

# CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

Componentes

***Prof.***

***Samuel Ribeiro (Samuka)***

**PROCESSWARE**

Consultoria e treinamento

**E-MAIL:** [professor.samuka@gmail.com](mailto:professor.samuka@gmail.com)

2004



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### *Componentes do Asp.Net*

A tecnologia .Net Sabendo da dificuldade em agregar componentes numa aplicação para Web, revolucionou com o asp.net a maneira de se trabalhar com a camada de interface

A plataforma dispõe de componentes prontos com os quais o desenvolvedor consegue manipular suas propriedades como: cor, tamanho, posição no formulário, dentre outros, e atribuir eventos para exercer uma funcionalidade a aplicação.

Existe um conjunto de propriedades comuns para os componentes e que podem ser definidas quando está criando o projeto (tempo desenvolvimento) ou durante a execução do projeto (tempo execução).



# CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

## SINTAXE

```
<asp:NomeComponente Id="IdentidadeComponente" runat="server"/>
```

```
<asp:Button Id="MeuComponente" runat="server"/>
```





# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### SINTAXE

Propriedade	Descrição	Conteúdo
BackColor	Cor de fundo	Cor(Custom, Web ou System)
BorderColor	Cor da borda	Cor(Custom, Web ou System)
Columns	Largura do controle em Caracteres	Número
CssClass	Classe do css.	Nome do css
Font	Formatações de fonte, negrito, itálico...	
ForeColor	Cor de fundo	Cor(Custom, Web ou System)
Text	Texto a ser escrito	Texto



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### SINTAXE

AccessKey	Define a tecla de atalho	Letra simples ( A, G, C )
AutoPostBack	Envia uma solicitação após uma alteração.	True / False
Enabled	Torna o controle ativado ou nao.	True / False
MantainState	Mantem o estado do controle	True / False
TabIndex	Índice ou a ordem de tabulação.	Número
TextMode	Modo de texto	NotSet, SingleLine, MultiLine, Password
ToolTip	Texto de dicas	Texto
Visible	Torna o componente visível ou não.	True / False
DataSource	Determina a origem de dados.	Array, expressão, dataset
DataTextField	Campo de Texto do DataSource definido.	Texto



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### SINTAXE

DataValueField	Campo de valor do DataSource definido.	Texto
Items	Lista de itens do controle	Texto
Height	Altura do componente	Número
Width	Largura do componente	Número
Wrap	Define se poderá ocorrer quebra de linha no texto.	True / False
Modifiers	Define o nível de visibilidade do componente	Public, Protected, Private, Internal.



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### SINTAXE

Quando uma página asp.net é carregada, todas as rotinas de suporte são invocadas. O prefixo <asp: indica o namespace do componente,

e a propriedade runat = "server" informa que serão rodados no servidor.

Quando uma pagina asp.net é carregada ela executa os seguintes passos:

- Os componentes são criados no servidor;

As propriedades dos componentes são ajustadas a partir do \_VIEWSTATE;

- Caso estes componentes tenham algum POST, são invocados os eventos respectivos;
- O componente cria a saída em HTML;
- A página é enviada para o usuário.



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### *TextBox*

O TextBox é um controle de contribuição que deixa o usuário entrar com um texto.

O TextMode da caixa de texto por default é SingleLine, mas pode ser modificado para ser MultiLine ou Password.

A largura de exibição da caixa de texto é determinada por sua propriedade de colunas.

Se for uma caixa de texto de multiline, sua altura de exibição é determinada pela propriedade de linhas.



# CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

## *TextBox Sintaxe*

```
<asp: TextBox id="Identidade"  
AutoPostBack="true/false"  
Columns="n"  
MaxLength="n"  
Rows="n"  
Text="Texto"  
TextMode="Single | MultiLine | Password"  
Wrap="true/false"  
OnTextChanged="Nome-do-Método"  
Runat="server"/>
```





# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### *Label*

O Label é usado para exibir texto em um local fixo da página.

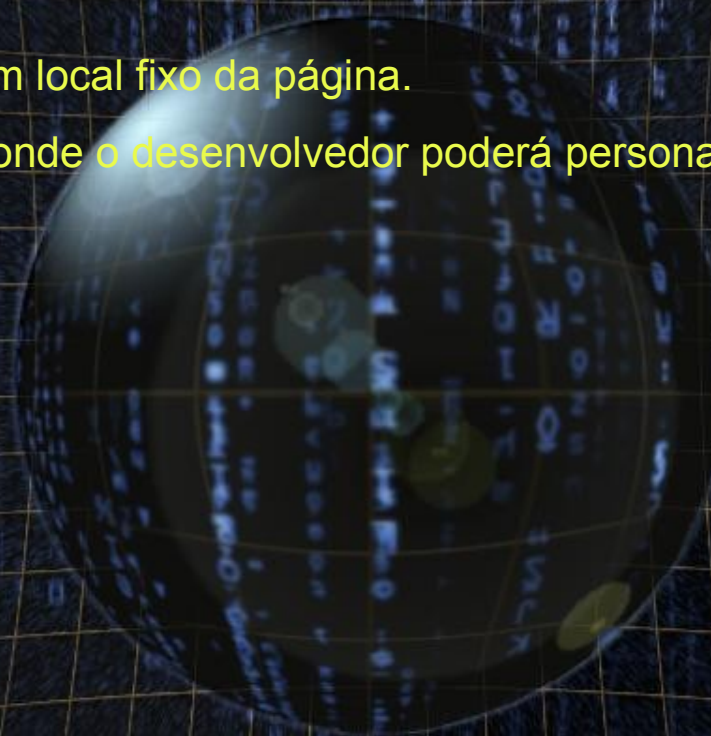
Este controle exibe um Texto estático, onde o desenvolvedor poderá personalizar o texto exibido pela propriedade Text.

### *Sintaxe*

```
<asp:Label id="Identidade"
```

```
Text="Texto"
```

```
runat="server"/>
```





# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### *Button*

Use o controle Button para criar um botão de clique na sua página de Web.

Um botão de comando tem uma identificação associada com o botão, fixado pela propriedade CommandName. Isto lhe permite criar múltiplos botões de controle em uma página Web , porque é possível determinar qual botão o usuário clicou. Você também pode usar a propriedade CommandArgument com um botão de comando para prover uma informação adicional sobre o comando a ser executado

### *Sintaxe*

```
<asp:Button id="Identidade"
```

```
Text="Título"
```

```
OnClick="Metodo-Controle"
```

```
Runat="server"/>
```



# CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

## *DropDownList (Ddl)*

### **DropDownList**

O objetivo de usar o DropDownList (Ddl) é disponibilizar uma lista para o internauta selecionar uma opção. Esta é uma das formas mais utilizadas nos aplicativos quando é preciso que o internauta apenas selecione uma opção ao invés de ter que digitar um dado.

Esta técnica é utilizada também quando se trabalha com banco de dados onde é montada uma Ddl baseada em uma tabela, assegurando a integridade da informação. Sempre que você precisar padronizar a interatividade com os internautas em situações que existe um conjunto de opções fixas. Uma boa alternativa é utilizar este componente.



# CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

## *DropDownList (Ddl)*

### *Sintaxe*

```
<asp:DropDownList id="Identidade"
```

```
AutoPostBack="true/false"
```

```
OnSelectedIndexChanged="Metodo-a-disparar"
```

```
runat="server"/>
```

```
<asp:ListItem Value="Valor" Selected="true/false">
```

```
Texto
```

```
</asp:ListItem>
```

```
</asp:DropDownList>
```



# CURSO DE EXTENSÃO

DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

## *DropDownList (Ddl)*

### *Sintaxe*

```
<asp:DropDownList id="Identidade"
```

```
AutoPostBack="true/false"
```

```
OnSelectedIndexChanged="Metodo-a-disparar"
```

```
runat="server"/>
```

```
<asp:ListItem Value="Valor" Selected="true/false">
```

```
Texto
```

```
</asp:ListItem>
```

```
</asp:DropDownList>
```



# CURSO DE EXTENSÃO

## DESENVOLVIMENTO DE APLICAÇÕES COMERCIAIS EM C# (.NET)

### *do while*

O loop **do while** tem a mesma função da instrução **While**, a única diferença é que a condição é testada no final da instrução. Com isso podemos assegurar em alguns casos na linguagem C# que o loop seja realizado pelo menos uma única vez.

#### Sintaxe

```
do
{
Instrucoes;
}
while ( condição );
```

#### Exemplo

```
do
{
Console.WriteLine( Numero );
}
while ( Numero < 10 );
```