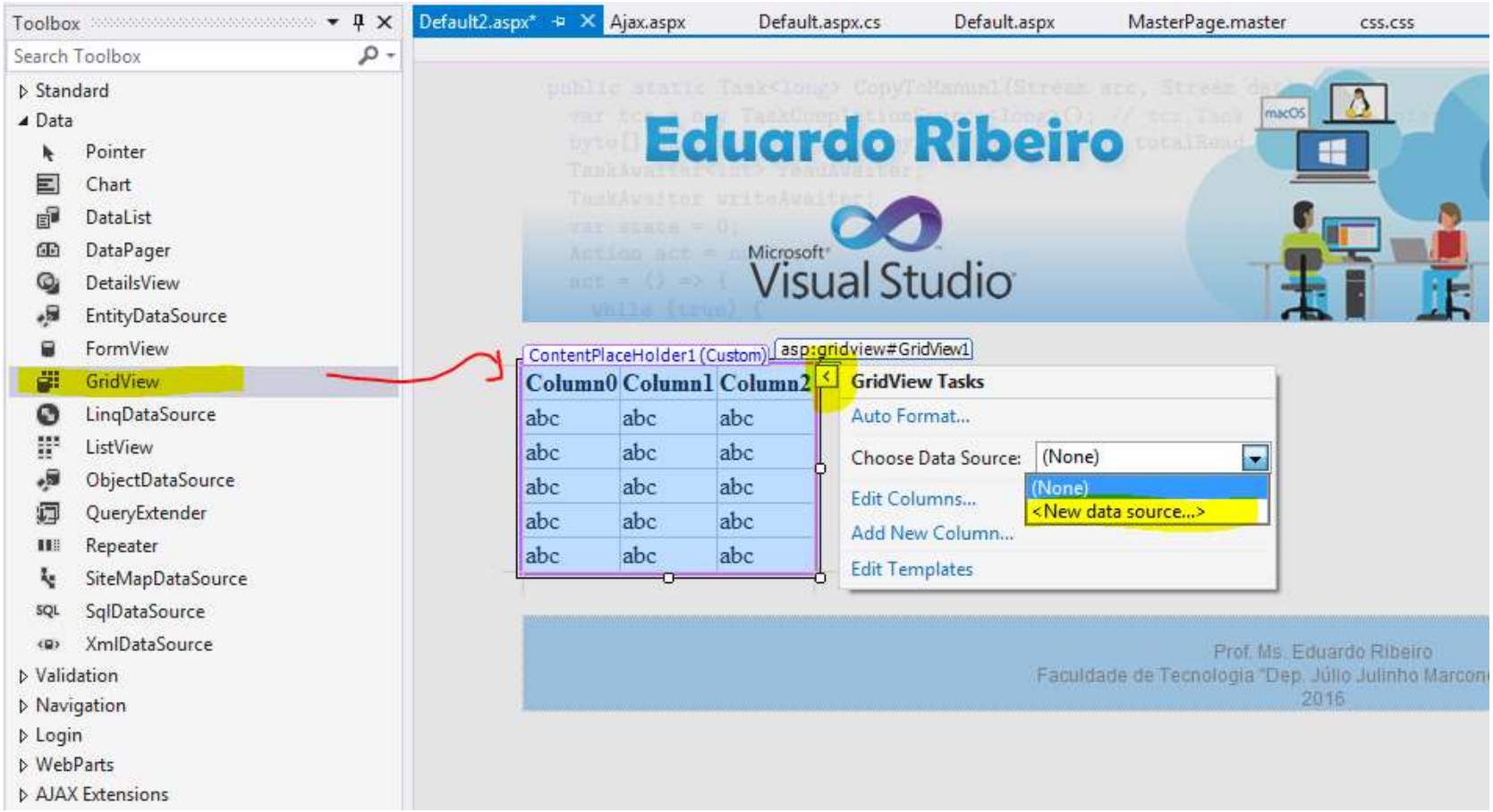
	BD_PaisId	BD_PaisNome	BD_PaisSigla
1	2	Brasil	BRA
2	3	Estados Unidos	USA
3	4	Argentina	ARG
4	5	México	MEX
5	6	Mongólia	MGL

Prof. Ms. Eduardo Ribeiro
Faculdade de Tecnologia "Dep. Júlio Julinho Marcondes de Moura"
2016

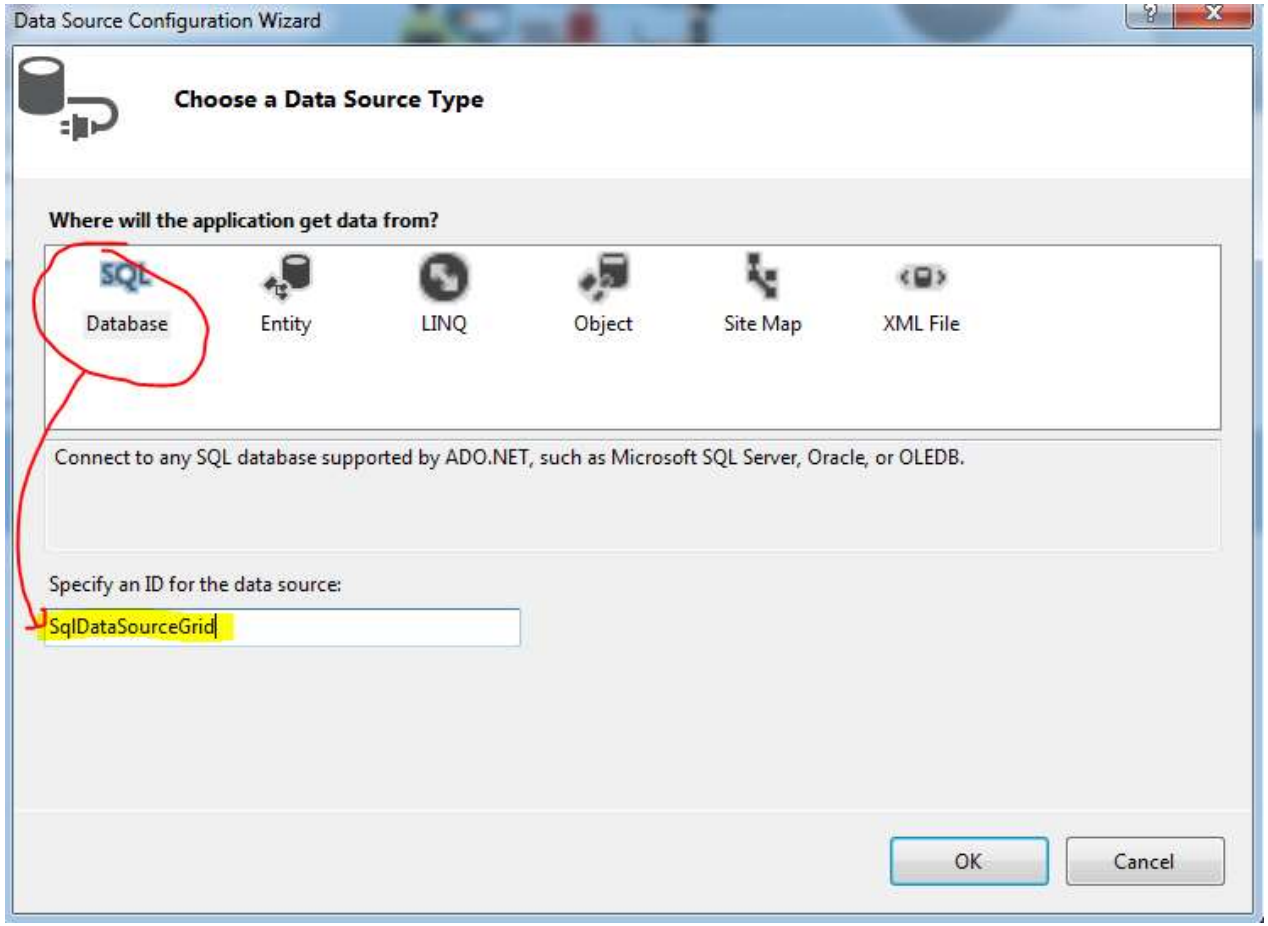
GridView


SQLQuery3.sql - (local).BD.POO.FATEC (sa (53))* - Microsoft SQL Server Management Studio
File Edit View Query Project Debug Tools Window Help
New Query
BD.POO.FATEC
Execute
Debug
Object Explorer
Connect
(local) (SQL Server 11.0.2100 - sa)
Databases
System Databases
Database Snapshots
BD.POO.FATEC
Database Diagrams
Tables
System Tables
SQLQuery3.sql - (l...POO.FATEC (sa (53))*
USE [BD.POO.FATEC]
GO
INSERT INTO [dbo].[BD_Pais]
([BD_PaisNome]
,[BD_PaisSigla])
VALUES
('Bélgica', 'BEL')
GO

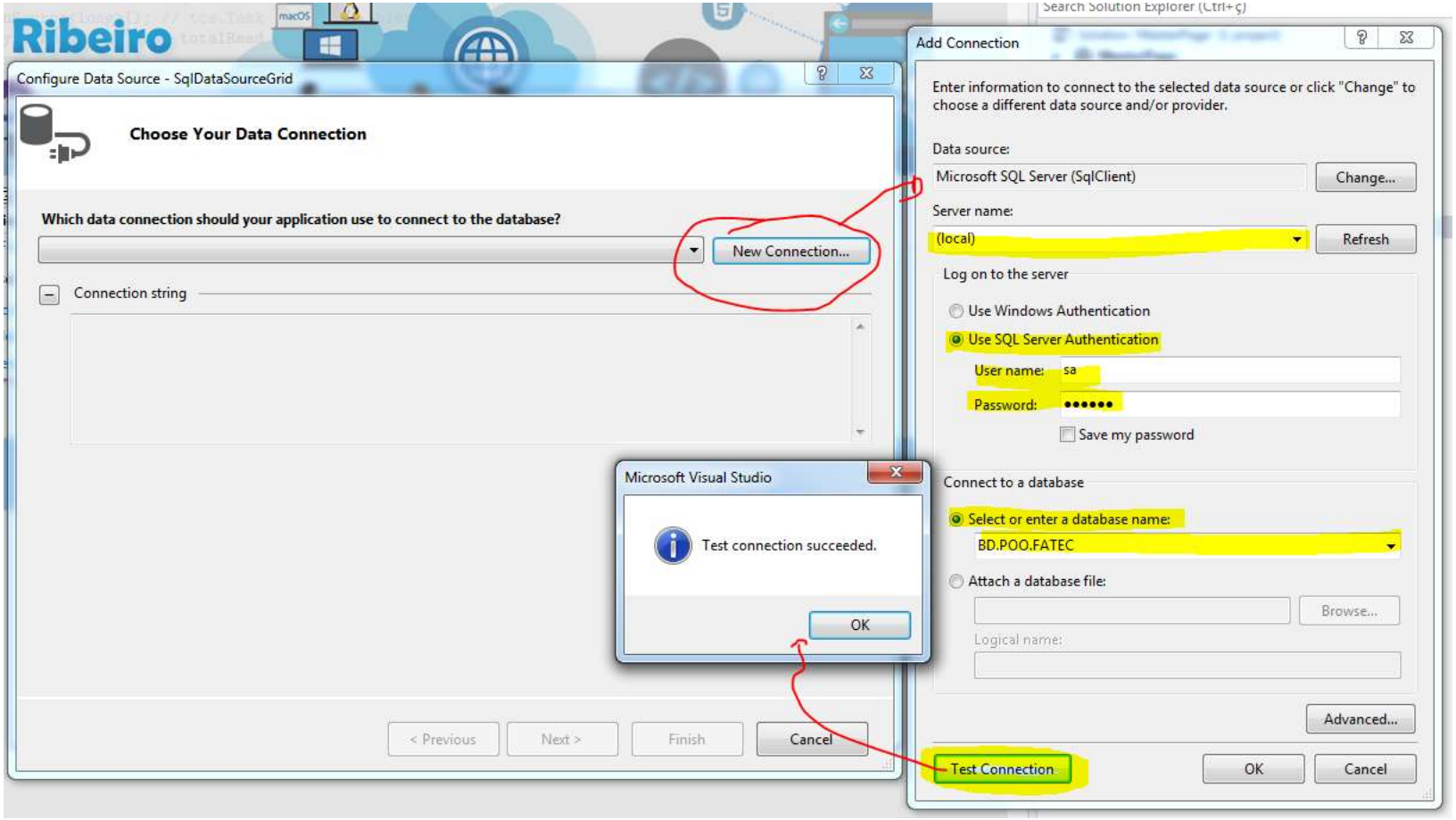
GridView


Toolbox
Search Toolbox
Standard
Data
Pointer
Chart
DataList
DataPager
DetailsView
EntityDataSource
FormView
GridView
LinqDataSource
ListView
ObjectDataSource
QueryExtender
Repeater
SiteMapDataSource
SqlDataSource
XmlDataSource
Validation
Navigation
Login
WebParts
AJAX Extensions
Default2.aspx*
Ajax.aspx
Default.aspx.cs
Default.aspx
MasterPage.master
css.css
Eduardo Ribeiro
Microsoft Visual Studio
ContentPlaceHolder1 (Custom)
asp:gridview#GridView1
Column0 Column1 Column2
abc abc abc
GridView Tasks
Auto Format...
Choose Data Source: (None)
(None)
<New data source...>
Edit Columns...
Add New Column...
Edit Templates
Prof. Ms. Eduardo Ribeiro
Faculdade de Tecnologia "Dep. Júlio Julinho Marcon
2016

GridView

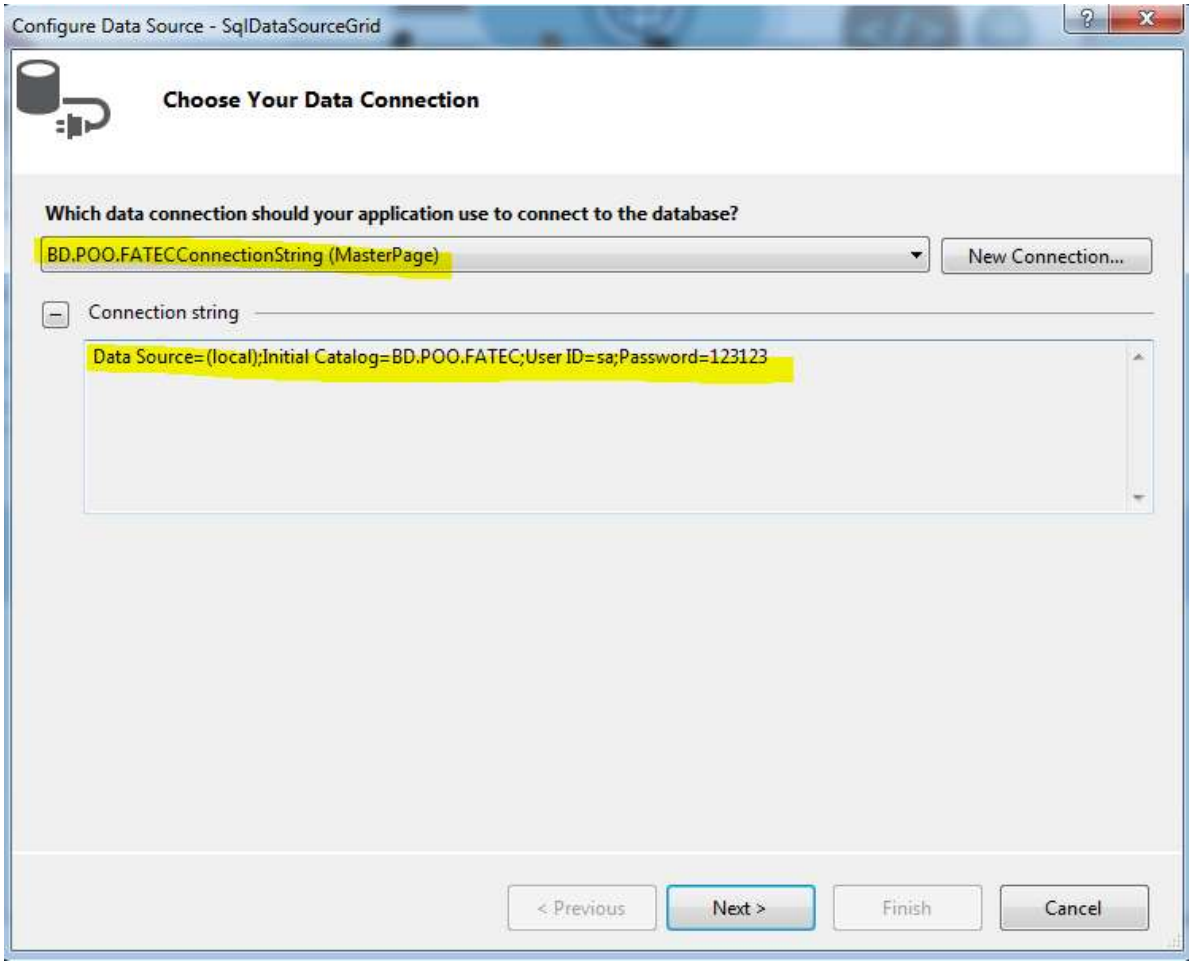

Data Source Configuration Wizard
Choose a Data Source Type
Where will the application get data from?
Database
Entity
LINQ
Object
Site Map
XML File
Connect to any SQL database supported by ADO.NET, such as Microsoft SQL Server, Oracle, or OLEDB.
Specify an ID for the data source:
SqlDataSourceGrid
OK
Cancel

GridView

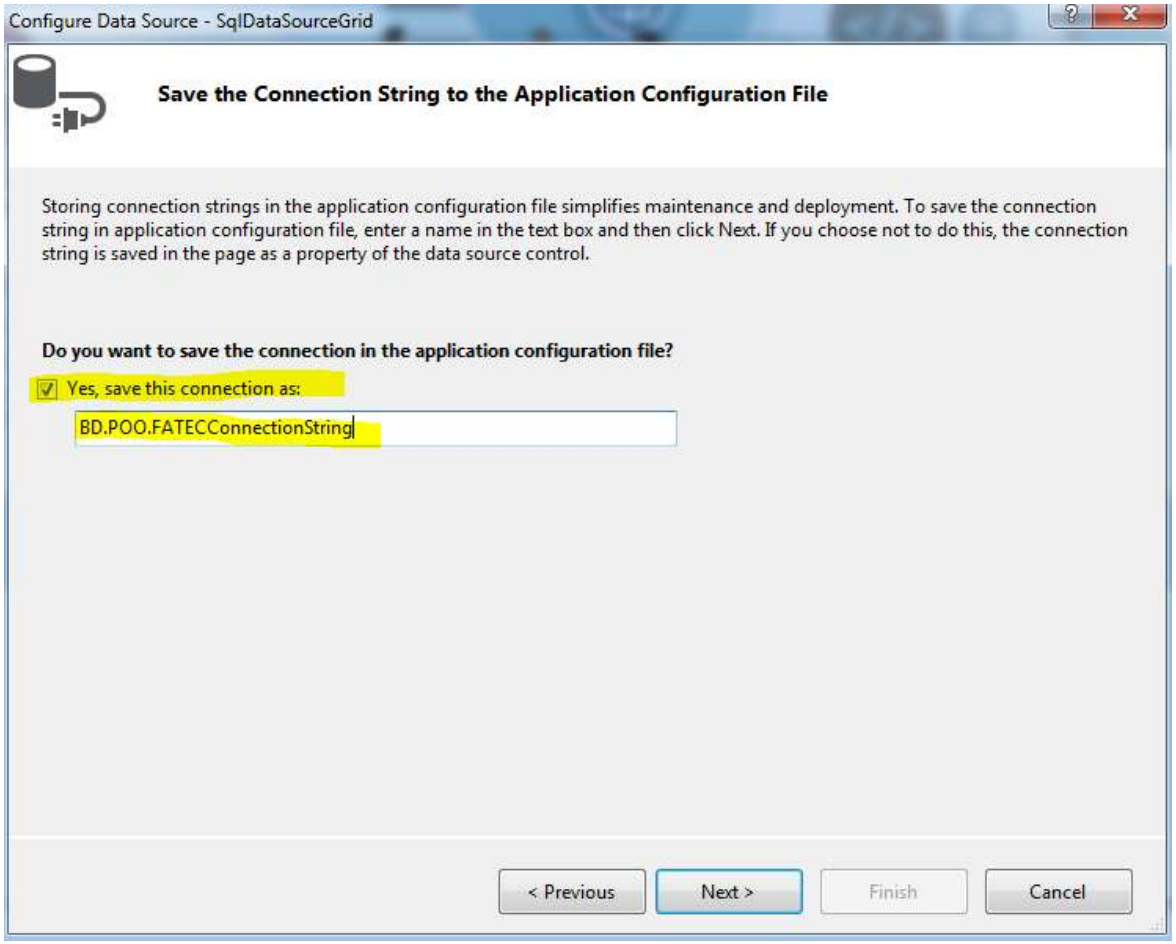


Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

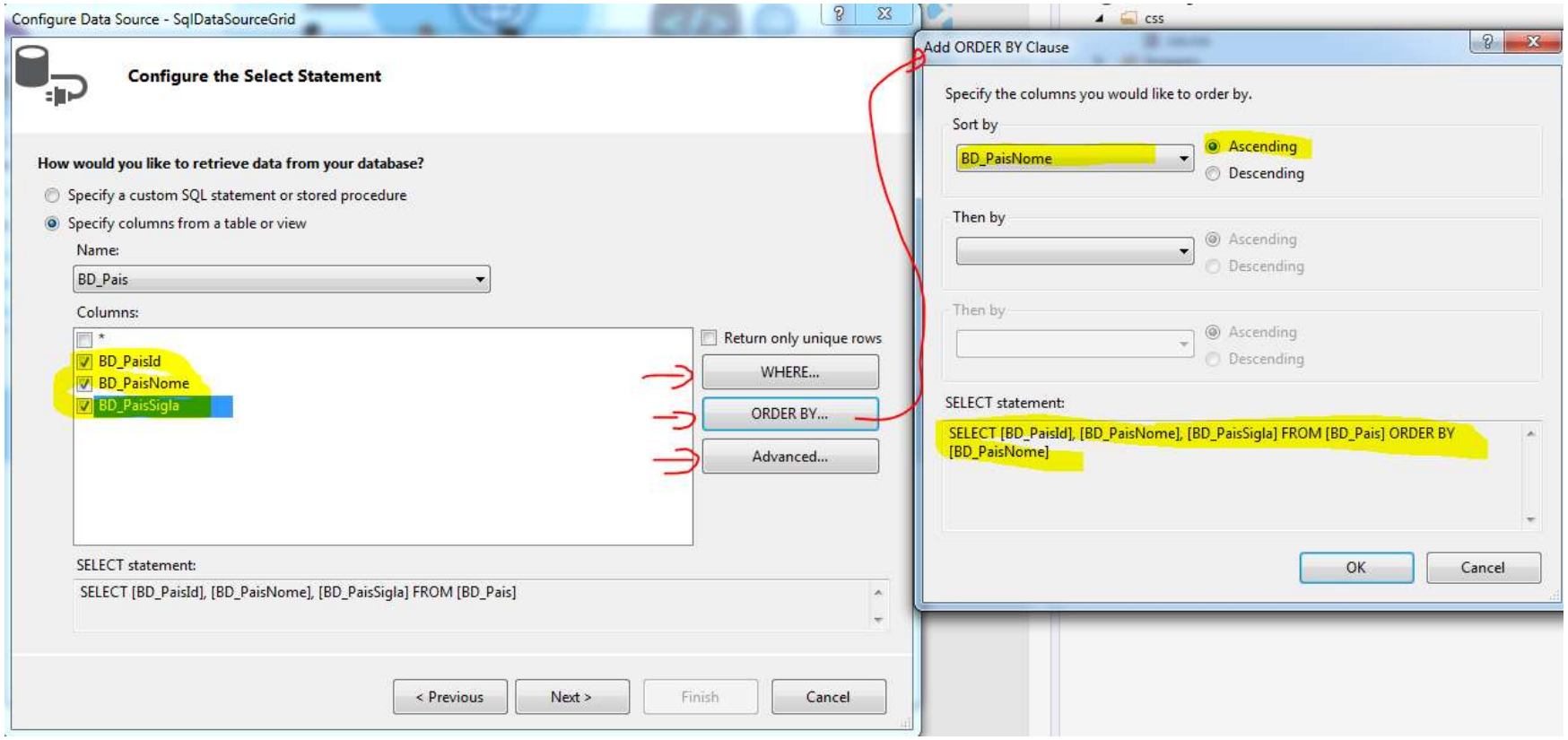
GridView


Configure Data Source - SqlDataSourceGrid
Choose Your Data Connection
Which data connection should your application use to connect to the database?
BD.POO.FATECConnectionString (MasterPage)
New Connection...
Connection string
Data Source=(local);Initial Catalog=BD.POO.FATEC;User ID=sa;Password=123123
< Previous
Next >
Finish
Cancel

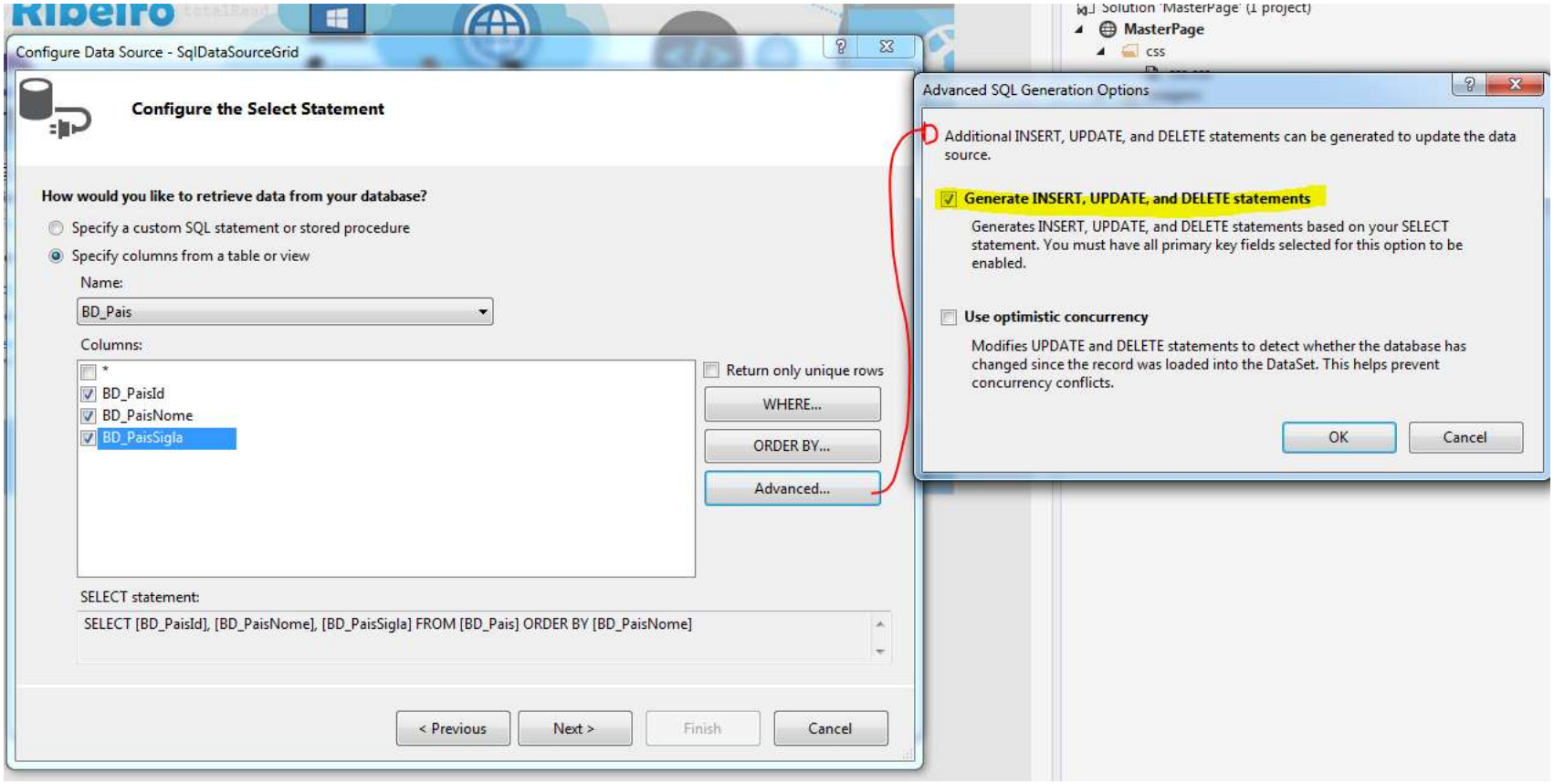
GridView


Configure Data Source - SqlDataSourceGrid
Save the Connection String to the Application Configuration File
Storing connection strings in the application configuration file simplifies maintenance and deployment. To save the connection string in application configuration file, enter a name in the text box and then click Next. If you choose not to do this, the connection string is saved in the page as a property of the data source control.
Do you want to save the connection in the application configuration file?
Yes, save this connection as:
BD.POO.FATECConnectionString
< Previous
Next >
Finish
Cancel

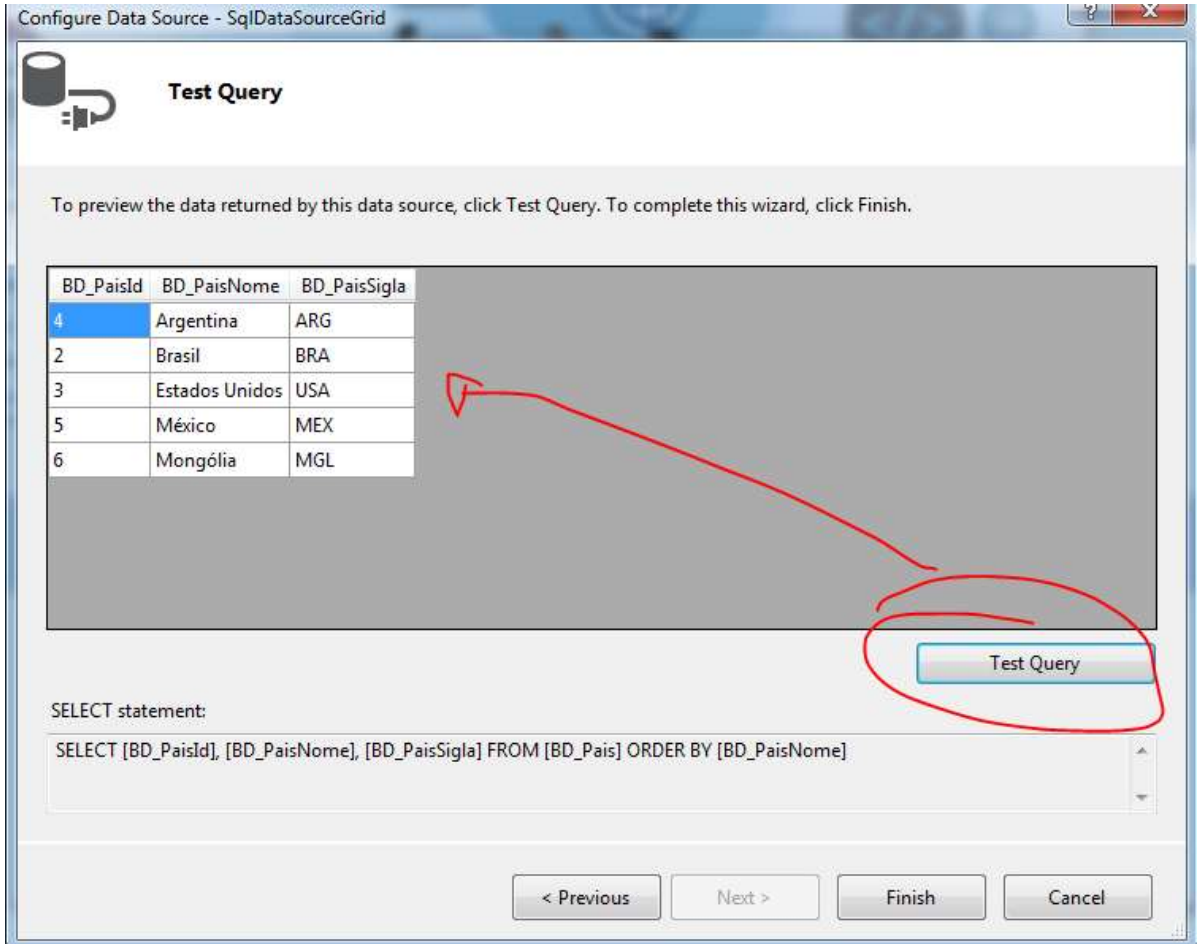
GridView


Configure Data Source - SqlDataSourceGrid
Configure the Select Statement
How would you like to retrieve data from your database?
Specify a custom SQL statement or stored procedure
Specify columns from a table or view
Name:
BD_Pais
Columns:
*
BD_PaisId
BD_PaisNome
BD_PaisSigla
Return only unique rows
WHERE...
ORDER BY...
Advanced...
SELECT statement:
SELECT [BD_PaisId], [BD_PaisNome], [BD_PaisSigla] FROM [BD_Pais]
< Previous
Next >
Finish
Cancel
Add ORDER BY Clause
Specify the columns you would like to order by.
Sort by
BD_PaisNome
Ascending
Descending
Then by
Ascending
Descending
Then by
Ascending
Descending
SELECT statement:
SELECT [BD_PaisId], [BD_PaisNome], [BD_PaisSigla] FROM [BD_Pais] ORDER BY [BD_PaisNome]
OK
Cancel

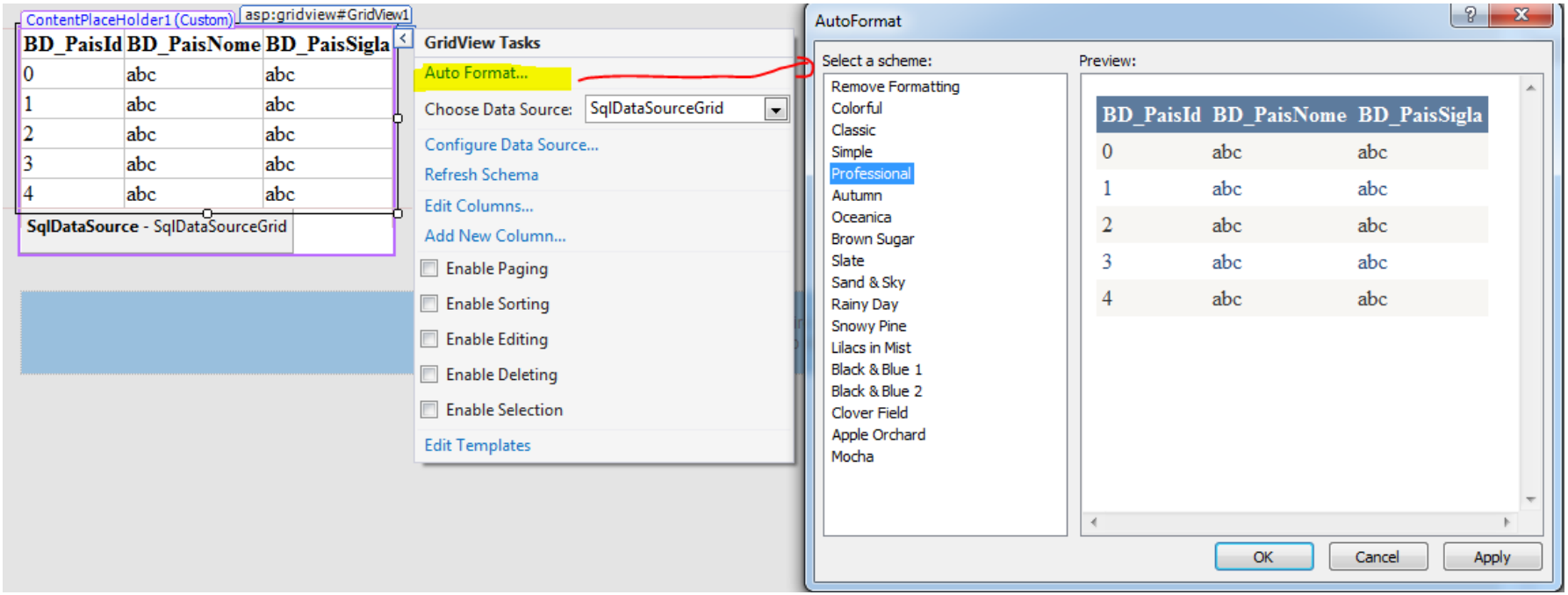
GridView


Configure Data Source - SqlDataSourceGrid
Configure the Select Statement
How would you like to retrieve data from your database?
Specify a custom SQL statement or stored procedure
Specify columns from a table or view
Name:
BD_Pais
Columns:
*
BD_PaisId
BD_PaisNome
BD_PaisSigla
Return only unique rows
WHERE...
ORDER BY...
Advanced...
SELECT statement:
SELECT [BD_PaisId], [BD_PaisNome], [BD_PaisSigla] FROM [BD_Pais] ORDER BY [BD_PaisNome]
< Previous
Next >
Finish
Cancel
Solution 'MasterPage' (1 project)
MasterPage
css
Advanced SQL Generation Options
Additional INSERT, UPDATE, and DELETE statements can be generated to update the data source.
Generate INSERT, UPDATE, and DELETE statements
Generates INSERT, UPDATE, and DELETE statements based on your SELECT statement. You must have all primary key fields selected for this option to be enabled.
Use optimistic concurrency
Modifies UPDATE and DELETE statements to detect whether the database has changed since the record was loaded into the DataSet. This helps prevent concurrency conflicts.
OK
Cancel

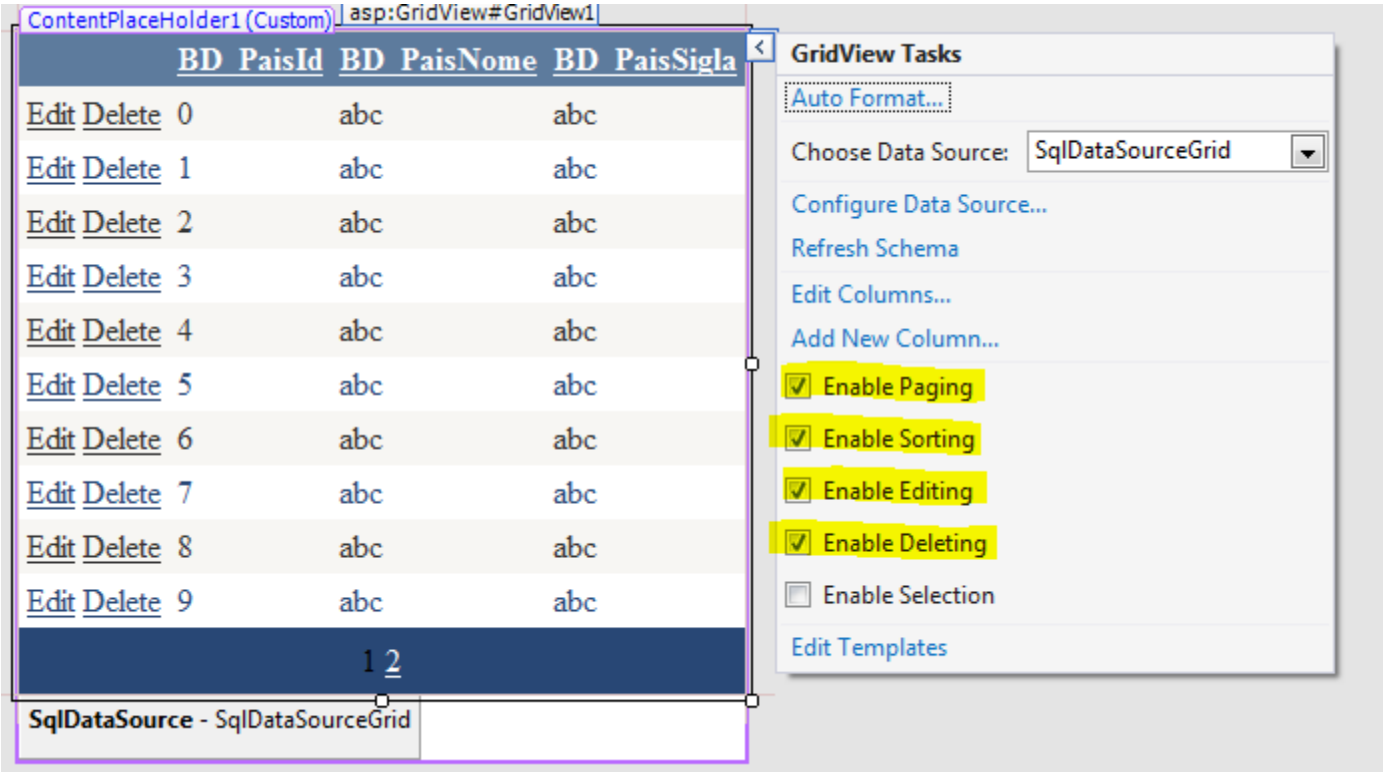
GridView


Configure Data Source - SqlDataSourceGrid
Test Query
To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.
BD_PaisId BD_PaisNome BD_PaisSigla
4 Argentina ARG
2 Brasil BRA
3 Estados Unidos USA
5 México MEX
6 Mongólia MGL
Test Query
SELECT statement:
SELECT [BD_PaisId], [BD_PaisNome], [BD_PaisSigla] FROM [BD_Pais] ORDER BY [BD_PaisNome]
< Previous
Next >
Finish
Cancel

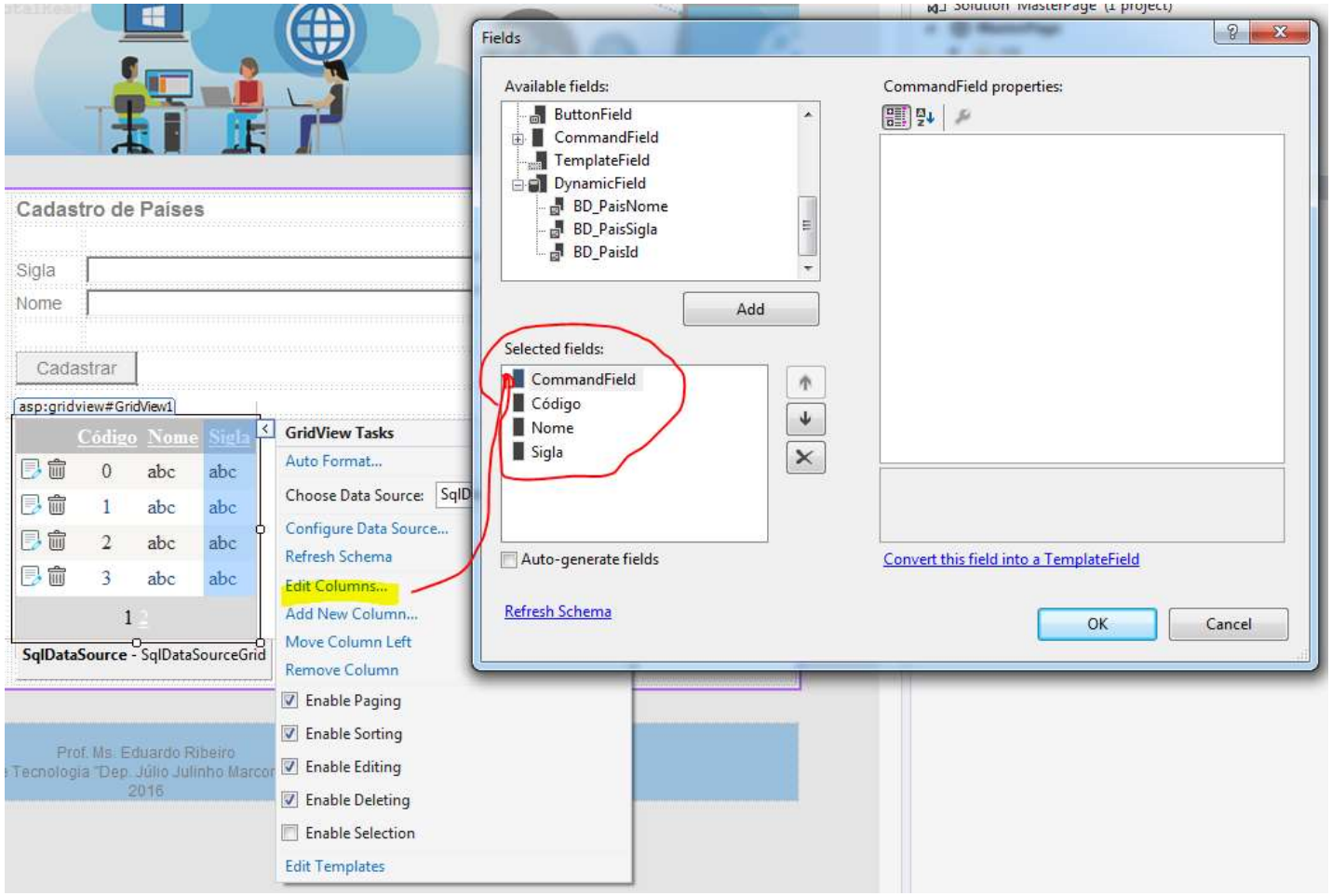
GridView


ContentPlaceHolder1 (Custom)
asp:gridview#GridView1
BD_PaisId BD_PaisNome BD_PaisSigla
0 abc abc
1 abc abc
2 abc abc
3 abc abc
4 abc abc
SqlDataSource - SqlDataSourceGrid
GridView Tasks
Auto Format...
Choose Data Source: SqlDataSourceGrid
Configure Data Source...
Refresh Schema
Edit Columns...
Add New Column...
Enable Paging
Enable Sorting
Enable Editing
Enable Deleting
Enable Selection
Edit Templates
AutoFormat
Select a scheme:
Remove Formatting
Colorful
Classic
Simple
Professional
Autumn
Oceanica
Brown Sugar
Slate
Sand & Sky
Rainy Day
Snowy Pine
Lilacs in Mist
Black & Blue 1
Black & Blue 2
Clover Field
Apple Orchard
Mocha
Preview:
BD_PaisId BD_PaisNome BD_PaisSigla
0 abc abc
1 abc abc
2 abc abc
3 abc abc
4 abc abc
OK
Cancel
Apply

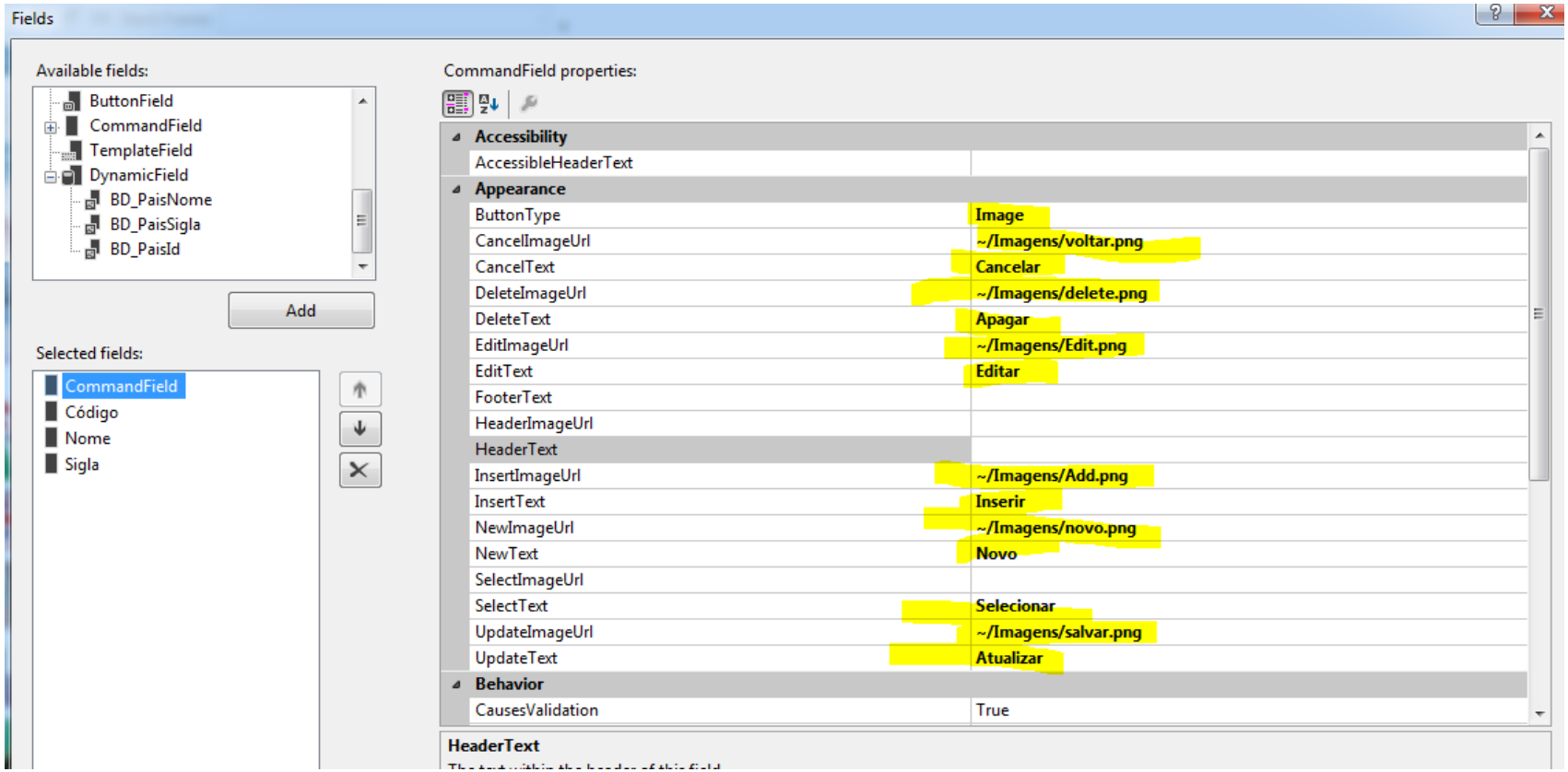
GridView


ContentPlaceHolder1 (Custom)
asp:GridView#GridView1
BD_PaisId
BD_PaisNome
BD_PaisSigla
Edit Delete 0 abc abc
Edit Delete 1 abc abc
Edit Delete 2 abc abc
Edit Delete 3 abc abc
Edit Delete 4 abc abc
Edit Delete 5 abc abc
Edit Delete 6 abc abc
Edit Delete 7 abc abc
Edit Delete 8 abc abc
Edit Delete 9 abc abc
1 2
SqlDataSource - SqlDataSourceGrid
GridView Tasks
Auto Format...
Choose Data Source:
SqlDataSourceGrid
Configure Data Source...
Refresh Schema
Edit Columns...
Add New Column...
Enable Paging
Enable Sorting
Enable Editing
Enable Deleting
Enable Selection
Edit Templates

GridView


Cadastro de Países
Sigla
Nome
Cadastrar
asp:gridview#GridView1
Código Nome Sigla
0 abc abc
1 abc abc
2 abc abc
3 abc abc
1 2
SqlDataSource - SqlDataSourceGrid
GridView Tasks
Auto Format...
Choose Data Source:
Configure Data Source...
Refresh Schema
Edit Columns...
Add New Column...
Move Column Left
Remove Column
Enable Paging
Enable Sorting
Enable Editing
Enable Deleting
Enable Selection
Edit Templates
Fields
Available fields:
ButtonField
CommandField
TemplateField
DynamicField
BD_PaisNome
BD_PaisSigla
BD_PaisId
Add
Selected fields:
CommandField
Código
Nome
Sigla
Auto-generate fields
Refresh Schema
CommandField properties:
Convert this field into a TemplateField
OK
Cancel
Prof. Ms. Eduardo Ribeiro
2016

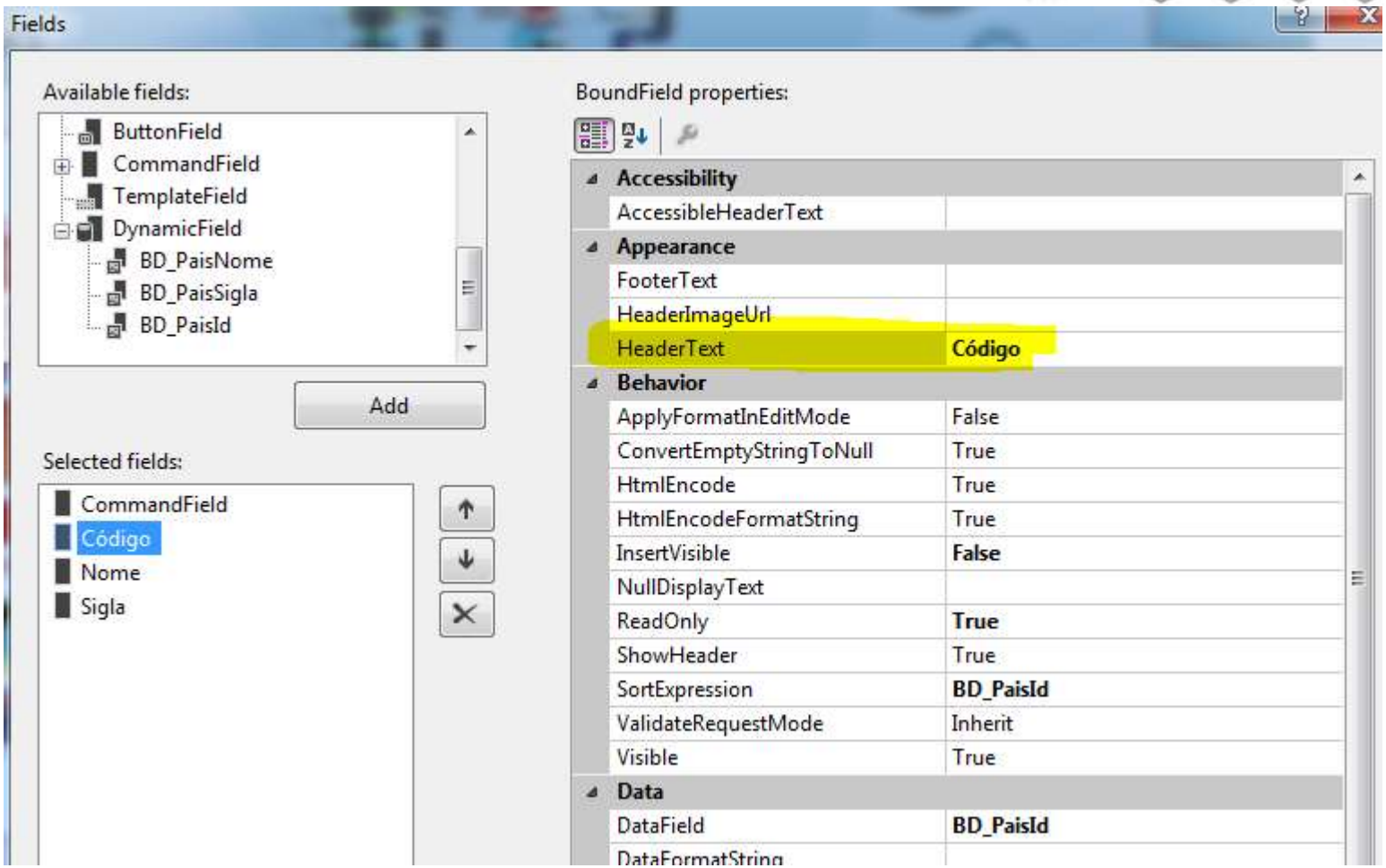
GridView



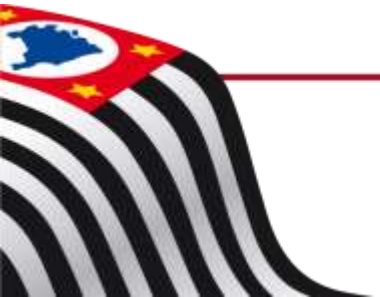
Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

GridView


Fields
Available fields:
ButtonField
CommandField
TemplateField
DynamicField
BD_PaisNome
BD_PaisSigla
BD_PaisId
Add
Selected fields:
CommandField
Código
Nome
Sigla
BoundField properties:
Accessibility
AccessibleHeaderText
Appearance
FooterText
HeaderImageUrl
HeaderText
Código
Behavior
ApplyFormatInEditMode
False
ConvertEmptyStringToNull
True
HtmlEncode
True
HtmlEncodeFormatString
True
InsertVisible
False
NullDisplayText
ReadOnly
True
ShowHeader
True
SortExpression
BD_PaisId
ValidateRequestMode
Inherit
Visible
True
Data
DataField
BD_PaisId

Capítulo 16b: ASP.NET



GridView

Cadastro de Países

Sigla

Nome

Cadastrar

Nome País Brasil Buscar Limpar

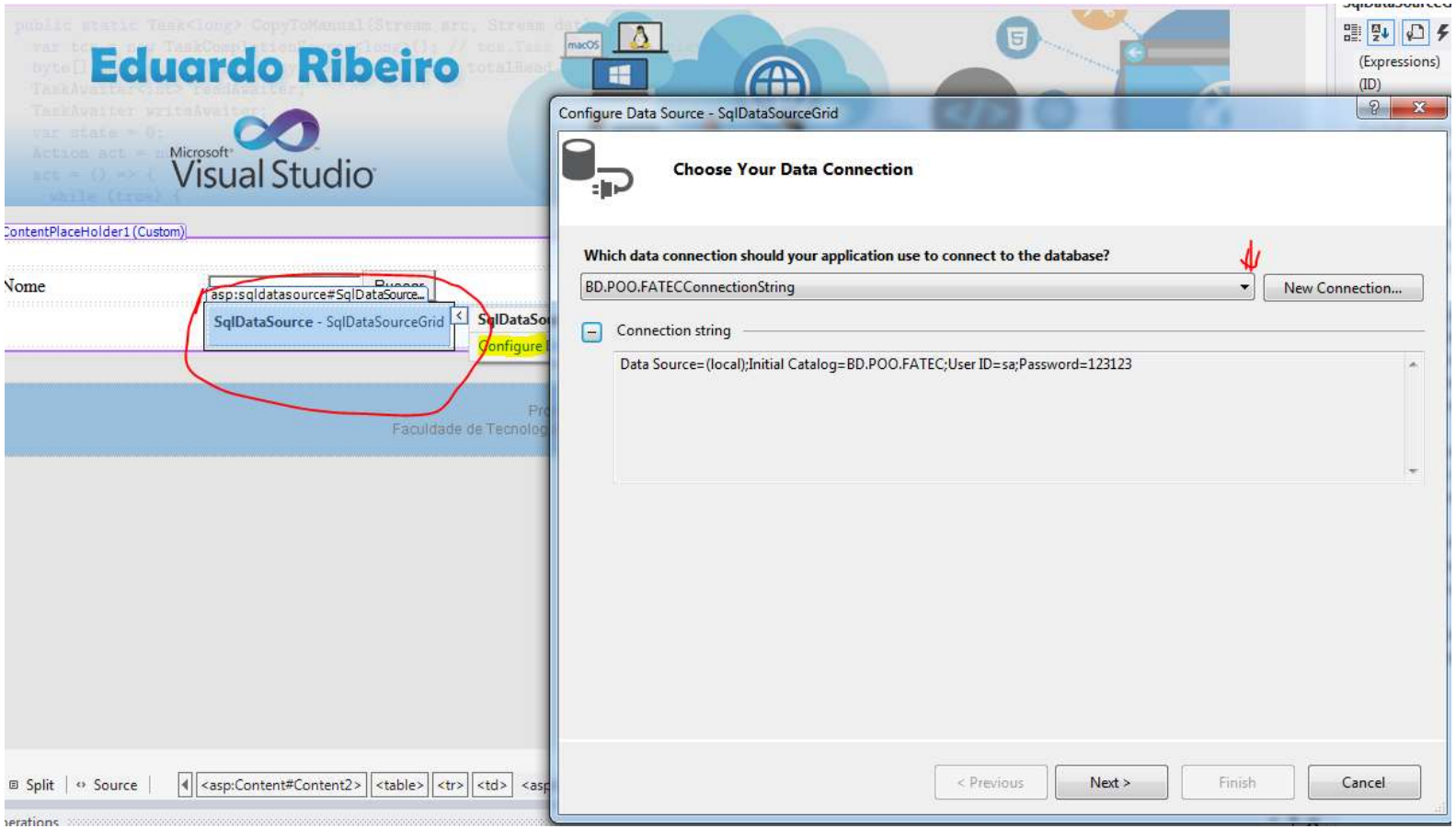
	Código	Nome	Sigla
	2	Brasil	BRA

Prof. Ms. Eduardo Ribeiro
Faculdade de Tecnologia "Dep. Júlio Julinho Marcondes de Moura"
2016

Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

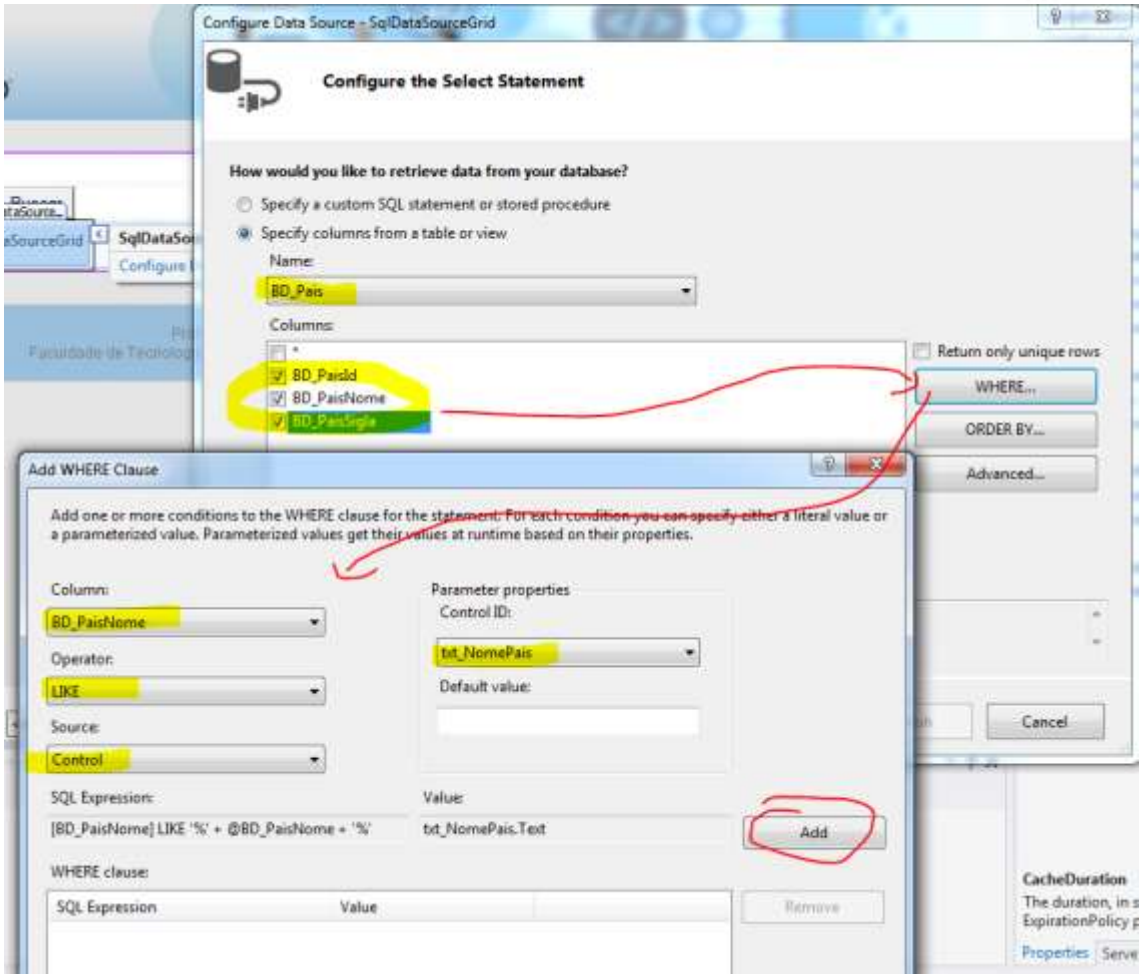
GridView



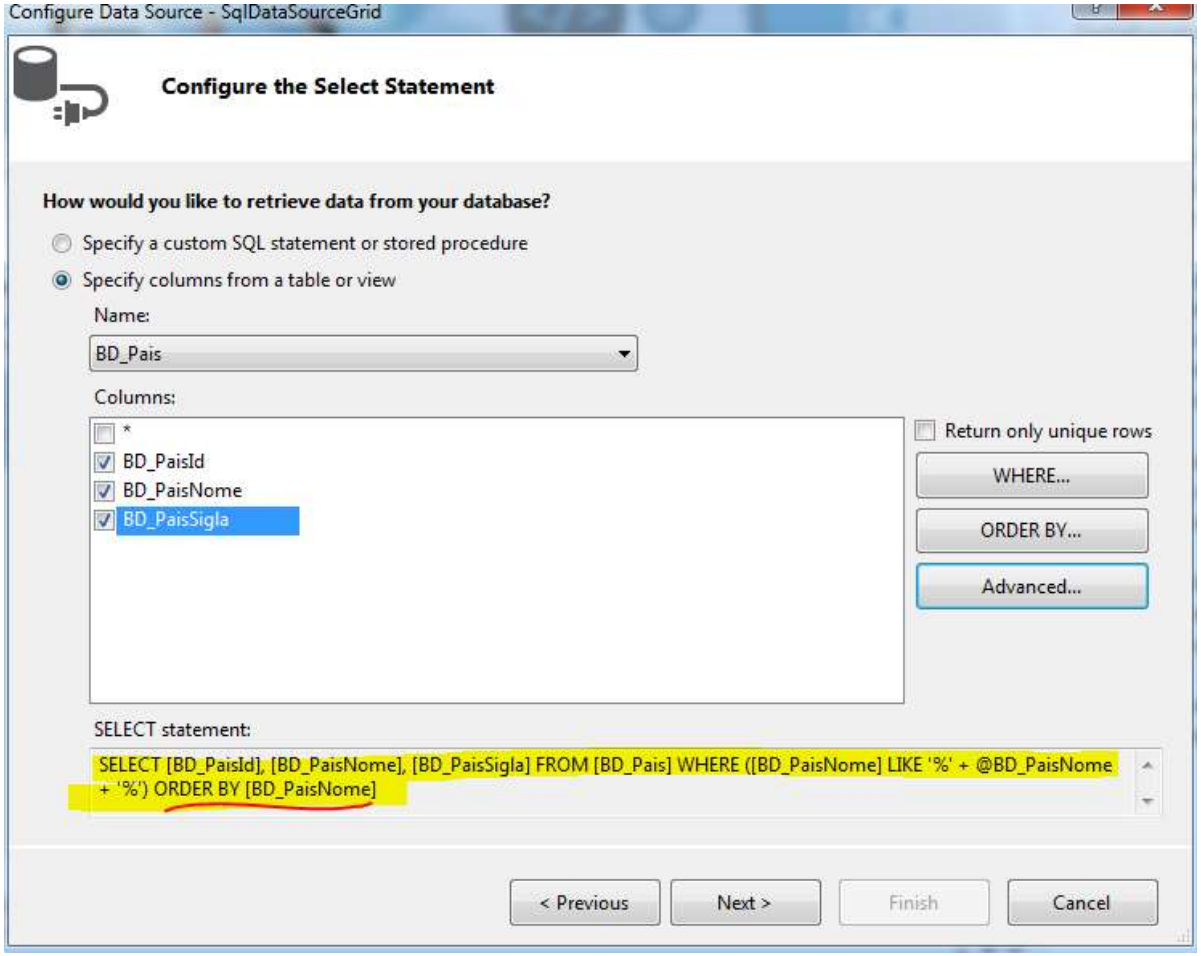
Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

GridView

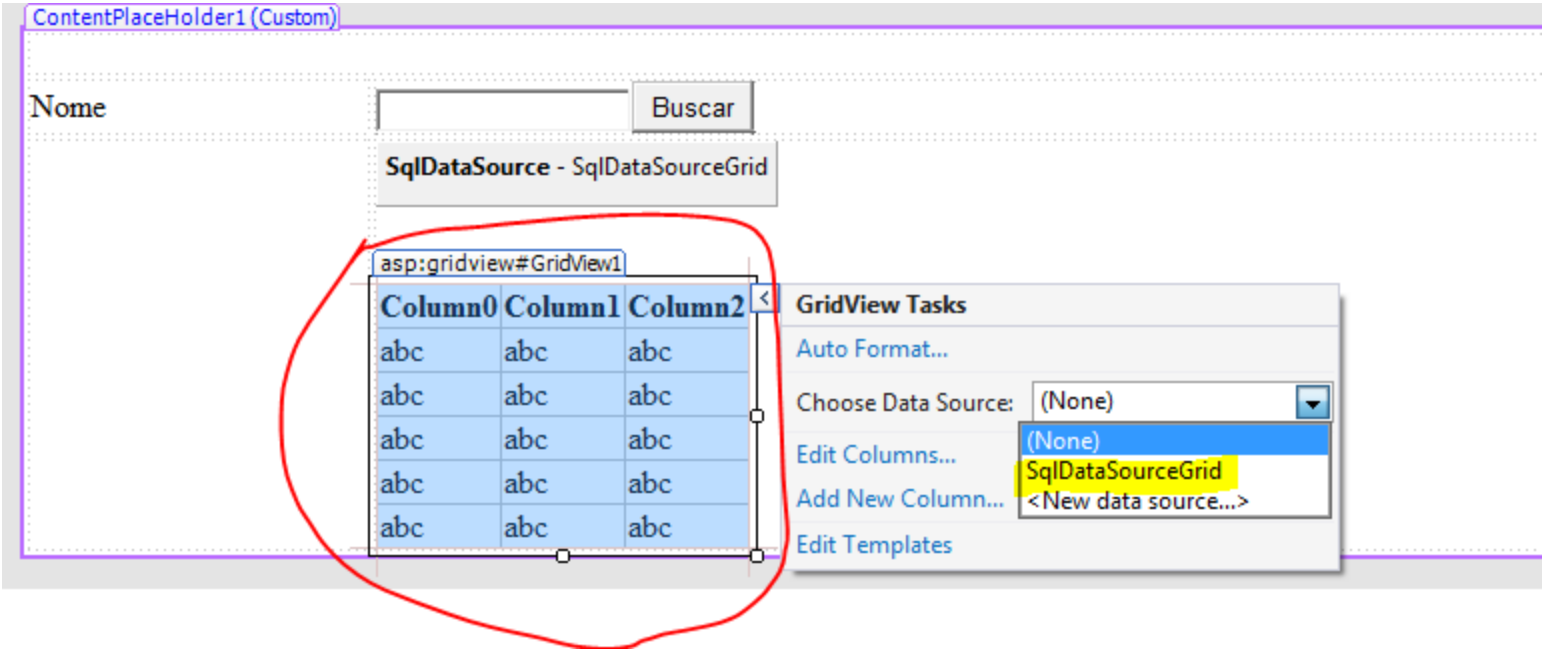

Configure Data Source - SqlDataSourceGrid
Configure the Select Statement
How would you like to retrieve data from your database?
Specify a custom SQL statement or stored procedure
Specify columns from a table or view
Name:
BD_Pais
Columns:
*
BD_PaisId
BD_PaisNome
BD_PaisSigla
Return only unique rows
WHERE...
ORDER BY...
Advanced...
Cancel
Add WHERE Clause
Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.
Column:
BD_PaisNome
Operator:
LIKE
Source:
Control
Parameter properties
Control ID:
txt_NomePais
Default value:
SQL Expression:
[BD_PaisNome] LIKE '%' + @BD_PaisNome + '%'
Value:
txt_NomePais.Text
Add
WHERE clause:
SQL Expression
Value
Remove
CacheDuration
Properties

GridView

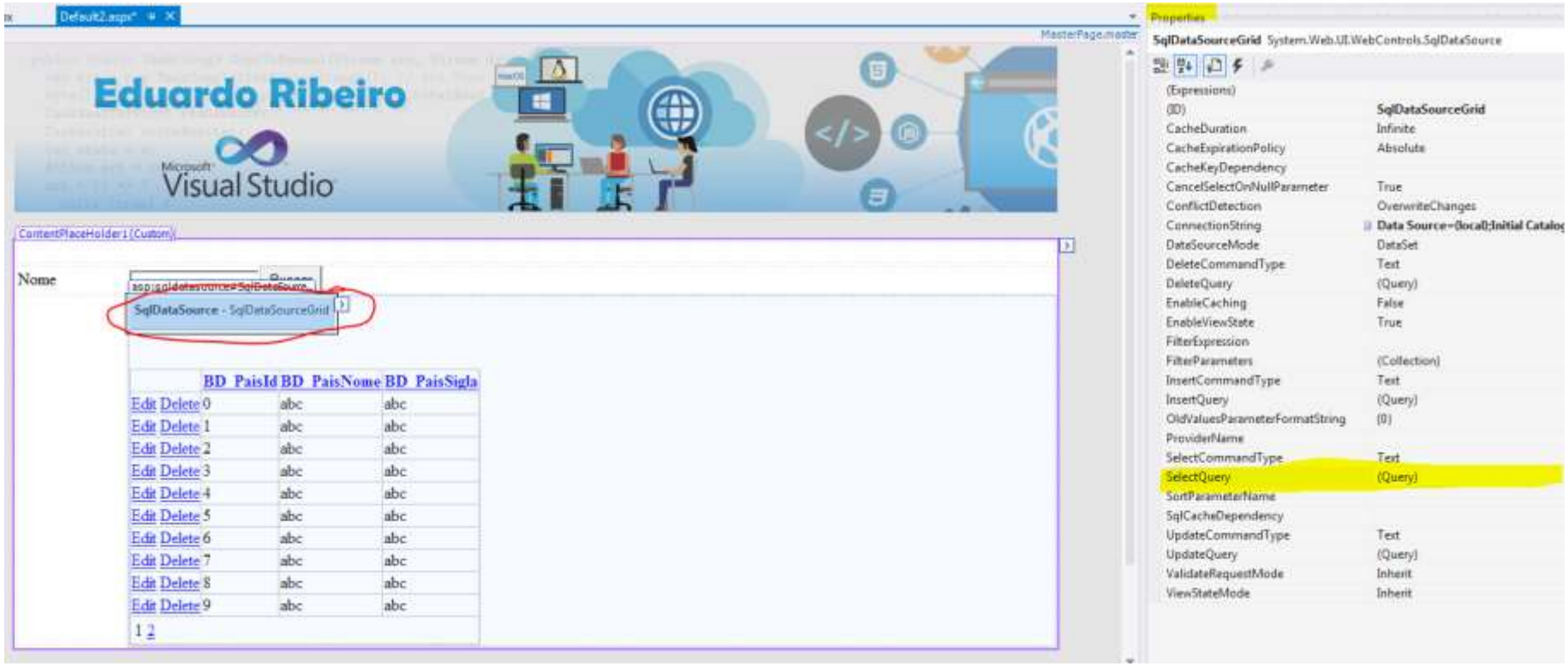


Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

GridView


ContentPlaceHolder1 (Custom)
Nome
Buscar
SqlDataSource - SqlDataSourceGrid
asp:gridview#GridView1
Column0 Column1 Column2
abc abc abc
abc abc abc
abc abc abc
abc abc abc
abc abc abc
GridView Tasks
Auto Format...
Choose Data Source:
(None)
(None)
SqlDataSourceGrid
<New data source...>
Edit Columns...
Add New Column...
Edit Templates

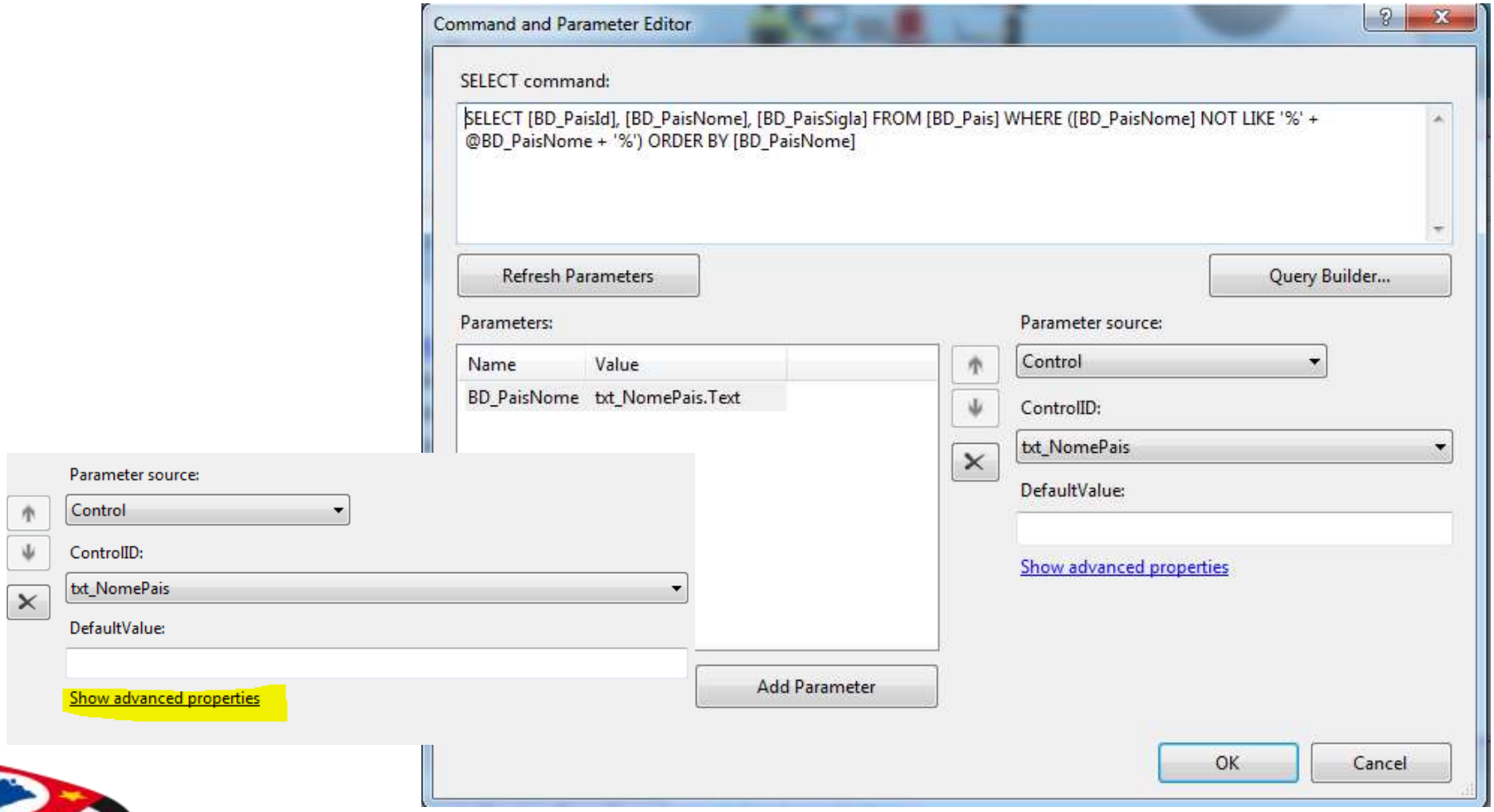
GridView



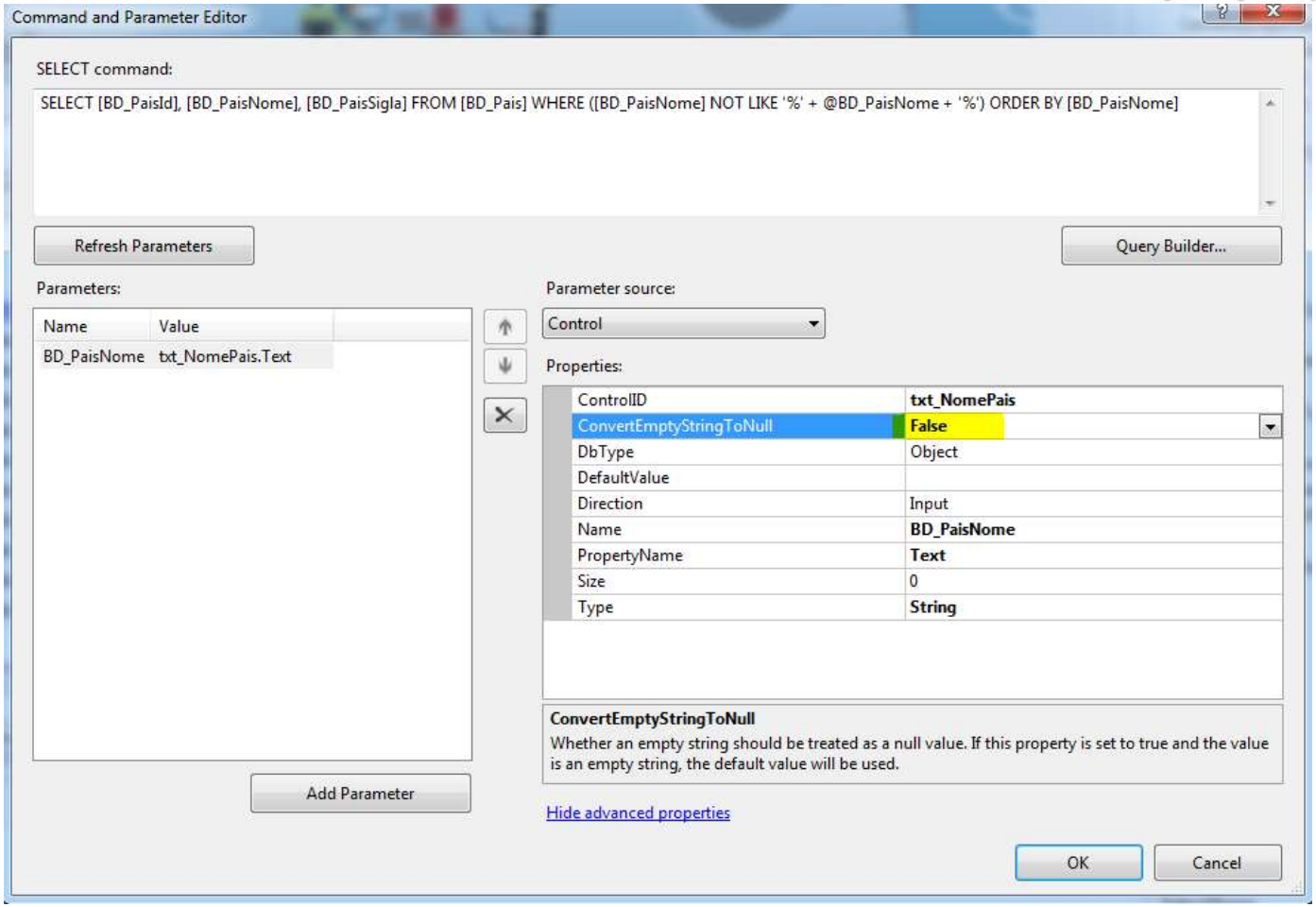
Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

GridView


Command and Parameter Editor
SELECT command:
SELECT [BD_PaisId], [BD_PaisNome], [BD_PaisSigla] FROM [BD_Pais] WHERE ([BD_PaisNome] NOT LIKE '%' + @BD_PaisNome + '%') ORDER BY [BD_PaisNome]
Refresh Parameters
Query Builder...
Parameters:
Name
Value
BD_PaisNome
txt_NomePais.Text
Parameter source:
Control
ControlID:
txt_NomePais
DefaultValue:
Show advanced properties
Add Parameter
OK
Cancel
Parameter source:
Control
ControlID:
txt_NomePais
DefaultValue:
Show advanced properties

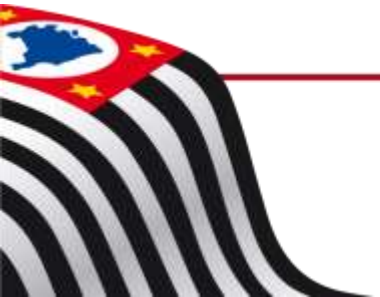
GridView



Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

Capítulo 17: ASP.NET



Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

ASP.NET

Cadastro de Países

Sigla

Nome

[lbl_Error]

Cadastrar

Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

Insert (1)

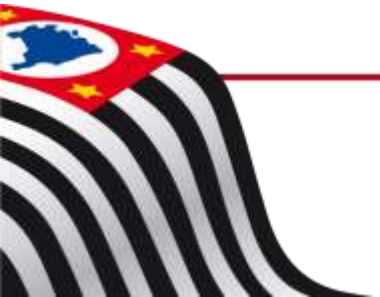
```
4   using System.Web;
5   using System.Web.UI;
6   using System.Web.UI.WebControls;
7
8   //Acesso a Dados
9   using System.Data;
10  using System.Data.SqlClient;
11
12  public partial class _Default : System.Web.UI.Page
13  {
14      string ConStr = @"Data Source=(local);Initial Catalog=BD.POO.FATEC;User ID=sa;Password=123123;";
15
16      protected void Page_Load(object sender, EventArgs e)
17      {
18
19      }
20      protected void Bt_Cadastrar_Click(object sender, EventArgs e)
21      {
22          SqlConnection con = new SqlConnection(ConStr);
23          con.Open();
24          try
25          {
26              string Sql = "insert into BD_Pais(BD_PaisNome, BD_PaisSigla) Values (@txt_NomePais, @txt_Sigla)";
27              SqlCommand cmd = new SqlCommand(Sql,con);
```

Insert (1)

```
28
29              cmd.Parameters.AddWithValue("@txt_NomePais", txt_NomePais.Text);
30              cmd.Parameters.AddWithValue("@txt_Sigla", txt_Sigla.Text);
31              cmd.ExecuteNonQuery();
32
33              GridViewPais.DataBind();
34
35              txt_NomePais.Text   = "";
36              txt_Sigla.Text      = "";
37          }
38          finally
39          {
40              con.Close();
41          }
42      }
43      protected void Bt_Limpar_Click(object sender, EventArgs e)
44      {
45          txt_Buscar.Text = "";
46      }
47  }
```

Capítulo 18: ASP.NET

SQL Server – Stored Procedure



Insert (2)

Stored Procedure

File Edit View Query Project Debug Tools Window Help

New Query | BD.POO.FATEC | Execute | Debug

Object Explorer

Connect

- EDUARDO-DELL (SQL Server 11.0.2100 - sa)
  - Databases
    - System Databases
    - Database Snapshots
    - BD.POO.FATEC
      - Database Diagrams
      - Tables
      - Views
      - Synonyms
      - Programmability
        - Stored Procedures
          - System Stored Procedures
          - dbo.sp_Pais
        - Functions
        - Database Triggers
        - Assemblies
        - Types

SQLQuery2.sql - ED...OO.FATEC (sa (53))*

```
Create procedure [dbo].[sp_Pais]
@NomePais varchar(80),
@SiglaPais  char(3)
as
begin
    insert into [dbo].[BD_Pais]([BD_PaisNome],[BD_PaisSigla])
    values(@NomePais, @SiglaPais);
end
```

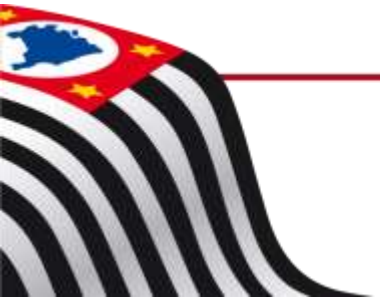
Insert (2)

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.UI;
6  using System.Web.UI.WebControls;
7
8  //Acesso a Dados
9  using System.Data;
10 using System.Data.SqlClient;
11
12 public partial class _Default : System.Web.UI.Page
13 {
14     string ConStr = @"Data Source=(local);Initial Catalog=BD.POO.FATEC;User ID=sa;Password=123123;";
15
16     protected void Page_Load(object sender, EventArgs e)
17     {
18
19     }
20     protected void Bt_Cadastrar_Click(object sender, EventArgs e)
21     {
22         SqlConnection con = new SqlConnection(ConStr);
23         con.Open();
24         try
25         {
26             //string Sql = "insert into BD_Pais(BD_PaisNome, BD_PaisSigla) Values (@txt_NomePais, @txt_Sigla)";
27             //SqlCommand cmd = new SqlCommand(Sql,con);
28
29             SqlCommand cmd = new SqlCommand("sp_Pais", con);
30             cmd.CommandType = CommandType.StoredProcedure;
```

Insert (2)

```
32              cmd.Parameters.AddWithValue("@NomePais", txt_NomePais.Text);
33              cmd.Parameters.AddWithValue("@SiglaPais", txt_Sigla.Text);
34              cmd.ExecuteNonQuery();
35
36              GridViewPais.DataBind();
37
38              txt_NomePais.Text   = "";
39              txt_Sigla.Text      = "";
40          }
41          finally
42          {
43              con.Close();
44          }
45      }
46      protected void Bt_Limpar_Click(object sender, EventArgs e)
47      {
48          txt_Buscar.Text = "";
49      }
50  }
```

Capítulo 19: ASP.NET



Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

ConnectionStrings

Web.config

Web.config | Default.aspx | Default.aspx.cs

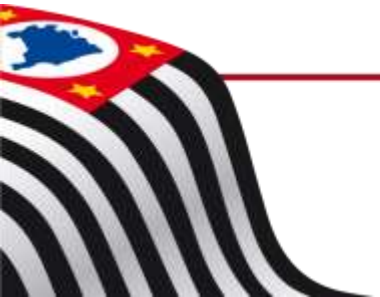
```xml
<?xml version="1.0"?>

<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
  -->

<configuration>

    <connectionStrings>
        <add name="BD.POO.FATECConnectionString" connectionString="Data Source=(local);Initial Catalog=BD.POO.FATEC;User ID=sa;Password=123123"
            providerName="System.Data.SqlClient" />
    </connectionStrings>

    <system.web>
      <compilation debug="true" targetFramework="4.5" />
      <httpRuntime targetFramework="4.5" />
    </system.web>

</configuration>
```

ConnectionStrings

Web.config

```
6   using System.Web.UI.WebControls;
7
8   //Acesso a Dados
9   using System.Data;
10  using System.Data.SqlClient;
11
12  using System.Configuration;
13
14  public partial class _Default : System.Web.UI.Page
15  {
16      //string ConStr = @"Data Source=(local);Initial Catalog=BD.POO.FATEC;User ID=sa;Password=123123;";
17      String ConStr = ConfigurationManager.ConnectionStrings["BD.POO.FATECConnectionString"].ConnectionString;
18
19      protected void Page_Load(object sender, EventArgs e)
20      {
21
```

Capítulo 20: ASP.NET

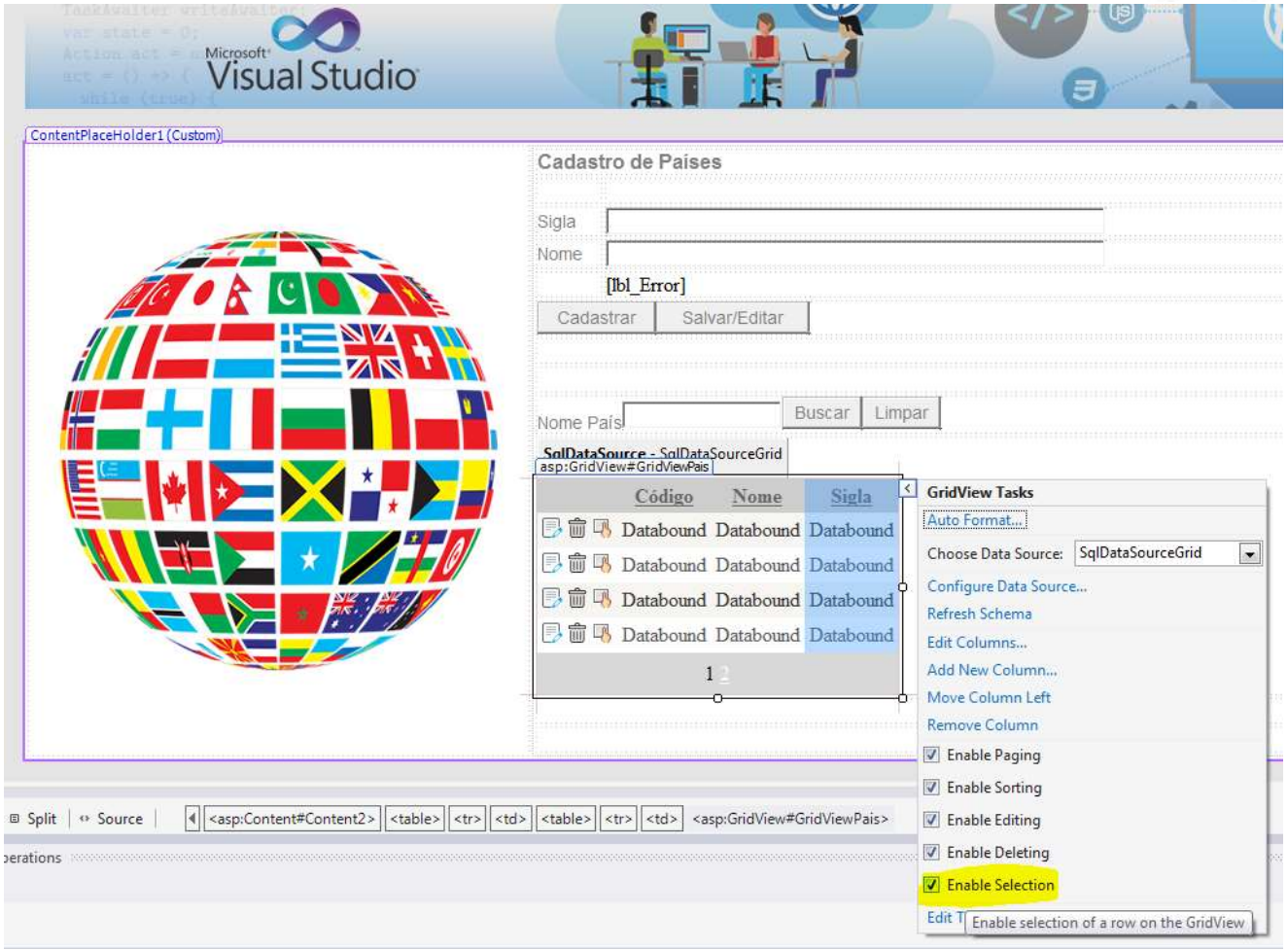


Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

UPDATE (2)

Cadastro de Países
Sigla
BEL
Nome
Bélgica
Cadastrar
Salvar/Editar
Nome País
Buscar
Limpar
Código
Nome
Sigla
17
Áustria
AUT
7
Bélgica
BEL
2
Brasil
BRA
3
Estados Unidos
USA
1 2

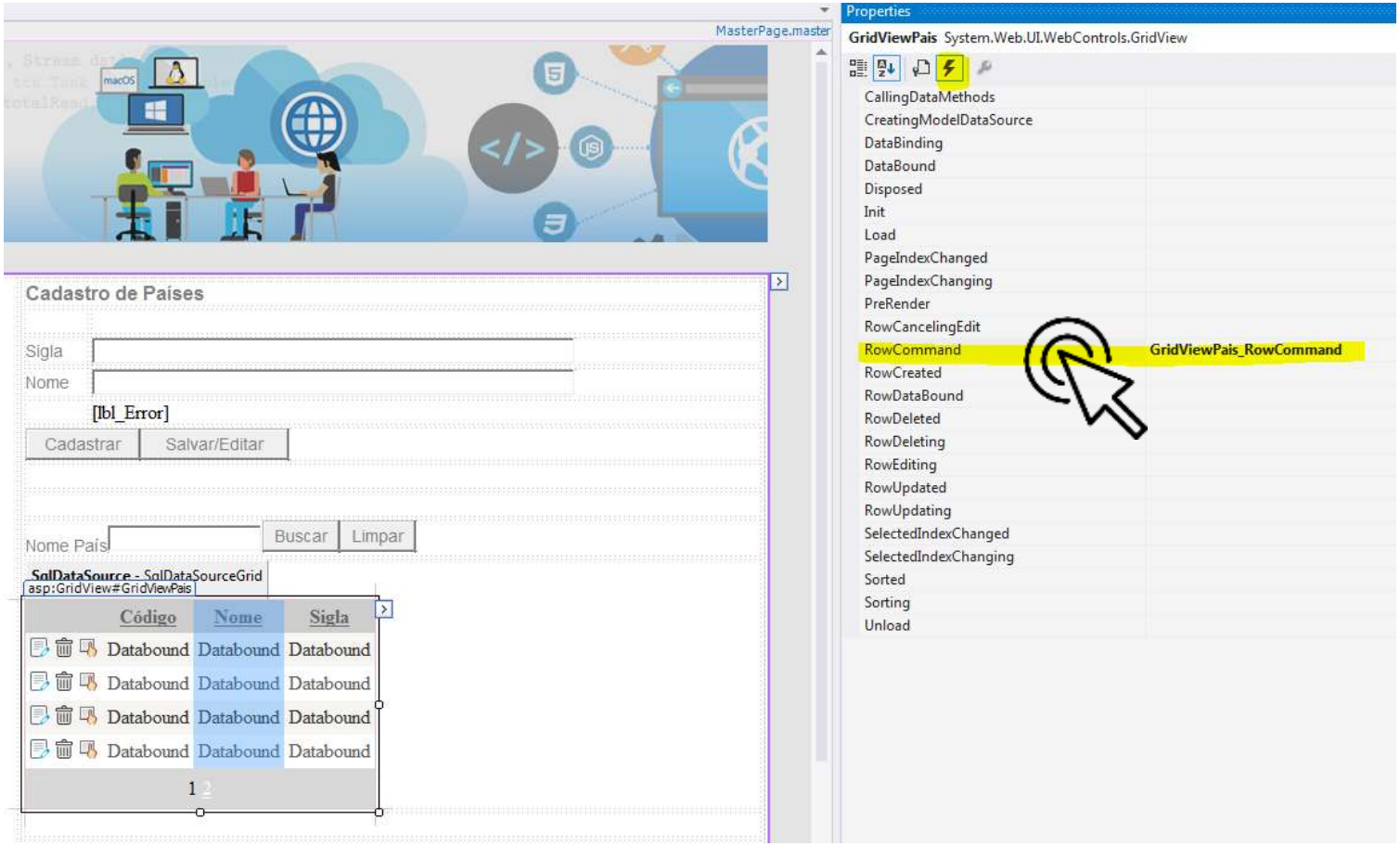
UPDATE (2)



Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

UPDATE (2)



Centro Paula Souza

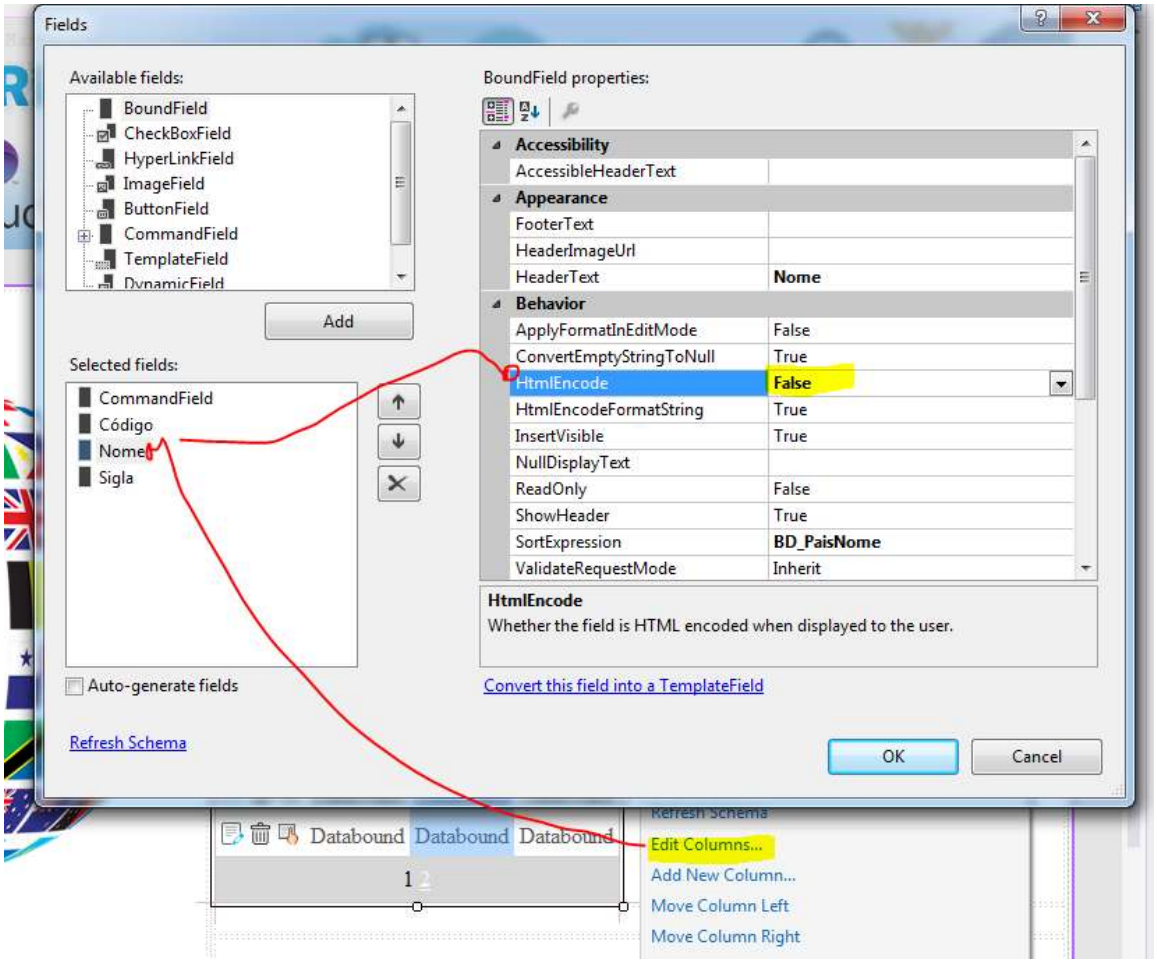
GOVERNO DO ESTADO SÃO PAULO

UPDATE (2)

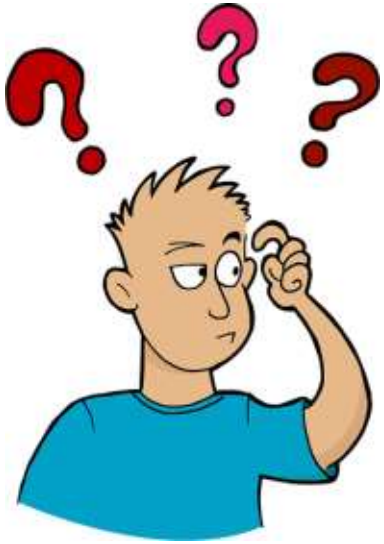
```
protected void GridViewPais_RowCommand(object sender, GridViewCommandEventArgs e)
{
    if (e.CommandName == "Select")
    {
        Int16 num = Convert.ToInt16(e.CommandArgument);
        //txt_NomePais.Text = GridViewPais.Rows[num].Cells[2].Text;
        txt_NomePais.Text = HttpUtility.HtmlDecode(GridViewPais.Rows[num].Cells[2].Text);
        txt_Sigla.Text = GridViewPais.Rows[num].Cells[3].Text;
    }
}
```

UPDATE (2)

```
//txt_NomePais.Text = GridViewPais.Rows[num].Cells[2].Text;
```



HTMLEncode



O **HTMLEncode** ajuda a prevenir ataques de Cross-site scripting (XSS) consiste em uma vulnerabilidade causada pela falha nas validações dos parâmetros de entrada do usuário e resposta do servidor na aplicação web. Este ataque permite que código HTML seja inserido de maneira arbitrária no navegador do usuário alvo. Esta propriedade deve ser Habilitada sempre que possível.

https://msdn.microsoft.com/pt-br/library/system.web.ui.webcontrols.boundfield.htmlencode.aspx

http://www.redesegura.com.br/2012/01/saiba-mais-sobre-o-cross-site-scripting-xss/

UPDATE (2)

localhost:64772/Default.aspx

Erro de Servidor no Aplicativo '/'.

Um valor possivelmente perigoso Request.Form foi detectado no cliente (ctl00$ContentPlaceHolder1$txt_NomePais="Áustria").

Descrição: O ASP.NET detectou dados na solicitação que são possivelmente perigosos porque podem incluir marcação ou script HTML. Os dados podem representar uma tentativa de comprometer a segurança do aplicativo, como um ataque de scripts entre sites. Se este tipo de entrada for adequado no aplicativo, você pode incluir código em uma página da Web para permiti-lo explicitamente. Para obter mais informações, consulte http://go.microsoft.com/fwlink/?LinkID=212874.

Detalhes da Exceção: System.Web.HttpRequestValidationException: Um valor possivelmente perigoso Request.Form foi detectado no cliente (ctl00$ContentPlaceHolder1$txt_NomePais="Áustria").

Erro de Origem:

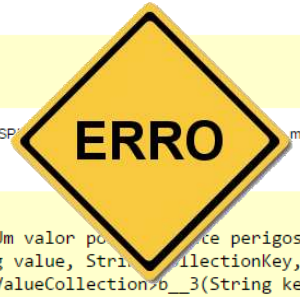
```
[Nenhuma linha de origem relevante]
```

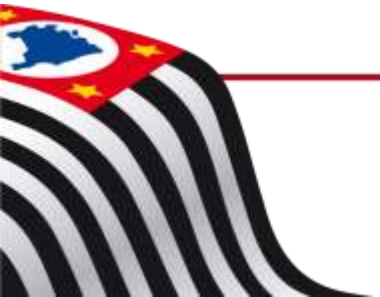
Arquivo de Origem: c:\Users\Eduardo\AppData\Local\Temp\Temporary ASP mdo1pviq.0.cs **Linha:** 0

Rastreamento de Pilha:

```
[HttpRequestValidationException (0x80004005): Um valor po ce perigoso Request.Form foi detectado no cliente (ctl00$ContentPlaceHolder1
   System.Web.HttpRequest.ValidateString(String value, Stri llectionKey, RequestValidationSource requestCollection) +9685609
   System.Web.<>c__DisplayClass5.<ValidateHttpValueCollection>b__3(String key, String value) +18
   System.Web.HttpValueCollection.EnsureKeyValidated(String key) +9685057
   System.Web.HttpValueCollection.Get(String name) +17
   System.Web.UI.WebControls.TextBox.LoadPostData(String postDataKey, NameValueCollection postCollection) +152
   System.Web.UI.WebControls.TextBox.System.Web.UI.IPostBackDataHandler.LoadPostData(String postDataKey, NameValueCollection postCollection) +
   System.Web.UI.Page.ProcessPostData(NameValueCollection postData, Boolean fBeforeLoad) +303
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +6704
   System.Web.UI.Page.ProcessRequest(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +245
   System.Web.UI.Page.ProcessRequest() +72
   System.Web.UI.Page.ProcessRequestWithNoAssert(HttpContext context) +21
   System.Web.UI.Page.ProcessRequest(HttpContext context) +58
   ASP.default_aspx.ProcessRequest(HttpContext context) in c:\Users\Eduardo\AppData\Local\Temp\Temporary ASP.NET Files\vs\1d6b14ff\ad979cf5\Ap
   System.Web.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute() +341
   System.Web.HttpApplication.ExecuteStep(IExecutionStep step, Boolean& completedSynchronously) +69
```

Informações sobre a Versão: Microsoft .NET Framework Versão:4.0.30319; Versão do ASP.NET:4.0.30319.34280





UPDATE (2)

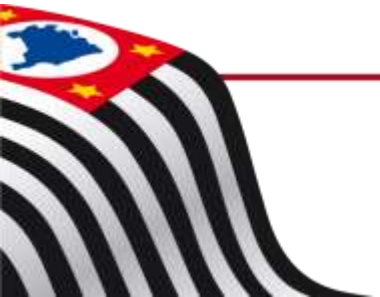
```
protected void Bt_Editar_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection(ConStr);
    con.Open();
    try
    {
        SqlCommand cmd = new SqlCommand("sp_Pais", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@Op", 2); //Editar
        cmd.Parameters.AddWithValue("@NomePais", txt_NomePais.Text);
        cmd.Parameters.AddWithValue("@SiglaPais", txt_Sigla.Text);
        cmd.ExecuteNonQuery();

        GridViewPais.DataBind();

        txt_NomePais.Text   = "";
        txt_Sigla.Text      = "";
    }
    catch (Exception erro)
    {
        lbl_Error.Text = erro.Message;
    }
    finally
    {
        con.Close();
    }
}
}
```

UPDATE (2)

SQLQuery1.sql - ED...OO.FATEC (sa (54))*

```sql
USE [BD.POO.FATEC]
GO
/****** Object:  StoredProcedure [dbo].[sp_Pais]    Script Date: 08
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[sp_Pais]
@Op         int,
@NomePais varchar(80),
@SiglaPais  char(3)
as
begin
if @Op  = 1
    insert into [dbo].[BD_Pais]([BD_PaisNome],[BD_PaisSigla])
    values(@NomePais, @SiglaPais);
else
    UPDATE [dbo].[BD_Pais]
    SET [BD_PaisNome] = @NomePais
      ,[BD_PaisSigla] = @SiglaPais
    WHERE BD_PaisSigla = @SiglaPais
end
```

Capítulo 21: ASP.NET



Centro Paula Souza
GOVERNO DO ESTADO SÃO PAULO

WEB SERVICE

Web service é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Os *Web services* são componentes que permitem às aplicações enviar e receber dados em formato XML. Cada aplicação pode ter a sua própria "linguagem", que é traduzida para uma linguagem universal, o formato *XML*.

XML

XML, do inglês eXtensible Markup Language, é uma linguagem de marcação recomendada pela W3C para a criação de documentos com dados organizados hierarquicamente, tais como textos, banco de dados ou desenhos vetoriais. A linguagem XML é classificada como extensível porque permite definir os elementos de marcação.

http://www.tecmundo.com.br/programacao/1762-o-que-e-xml-.htm

WEB SERVICE/XML

WEB SERVICE/XML

cep.republicavirtual.com.br/web_cep.php?cep=17506000

This XML file does not appear to have any style information associated with


```
▼<webservicecep>
   <resultado>1</resultado>
   <resultado_txt>sucesso - cep completo</resultado_txt>
   <uf>SP</uf>
   <cidade>Marília</cidade>
   <bairro>Somenzari</bairro>
   <tipo_logradouro>Avenida</tipo_logradouro>
   <logradouro>Castro Alves</logradouro>
 </webservicecep>
```

WEB SERVICE/XML

Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

WEB SERVICE/XML

Centro Paula Souza

GOVERNO DO ESTADO SÃO PAULO

WEB SERVICE/XML

```
protected void Bt_Buscar_Click1(object sender, EventArgs e)
{
    DataSet ds = new DataSet();
    ds.ReadXml("http://cep.republicavirtual.com.br/web_cep.php?cep=" + txt_CEP.Text);

    if (ds != null)
    {
        if (ds.Tables[0].Rows.Count > 0)
        {
            txt_Estado.Text = ds.Tables[0].Rows[0]["uf"].ToString().Trim();
            txt_Cidade.Text = ds.Tables[0].Rows[0]["cidade"].ToString().Trim();
            txt_Bairro.Text = ds.Tables[0].Rows[0]["bairro"].ToString().Trim();
            txt_tipo_logradouro.Text = ds.Tables[0].Rows[0]["tipo_logradouro"].ToString().Trim();
            txt_logradouro.Text = ds.Tables[0].Rows[0]["logradouro"].ToString().Trim();
            lbl_resultado.Text = ds.Tables[0].Rows[0]["resultado_txt"].ToString();
        }
    }
}
```

http://www.republicavirtual.com.br/cep/codigos.php

WEB SERVICE/XML

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Xml;
6
7  namespace ConsoleApplication1
8  {
9      class Program
10     {
11         static void Main(string[] args)
12
13         {
14             Console.WriteLine("Informe um CEP");
15             string ax = Console.ReadLine();
16             string filename = "http://cep.republicavirtual.com.br/web_cep.php?cep=" + ax;
17
18             XmlTextReader tr = new XmlTextReader(filename);
19
20             while (tr.Read())
21             {
22                 if (tr.NodeType == XmlNodeType.Text)
23                     System.Console.WriteLine(tr.Value);
24             }
25
26             tr.Close();
27         }
28     }
29 }
30
```

http://www.republicavirtual.com.br/cep/codigos.php

Fim

Sim, eu fiz!
Eu vou fazer.
Eu consigo fazer.
Eu vou tentar fazer.
Como eu faço?
Eu quero fazer.
Eu não consigo fazer.
Eu não vou fazer.