

# Introdução ao ASP.NET – Aula 3

Prof. Me. Sérgio Carlos Portari Júnior

[portari@gmail.com](mailto:portari@gmail.com)

<http://www.sergioportari.com.br>

# Explorando objetos e propriedades

- \* Vamos explorar alguns os objetos e as propriedades de cada um na prática.
- \* Crie um novo projeto vazio asp.net com C#.(EU chamei de Aula 3) .
- \* Adicione um novo item do tipo webform (chamei de formulario1.aspx)
- \* Iremos criar alguns objetos e explorar o que cada um deles faz.

# Label

- \* No novo formulário, insira um objeto Label (lbl\_titulo1)
- \* Na propriedade Text, insira um texto (Exemplo de objetos e propriedades)
- \* Explore as propriedades deste objeto (Font, Forecolor, BorderColor, Backcolor, Border Style, BorderWidth, TooTip, Width, Height, etc)

# Button

- \* Agora, insira abaixo do Label, um objeto Button (btn\_botao1)
- \* Explore as propriedades deste objeto (Font, Forecolor, BorderColor, Backcolor, Border Style, BorderWidth, TooTip, Width, Height, AccessKey, etc)
- \* Iremos criar um código que, quando acionado, troca o texto da label e desativa o botão para que não seja novamente acionado.

# Button

- \* Para isso, dê dois cliques no botão para abrir o source do c#, e insira o seguinte código:

```
protected void btn_botao1_Click(object sender, EventArgs e)
{
    lbl_titulo.Text = "Estou gostando do curso de ASP.NET pra valer!";
    btn_botao1.Enabled = false;
}
```

- \* Execute o projeto e vamos testar

# Button – Chamando um método

- \* Insira outro botão, e vamos agora fazer com que o evento Click acione um método ao invés de executar um código.
- \* Vamos criar um código que habilita mais uma vez o botão anterior e reseta o texto da Label por um método. Para isso, selecione o botão e vá na caixa de propriedades. Click no botão Events (parece um raio).
- \* No evento Click, digite o nome do método a ser criado (coloquei Reiniciar)

# Button – Chamando um método

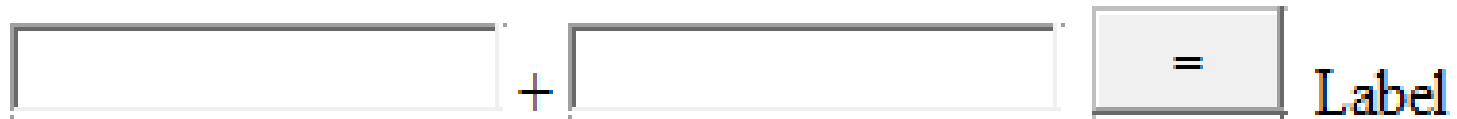
- \* Entre o código abaixo

```
protected void Reiniciar(object sender, EventArgs e)
{
    lbl_titulo.Text = "Exemplo de objetos e propriedades";
    btn_botao1.Enabled = true;
}
```

- \* Execute o código
- \* Lembrem-se, quase todas as propriedades podem ser acessadas e trocadas em tempo de execução (cor, texto, tamanho, fonte, etc).

# Para exercitar

- \* Vamos fazer então uma conta de dois valores digitados em dois textboxes, com o resultado apresentado em uma label?
- \* Monte o layout:



txt\_num1

txt\_num2

btn\_igual lbl\_result



# Para exercitar

- \* O código do btn\_igual é esse?

```
protected void btn_igual_Click(object sender, EventArgs e)
{
    lbl_result.Text = txt_num1.Text + txt_num2.Text;
}
```

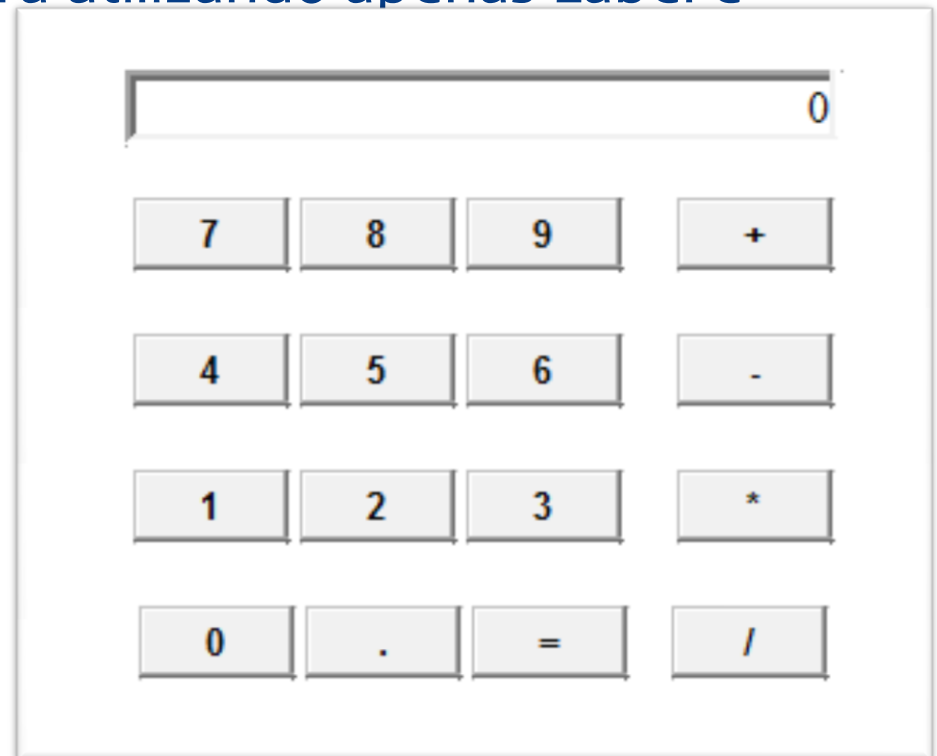
# Para exercitar

- \* Será necessário converter os tipos de dados String para inteiro, somá-los, e depois convertê-los novamente para String para levá-los à propriedade Text da Label;

```
protected void btn_igual_Click(object sender, EventArgs e)
{
    lbl_result.Text = Convert.ToString(Convert.ToInt16(txt_num1.Text)
+ Convert.ToInt16(txt_num2.Text));
}
```

# Agora vamos praticar mais

- \* Monte uma calculadora utilizando apenas Label e Buttons em Asp.Net



# Alterações feitas - Layout

- \* Label:
- \* Width: 250px
- \* BorderStyle: Insert
- \* Text: 0 (zero)
- \* Id: lbl\_total
- \* Botão . :
- \* Width: 56px
- \* Text: . (ponto)
- \* Id: btn\_ponto
- \* Botões de 1 a 9 e operações:
- \* Width: 56px
- \* Text: O mesmo (1 a 9 e operações)
- \* Id:
  - \* btn\_(1 a 9) nos números, ex: btn\_3, btn\_4, etc
  - \* btn\_soma (+); btn\_subtrai (-); btn\_vezes (\*); btn\_divide (/);
- \* Em todos botões e label, o bold do Font foi colocado true

# Alterações feitas - Layout

- \* Botão 0 (Zero):
  - \* Width: 56px
  - \* Text: 0 (Zero)
  - \* Id: btn\_0 (Zero)
- \* Botão = :
  - \* Width: 56px
  - \* Text: = (Zero)
  - \* Id: btn\_calcula
- \* Entre um botão de número e outro existe 1 espaço em branco;
- \* Entre os botões de número e operações, existem 5 espaços em branco.
- \* Entre o botão ponto e o botão de operação existe 5 espaços em branco

# Alterações feitas - Eventos

- \* Selecione todos os botões numéricos (de 0 a 9) e, com todos juntos selecionados, no evento Click da Janela de Propriedades (apertando o raio, lembram?) criem uma chamada a um novo método chamado `insere_numero`.
- \* Para selecionar os botões, pressione e mantenha a tecla Ctrl do teclado e vá clicando com o mouse sobre os botões.

# Alterações feitas - Eventos

```
protected void insere_numero(object sender, EventArgs e)
{
    Button numero = (Button) sender;
    if (lbl_total.Text != "0")
        lbl_total.Text += numero.Text;
    else
    {
        lbl_total.Text = numero.Text;
    }
}
```

Cria uma variável tipo button chamada numero com o objeto que chamou este método (sender), para acessarmos suas propriedades.

# Alterações feitas - Eventos

- \* Para o ponto, temos uma diferença no código: Só pode haver um ponto por número entrado.
- \* Então faremos uma simples repetição, percorrendo todos os números que estiverem na label e, um por um, procurando por um ponto. Se não encontrar o ponto, coloque-o, caso contrário não faça nada.
- \* Faremos então da seguinte maneira:



# Alterações feitas - Eventos

```
protected void btn_ponto_Click(object sender, EventArgs e)
{
    string num = lbl_total.Text;
    char c; int i;
    bool jatem=false;
    for (i = 0; i <num.Length; i++)
    {
        if (num[i] == '.')
            jatem = true;
    }
    if(jatem==false)
    {
        lbl_total.Text += ".";
    }
}
```

# Alterações feitas - Eventos

- \* Para podermos ir acumulando os subtotais e as operações, iremos agora criar dois objeto do tipo HiddenField, o hd\_total e hd\_operacao.
- \* Este tipo de objeto é um campo que fica oculto do usuário, mas possui diversas propriedades que podemos utilizar, e uma delas é a Value, que guarda um valor.
- \* Então, quando apertarmos algum sinal de operação, ele irá processar a operação com o valor do total e com o valor da label.

# Alterações feitas - Eventos

- \* Então vamos configurar os botões de operação.
- \* Lembrem-se que a divisão por zero não pode ser efetuada.
- \* Quando houver uma divisão por zero, vamos zerar todos os parâmetros e começar os cálculos novamente.

# Botão de Soma

```
protected void btn_soma_Click(object sender, EventArgs e)
{
    hd_operacao.Value = "+";
    if (hd_ultvlr.Value != "")
    {
        hd_ultvlr.Value = Convert.ToString(Convert.ToDouble(hd_ultvlr.Value) + Convert.ToDouble(lbl_total.Text));
        lbl_total.Text = "0";
    }
    else
    {
        hd_ultvlr.Value = lbl_total.Text;
        lbl_total.Text = "0";
    }
}
```

# Botão Diferença

```
protected void btn_subtrai_Click(object sender, EventArgs e)
{
    hd_operacao.Value = "-";
    if (hd_ultvlr.Value != "")
    {
        hd_ultvlr.Value = Convert.ToString(Convert.ToDouble(hd_ultvlr.Value) - Convert.ToDouble(lbl_total.Text));
        lbl_total.Text = "0";
    }
    else
    {
        hd_ultvlr.Value = lbl_total.Text;
        lbl_total.Text = "0";
    }
}
```

# Botão multiplica

```
protected void btn_vezes_Click(object sender, EventArgs e)
{
    hd_operacao.Value = "*";
    if (hd_ultvlr.Value != "")
    {
        hd_ultvlr.Value = Convert.ToString(Convert.ToDouble(hd_ultvlr.Value) * Convert.ToDouble(lbl_total.Text));
        lbl_total.Text = "0";
    }
    else
    {
        hd_ultvlr.Value = lbl_total.Text;
        lbl_total.Text = "0";
    }
}
```

# Botão divide

```
protected void btn_divide_Click(object sender, EventArgs e)
{
    hd_operacao.Value = "/";
    if (hd_ultvlr.Value != "")
    {
        if (lbl_total.Text != "0")
        {
            hd_ultvlr.Value = Convert.ToString(Convert.ToDouble(hd_ultvlr.Value) / Convert.ToDouble(lbl_total.Text));
            lbl_total.Text = "0";
        }
        else
        {
            lbl_total.Text = "0";
            hd_operacao.Value = "";
            hd_ultvlr.Value = "";
        }
    }
    else
    {
        hd_ultvlr.Value = lbl_total.Text;
        lbl_total.Text = "0";
    }
}
```

# Botão de cálculo (=)

- \* Este botão tem algumas particularidades.
- \* Para realizar o cálculo, o `hd_operacao` deve ter recebido antes qual operação será realizado, senão ele não faz nada.
- \* O cálculo utilizará dois valores. O Primeiro é entrado antes de apertar o sinal de operação, e quando isso ocorre, ele é transferido para o campo `hd_ultvlr` (em cada botão tem essa operação).
- \* Quando o operador é selecionado, este também vai para `hd_operacao`.
- \* Por último, a operação utiliza o valor da `lbl_valor` para realizar a conta com o operador escolhido.



# Botão de cálculo (=)

```
protected void btn_calcula_Click(object sender, EventArgs e)
{
    if (hd_operacao.Value == "+")
        lbl_total.Text = Convert.ToString(Convert.ToDouble(hd_ultvlr.Value) + Convert.ToDouble(lbl_total.Text));
    if (hd_operacao.Value == "-")
        lbl_total.Text = Convert.ToString(Convert.ToDouble(hd_ultvlr.Value) - Convert.ToDouble(lbl_total.Text));
    if (hd_operacao.Value == "*")
        lbl_total.Text = Convert.ToString(Convert.ToDouble(hd_ultvlr.Value) * Convert.ToDouble(lbl_total.Text));
    if (hd_operacao.Value == "/")
    {
        if (lbl_total.Text == "0")
            lbl_total.Text = "0";
        else
            lbl_total.Text = Convert.ToString(Convert.ToDouble(hd_ultvlr.Value) / Convert.ToDouble(lbl_total.Text));
    }
    hd_ultvlr.Value = "";
    hd_operacao.Value = "";
}
```

# Finalizando... Façam vocês mesmos

- \* Por fim, como exercício, crie três botões a mais, onde
  - \* um deles zera todos os valores e operações para iniciarmos uma nova conta;
  - \* outro para podermos apagar um valor digitado errado;
  - \* e um terceiro, que funcione como um BackSpace, apagando os dígitos entrados para correção, um por um. Lembre-se que se apagar tudo, deve ser exibido o número zero. Apagando o ponto decimal do número pode ser entrado outro novamente.
- \* ENVIAR a pasta toda do projeto em ZIP ou RAR para [portari.uemgituiutaba@gmail.com](mailto:portari.uemgituiutaba@gmail.com)

# Referências

- \* Payne, Chris. **Aprenda em 21 dias ASP.NET**. Rio de Janeiro: Campus, 2001.
- \* **Microsoft .Net Framework SDK Documentation**. Microsoft [2003].
- \* **ASP.NET Web: The Official Microsoft ASP.NET Site : Home Page**. Disponível em :
- \* <<http://www.asp.net>>
- \* World Wide Web Consortium. **Extensible Markup Language (XML)**. Disponível em
- \* <<http://www.w3.org/XML/>>.