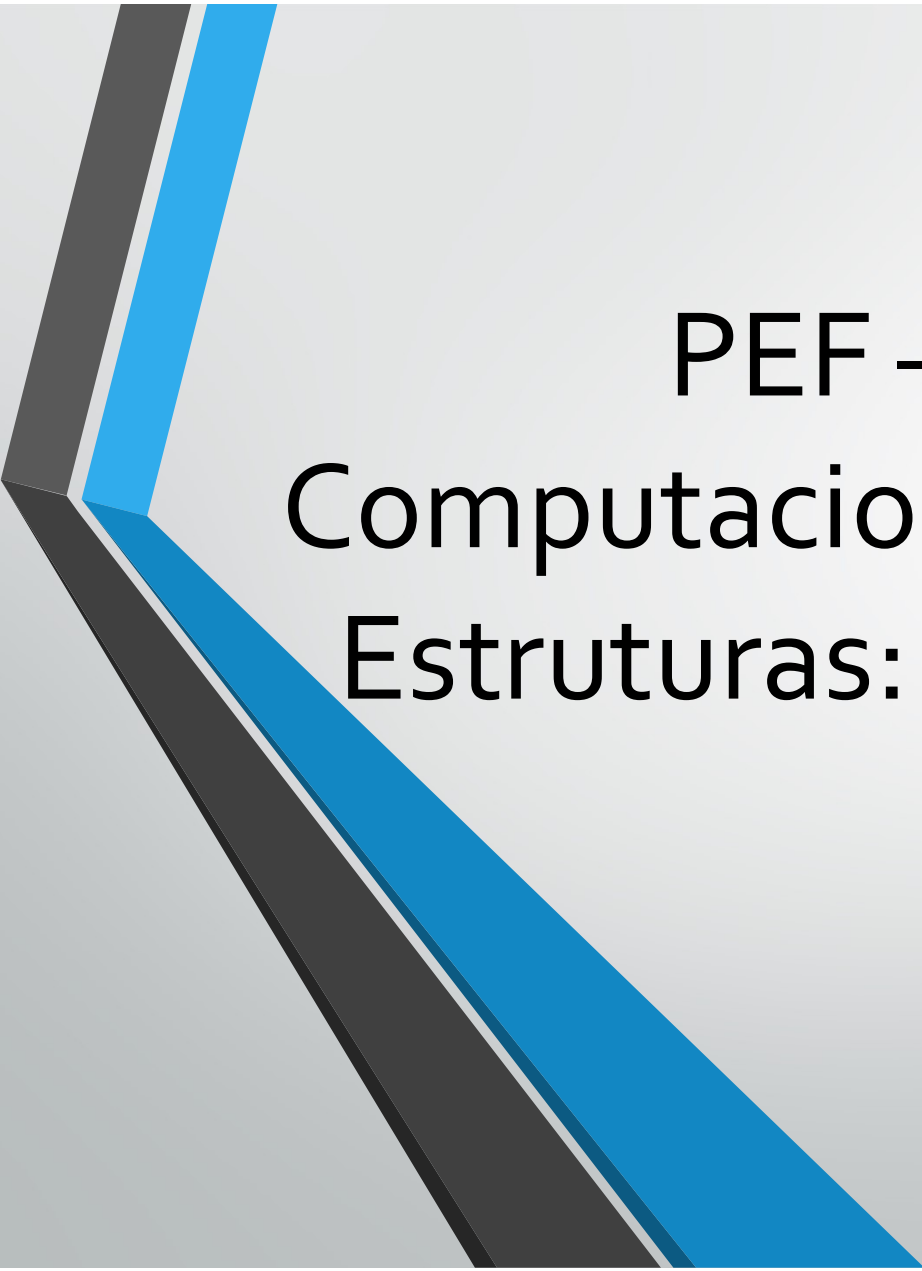




PEF – 3528 – Ferramentas Computacionais na Mecânica das Estruturas: Criação e Concepção

Prof. Dr. Rodrigo Provasi

e-mail: provasi@usp.br

Sala 09 – LEM – Prédio de Engenharia Civil



Interfaces em C#

Elementos

Elementos em WPF

- Existem vários elementos que podem ser usados no WPF.
- Aqui serão listados os mais usuais e comentários serão feitos.

Botões

- Existem diversos tipos de botões no WPF. Todos derivam de *ButtonBase*. São:
- **Button:** botão tradicional.
- **RepeatButton:** botão com repetição. Fica chamando constantemente o evento de click do botão quando pressionado.
- **ToggleButton:** botão que permanece selecionado. Possui três estados: *true*, *null* e *false*. Cada click no botão faz com que mova para o próximo estado em ciclos.
- **CheckBox:** é o *ToggleButton* com uma aparência mais simples (e é mais usado).
- **RadioButton:** também deriva de *ToggleButton*, mas tem um controle de exclusão mútua. Quando agrupados permitem a seleção de apenas um dos possíveis casos.

Radio Button

```
<StackPanel>  
  <RadioButton>Option 1</RadioButton>  
  <RadioButton>Option 2</RadioButton>  
  <RadioButton>Option 3</RadioButton>  
</StackPanel>
```

```
<StackPanel>  
  <StackPanel>  
    <RadioButton GroupName="A">Option 1</RadioButton>  
    <RadioButton GroupName="A">Option 2</RadioButton>  
  </StackPanel>  
  <StackPanel>  
    <RadioButton GroupName="A">Option 3</RadioButton>  
  </StackPanel>  
</StackPanel>
```

Radio Button

```
<StackPanel>
```

```
  <RadioButton GroupName="A">Option 1</RadioButton>
```

```
  <RadioButton GroupName="A">Option 2</RadioButton>
```

```
  <RadioButton GroupName="B">A Different Option 1</RadioButton>
```

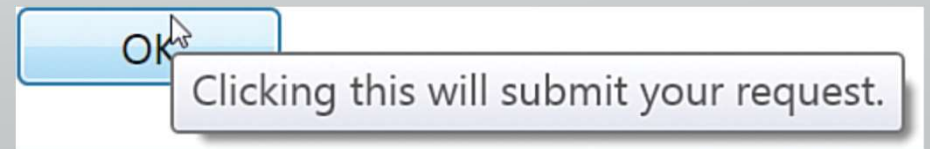
```
  <RadioButton GroupName="B">A Different Option 2</RadioButton>
```

```
</StackPanel>
```

Containers

- **Label:** o *label* é o controle mais clássico possível e só armazena texto.
- **Tooltip:** esse controle serve para mostrar uma dica sobre algum elemento:

```
<Button>  
  OK  
  <Button.ToolTip>  
    <ToolTip>  
      Clicking this will submit your request.  
    </ToolTip>  
  </Button.ToolTip>  
</Button>
```



Containers

```
<CheckBox>
  CheckBox
  <CheckBox.ToolTip>
    <StackPanel>
      <Label FontWeight="Bold" Background="Blue" Foreground="White">
        The CheckBox
      </Label>
      <TextBlock Padding="10" TextWrapping="WrapWithOverflow" Width="200">
        CheckBox is a familiar control. But in WPF, it's not much more than a ToggleButton styled differently!
      </TextBlock>
      <Line Stroke="Black" StrokeThickness="1" X2="200"/>
      <StackPanel Orientation="Horizontal">
        <Image Margin="2" Source="help.gif"/>
        <Label FontWeight="Bold">Press F1 for more help.</Label>
      </StackPanel>
    </StackPanel>
  </CheckBox.ToolTip>
</CheckBox>
```



Containers

- É possível definir uma duração para a *tooltip*:

```
<Button ToolTipService.ShowDuration="3000">
```

```
...
```

```
</Button>
```

GroupBox

- Permite agrupar controles.

```
<GroupBox Header="Grammar">
```

```
  <StackPanel>
```

```
    <CheckBox>Check grammar as you type</CheckBox>
```

```
    <CheckBox>Hide grammatical errors in this document</CheckBox>
```

```
    <CheckBox>Check grammar with spelling</CheckBox>
```

```
  </StackPanel>
```

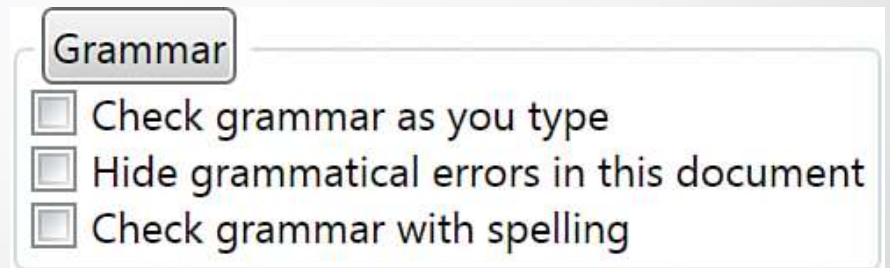
```
</GroupBox>
```

Grammar

- ☐ Check grammar as you type
- ☐ Hide grammatical errors in this document
- ☐ Check grammar with spelling

Groupbox

```
<GroupBox>
  <GroupBox.Header>
    <Button>Grammar</Button>
  </GroupBox.Header>
  <StackPanel>
    <CheckBox>Check grammar as you type</CheckBox>
    <CheckBox>Hide grammatical errors in this document</CheckBox>
    <CheckBox>Check grammar with spelling</CheckBox>
  </StackPanel>
</GroupBox>
```



Expander

- Permite expandir ou ocultar os itens internos do controle.

`<Expander Header="Grammar">`

`<StackPanel>`

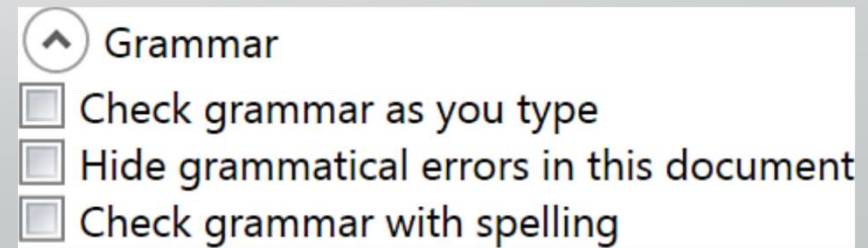
`<CheckBox>Check grammar as you type</CheckBox>`

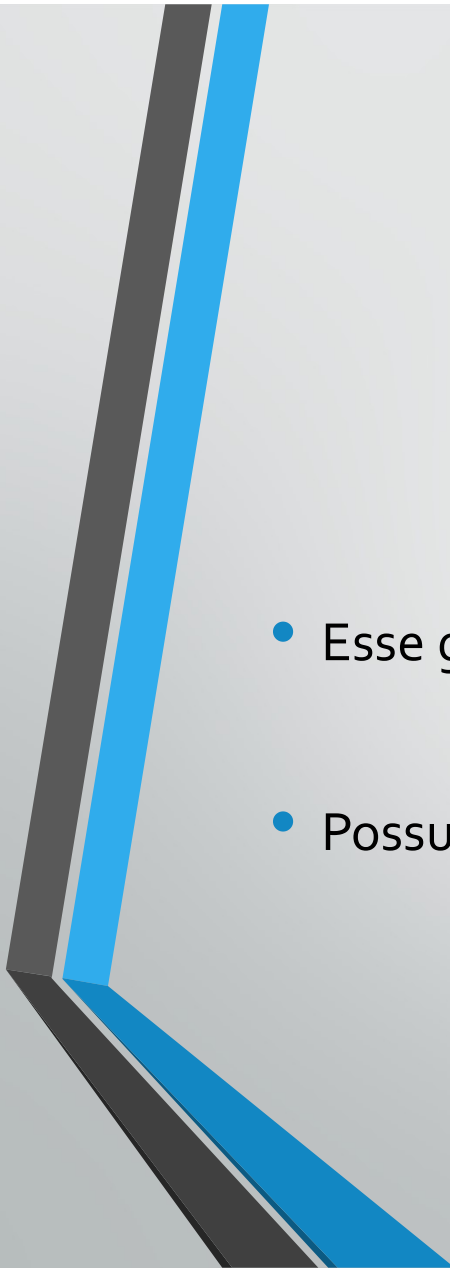
`<CheckBox>Hide grammatical errors in this document</CheckBox>`

`<CheckBox>Check grammar with spelling</CheckBox>`

`</StackPanel>`

`</Expander>`





Item Controls

- Esse grupo de elementos permitem o controle de itens contido neles.
- Possuem uma propriedade *Items* que armazena os itens a serem usados.

Listbox

- Adiciona elementos em forma de lista.

```
<ListBox xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:sys="clr-namespace:System;assembly=mscorlib">
  <Button>Button</Button>
  <Expander Header="Expander"/>
  <sys:DateTime>1/1/2007</sys:DateTime>
  <sys:DateTime>1/2/2007</sys:DateTime>
  <sys:DateTime>1/3/2007</sys:DateTime>
</ListBox>
```



Listbox

- O elemento possui uma propriedade denominada *ItemsSource* que permite ligar diretamente o conteúdo do controle à um objeto que tenha itens (por exemplo, uma lista ligada).
- Há ainda ferramentas para seleção dos itens.

Seletores

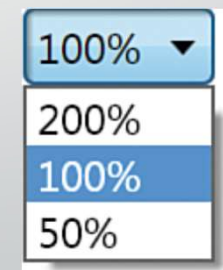
- **SelectedIndex**: retorna o índice do item selecionado (lembrando que as listas são *zero-based*).
- **SelectedItem**: retorna o item selecionado.
- **SelectedValue**: retorna o valor do item selecionado. Por padrão, o valor do item é o mesmo que o item. Porém, setando um *SelectedValuePath* é possível retornar algo que represente o valor do item.

Seletores

- **IsSelected**: retorna se o item está selecionado ou não.
- **IsSelectionActive**: retorna se o item selecionado tem ou não o foco.
- Os controles que possuem esses itens de seleção são:
 - *ComboBox*
 - *ListBox*
 - *ListView*
 - *TabControl*

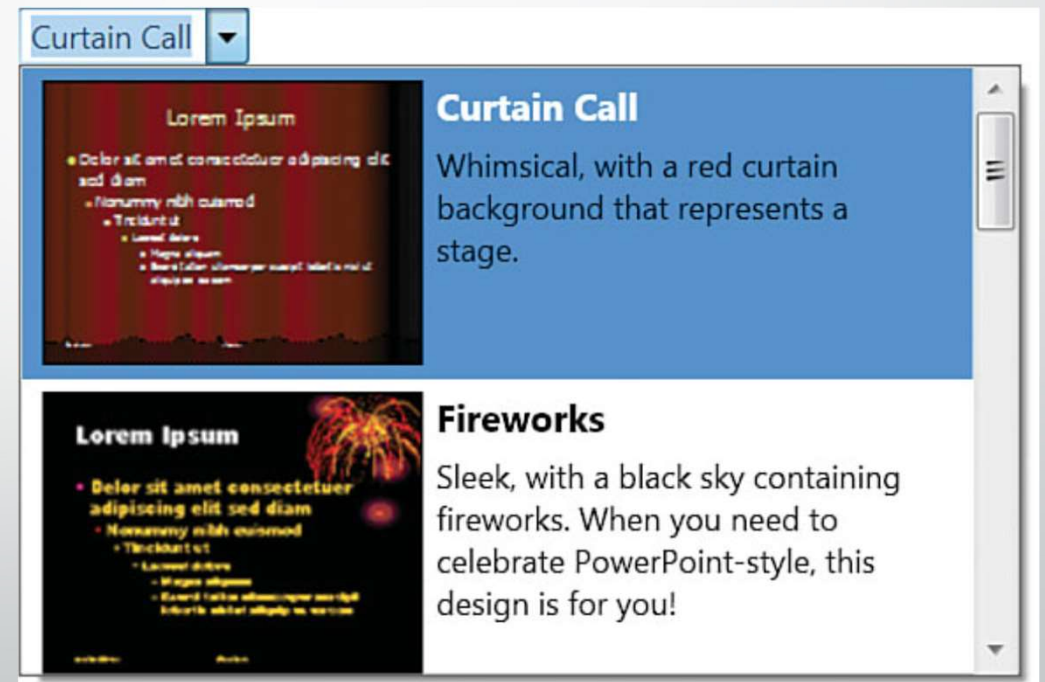
ComboBox

- Permite o usuário selecionar algum item de uma lista que aparece quando se clica no botão (*drop down list*).
- Ele exibe apenas o item que está selecionado.



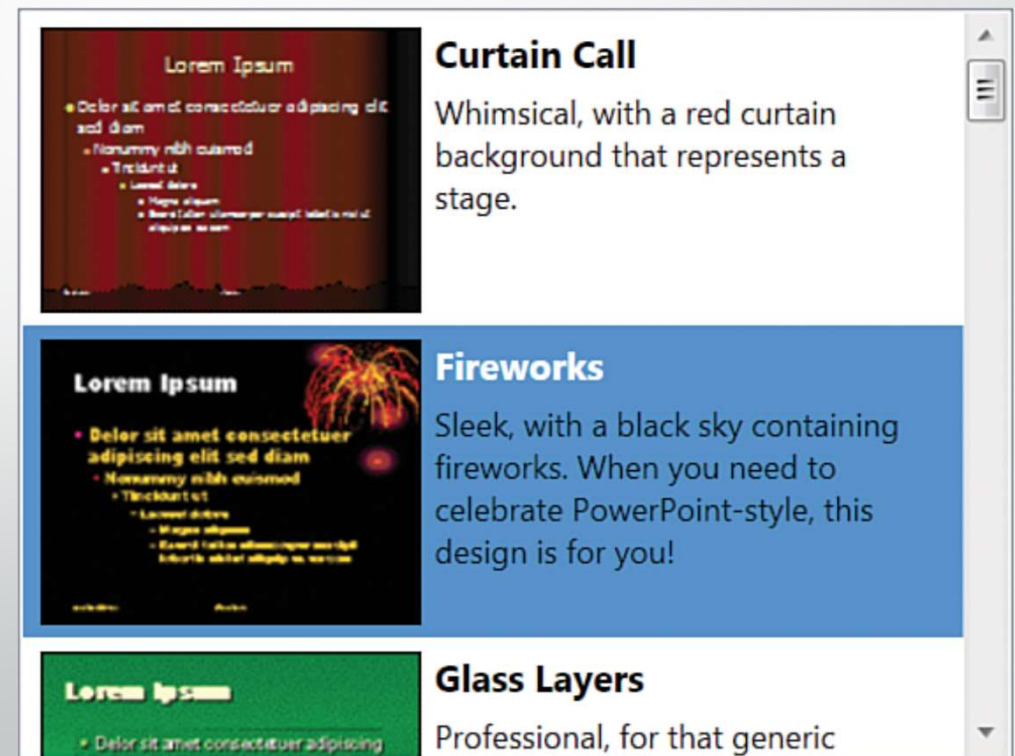
Combobox

- Permite várias modificações nos itens.



ListBox

- É similar ao *ComboBox*, porém exibe os itens dentro do contorno do controle.
- Permite seleção única, múltipla e estendida.





ListView

- Similar ao *ListBox*, com comportamento *extended* por padrão.
- Permite controlar o *view*.

List View

```
<ListView>
  <ListView.View>
    <GridView>
      <GridViewColumn Header="Date"/>
      <GridViewColumn Header="Day of Week" DisplayMemberBinding="{Binding DayOfWeek}"/>
      <GridViewColumn Header="Year" DisplayMemberBinding="{Binding Year}"/>
    </GridView>
  </ListView.View>
  <sys:DateTime>1/1/2007</sys:DateTime>
  <sys:DateTime>1/2/2007</sys:DateTime>
  <sys:DateTime>1/3/2007</sys:DateTime>
</ListView>
```

Date	Day of Week	Year	
1/1/2007	Monday	2007	
1/2/2007	Tuesday	2007	
1/3/2007	Wednesday	2007	

TabControl

- Permite colocar objetos em abas. Útil para selecionar entre várias páginas de conteúdo.

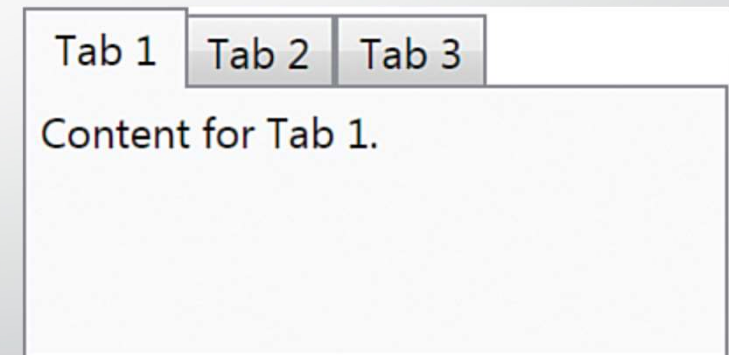
<TabControl>

<TabItem Header="Tab 1">Content for Tab 1.</TabItem>

<TabItem Header="Tab 2">Content for Tab 2.</TabItem>

<TabItem Header="Tab 3">Content for Tab 3.</TabItem>

</TabControl>



Menu

- Menus implementam os menus tradicionais.

`<Menu>`

`<MenuItem Header="_File">`

`<MenuItem Header="_New..." />`

`<MenuItem Header="_Open..." />`

`<Separator />`

`<MenuItem Header="Send To">`

`<MenuItem Header="Mail Recipient" />`

`<MenuItem Header="My Documents" />`

`</MenuItem>`

`</MenuItem>`

`<MenuItem Header="_Edit">`

`...`

`</MenuItem>`

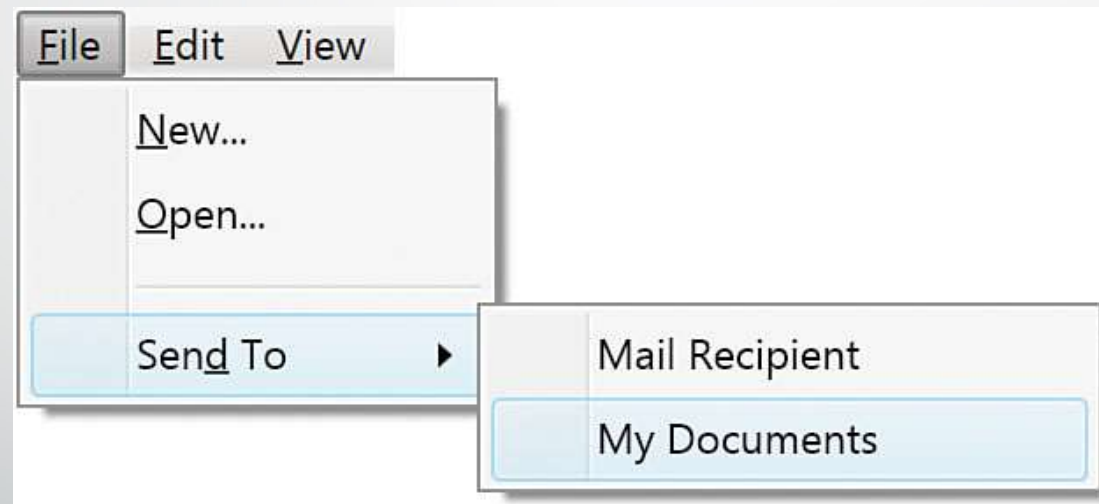
`<MenuItem Header="_View">`

`...`

`</MenuItem>`

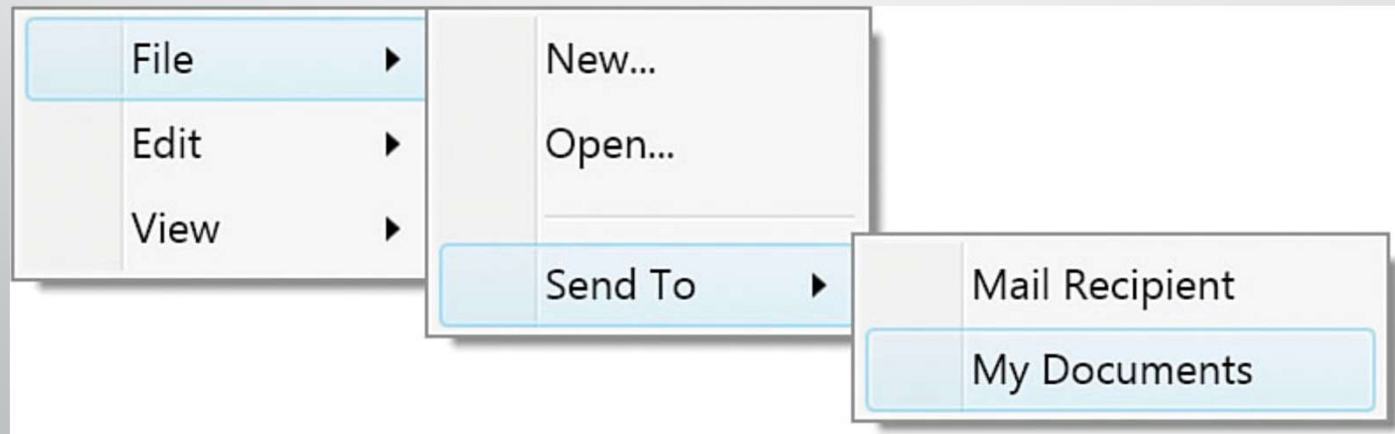
`</Menu>`

Menu



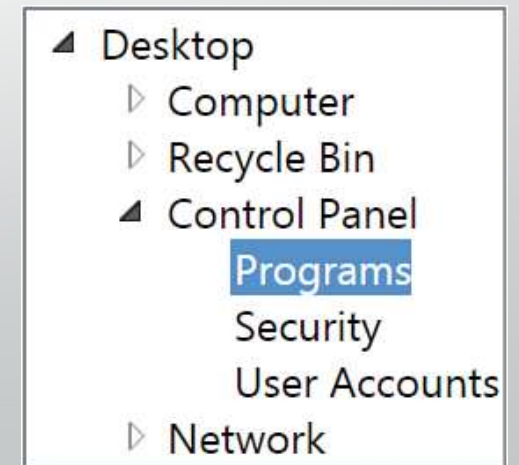
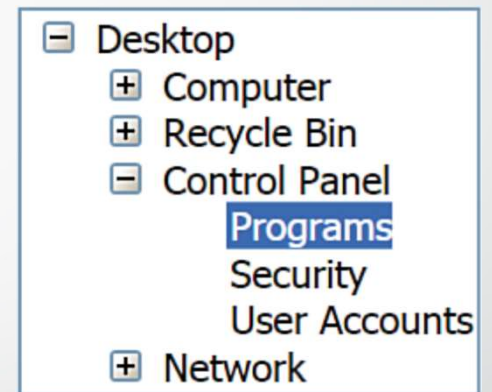
ContextMenu

- Funciona como o menu.
- Acesso com botão direito do *mouse*.



TreeView

```
<TreeView>
  <TreeViewItem Header="Desktop">
    <TreeViewItem Header="Computer">
      ...
    </TreeViewItem>
    <TreeViewItem Header="Recycle Bin">
      ...
    </TreeViewItem>
    <TreeViewItem Header="Control Panel">
      <TreeViewItem Header="Programs"/>
      <TreeViewItem Header="Security"/>
      <TreeViewItem Header="User Accounts"/>
    </TreeViewItem>
    <TreeViewItem Header="Network">
      ...
    </TreeViewItem>
  </TreeViewItem>
</TreeView>
```



Toolbar

<ToolBar>

<Button><Image Source="copy.gif"/></Button>

<Separator/>

<ToggleButton><Image Source="bold.gif"/></ToggleButton>

<ToggleButton><Image Source="italic.gif"/></ToggleButton>

<ToggleButton><Image Source="underline.gif"/></ToggleButton>

<Separator/>

<ToggleButton><Image Source="left.gif"/></ToggleButton>

<ToggleButton><Image Source="right.gif"/></ToggleButton>

<ToggleButton><Image Source="justify.gif"/></ToggleButton>

<Separator/>

<Label>Zoom</Label>

<ComboBox>

...

</ComboBox>

<Separator/>

<Button><Image Source="superscript.gif"/></Button>

<Button><Image Source="subscript.gif"/></Button>

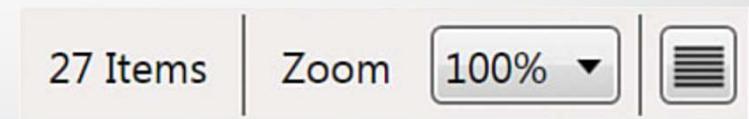
...

</ToolBar>



Statusbar

```
<StatusBar>  
  <Label>27 Items</Label>  
  <Separator/>  
  <Label>Zoom</Label>  
  <ComboBox>  
    ...  
  </ComboBox>  
  <Separator/>  
  <Button><Image Source="justify.gif"/></Button>  
</StatusBar>
```



Controles de intervalo

- Eles guardam e exibem valores em um determinado intervalo.
- Possuem propriedades como: *Value*, *Minimum* e *Maximum*.
- São: *ProgressBar* e *Slider*

ProgressBar

- ***ProgressBar***: Exibe uma barra de progresso.



- ***Slider***: exibe um controle que permite alterar a posição da barra que indica o *status*.



Texto e Tinta

- Existem controles que permitem a entrada via texto e por caneta. São:
 - *TextBox*
 - *RichTextBox*
 - *PasswordBox*
 - *InkCanvas*

TextBox

- É um controle que permite a entrada de uma ou mais linhas de texto.
- Diferente dos outros controles, o que o controle armazena é um texto (propriedade *Text* ao invés de *Content*).



TextBox

RichTextBox

- É um controle similar ao *TextBox* que permite controles mais avançados.
- Permite, por exemplo, formatações como no exemplo:

RichTextBox

PasswordBox

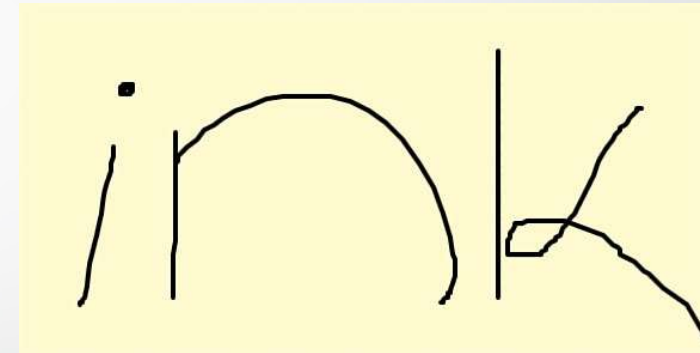
- É um *TextBox* para a entrada de senhas, exibindo o texto como círculos.
- Não deriva de *TextBase* e, portanto, não possui *Cut*, *Copy*, *Paste*, *Undo* e *Redo*.



InkCanvas

- Permite a entrada de via caneta em um *Canvas*.

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
SizeToContent="WidthAndHeight">
  <Grid>
    <InkCanvas>
      <Image Source="http://pinvoke.net/blog/images/anathan.png"/>
    </InkCanvas>
  </Grid>
</Window>
```





Interfaces em C#

Posicionamento e agrupamento

Controlando o Tamanho

- Quando a janela redimensiona, os elementos contidos nesse, são reajustados.
- Nesse redimensionamento, os elementos tendem a se ajustar ao conteúdo.
- As propriedades de tamanho podem ser alteradas pelo desenvolvedor.

Controlando o tamanho

- ***Height e Width***: controlam altura e largura do controle. Possuem valores máximos e mínimos, além de preferenciáveis.
- ***Auto*** → Propriedade para tamanho que significa que o ajuste é automático (o redimensionamento fará com que o controle ocupe o espaço máximo).

Controlando o tamanho

- ***Margin e Padding***: são controles similares, definido uma margem externa (aumentando o tamanho do elemento) ou uma interna (diminuindo o tamanho do elemento).

Controlando o tamanho

<!-- PADDING: -->

<!-- 1 value: The same padding on all four sides: -->

<Label Padding="0" Background="Orange">0</Label>

<Label Padding="10" Background="Orange">10</Label>

<!-- 2 values: Left & Right get the 1st value,

Top & Bottom get the 2nd value: -->

<Label Padding="20,5" Background="Orange">20,5</Label>

<!-- 4 values: Left,Top,Right,Bottom: -->

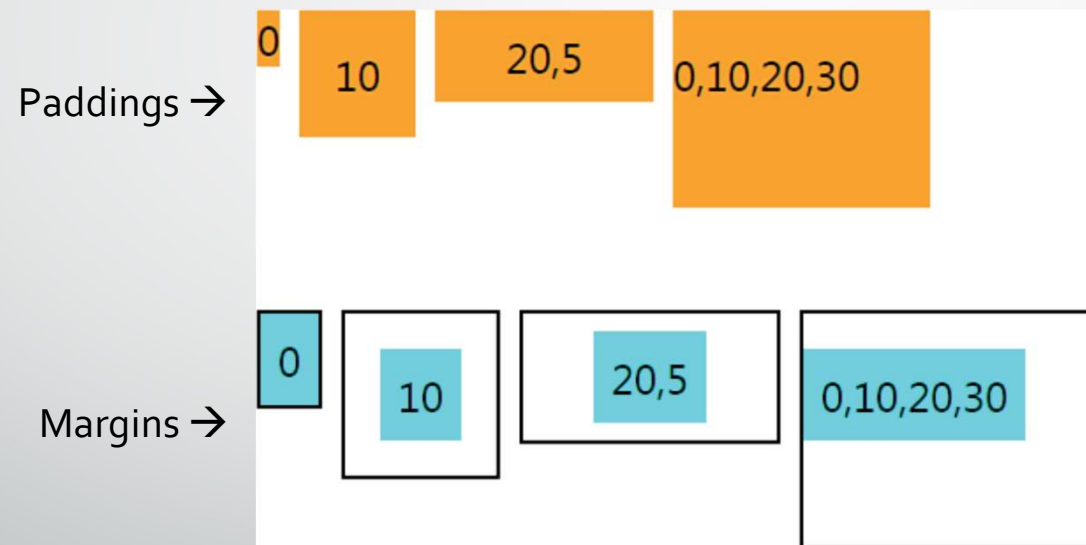
<Label Padding="0,10,20,30" Background="Orange">0,10,20,30</Label>

Controlando o tamanho

```
<!-- MARGIN: -->
<Border BorderBrush="Black" BorderThickness="1">
    <!-- No margin: -->
    <Label Background="Aqua">0</Label>
</Border>
<Border BorderBrush="Black" BorderThickness="1">
    <!-- 1 value: The same margin on all four sides: -->
    <Label Margin="10" Background="Aqua">10</Label>
</Border>
```

```
<Border BorderBrush="Black" BorderThickness="1">
    <!-- 2 values: Left & Right get the 1st value, Top & Bottom get the
    2nd value: -->
    <Label Margin="20,5" Background="Aqua">20,5</Label>
</Border>
<Border BorderBrush="Black" BorderThickness="1">
    <!-- 4 values: Left,Top,Right,Bottom: -->
    <Label Margin="0,10,20,30"
    Background="Aqua">0,10,20,30</Label>
</Border>
```

Controlando o tamanho

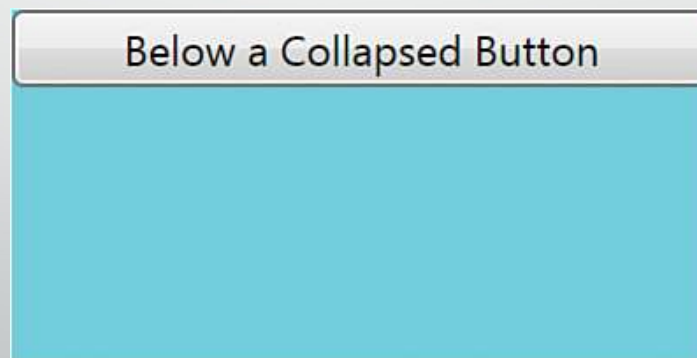


Visibilidade

- Controles permitem mudança da visibilidade:
 - **Visible**— o elemento é visível e participa do *layout*.
 - **Collapsed**— o elemento é invisível e não participa do *layout*.
 - **Hidden**—o elemento é invisível e ainda assim patiticipa do *layout*.

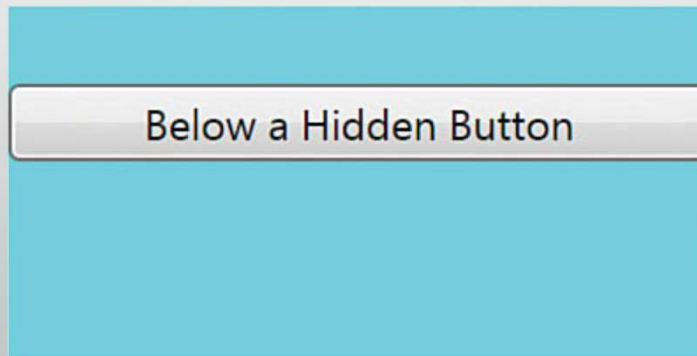
Visibilidade

```
<StackPanel Height="100" Background="Aqua">  
    <Button Visibility="Collapsed">Collapsed Button</Button>  
    <Button>Below a Collapsed Button</Button>  
</StackPanel>
```



Visibilidade

```
<StackPanel Height="100" Background="Aqua">  
    <Button Visibility="Hidden">Hidden Button</Button>  
    <Button>Below a Hidden Button</Button>  
</StackPanel>
```



Controlando a posição

- É possível controlar a posição dos elementos:
 - *HorizontalAlignment: Left, Center, Right, and Stretch*
 - *VerticalAlignment: Top, Center, Bottom, and Stretch*

Controlando a posição

```
<StackPanel>
```

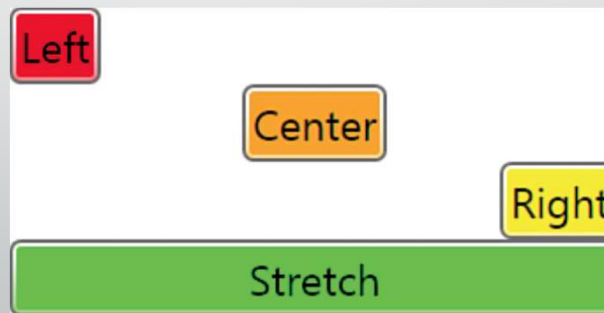
```
  <Button HorizontalAlignment="Left" Background="Red">Left</Button>
```

```
  <Button HorizontalAlignment="Center" Background="Orange">Center</Button>
```

```
  <Button HorizontalAlignment="Right" Background="Yellow">Right</Button>
```

```
  <Button HorizontalAlignment="Stretch" Background="Lime">Stretch</Button>
```

```
</StackPanel>
```



Controlando a posição

- É possível controlar a posição do conteúdo dos elementos.
- Não afeta o tamanho do elemento.

`<StackPanel>`

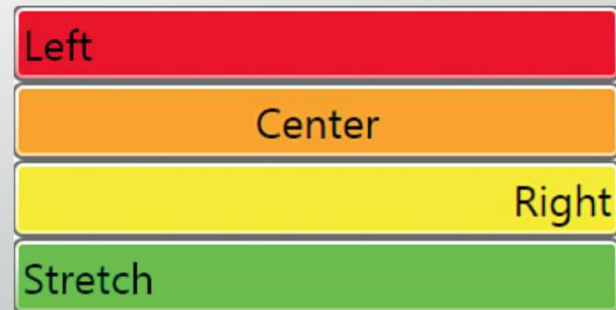
`<Button HorizontalContentAlignment="Left" Background="Red">Left</Button>`

`<Button HorizontalContentAlignment="Center" Background="Orange">Center</Button>`

`<Button HorizontalContentAlignment="Right" Background="Yellow">Right</Button>`

`<Button HorizontalContentAlignment="Stretch" Background="Lime">Stretch</Button>`

`</StackPanel>`



Controlando a posição

- **FlowDirection:** permite o controle de como controle é preenchido.

`<StackPanel>`

```
<Button FlowDirection="LeftToRight" HorizontalContentAlignment="Left" VerticalContentAlignment="Top"
Height="40" Background="Red">LeftToRight</Button>
```

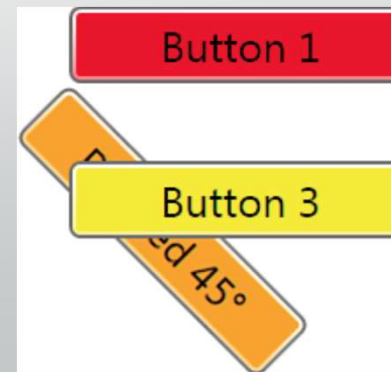
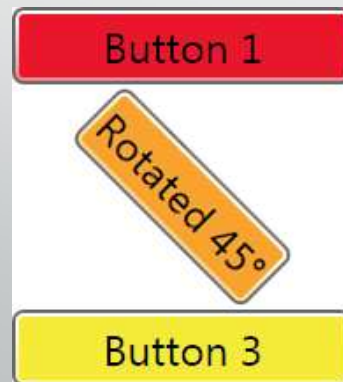
```
<Button FlowDirection="RightToLeft" HorizontalContentAlignment="Left" VerticalContentAlignment="Top"
Height="40" Background="Orange">RightToLeft</Button>
```

`</StackPanel>`



Transformações

- O WPF possui controles embutidos de elementos 2D que permite mudar os componentes independentemente dos controles anteriormente vistos.
- São possíveis as seguintes transformações:
 - *LayoutTransform*: é aplicado antes do processo de *layout* ser feito.
 - *RenderTransform*: é aplicado depois do processo de *layout* (em seguida a renderização do elemento).



Transformações

- Existem várias transformações possíveis:
 - *RotateTransform*
 - *ScaleTransform*
 - *SkewTransform*
 - *TranslateTransform*
- É possível ainda combinar as transformações.

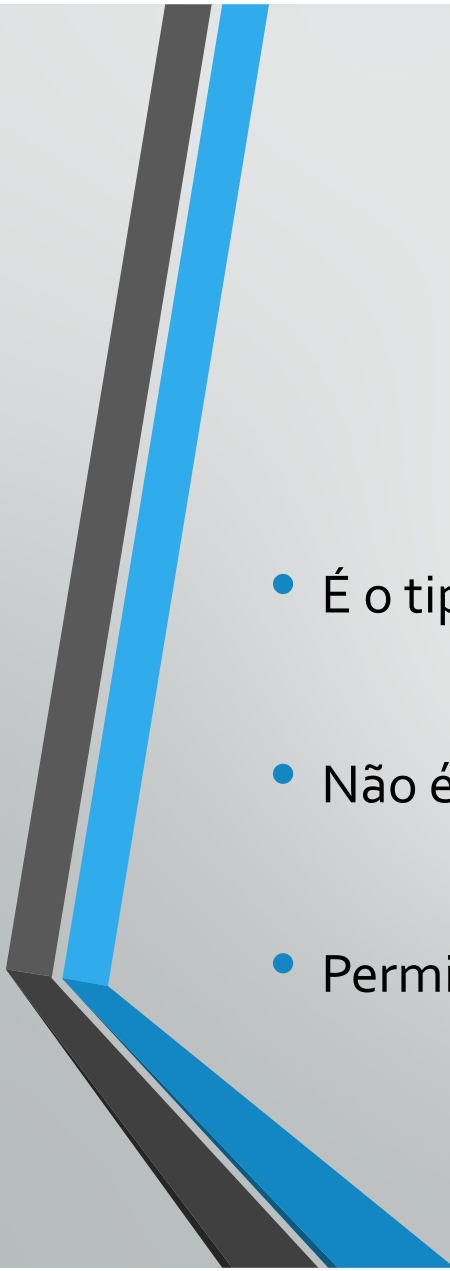


Interfaces em C#

Painéis

Painéis

- O WPF permite organizar os itens em painéis.
- Existem os seguintes controles de painéis:
 - *Canvas*
 - *StackPanel*
 - *WrapPanel*
 - *DockPanel*
 - *Grid*



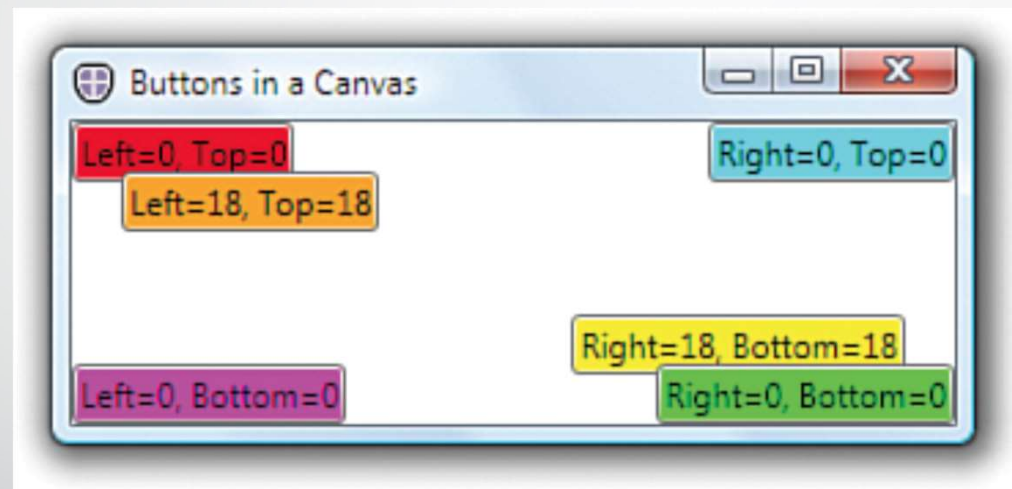
Canvas

- É o tipo mais básico de painel.
- Não é utilizado em interfaces (usado para desenho).
- Permite o posicionamento absoluto dos itens.

Canvas

```
<Window xmlns=http://schemas.microsoft.com/winfx/2006/xaml/presentation Title="Buttons in a Canvas">
  <Canvas>
    <Button Background="Red">Left=0, Top=0</Button>
    <Button Canvas.Left="18" Canvas.Top="18" Background="Orange">Left=18, Top=18</Button>
    <Button Canvas.Right="18" Canvas.Bottom="18" Background="Yellow">Right=18, Bottom=18</Button>
    <Button Canvas.Right="0" Canvas.Bottom="0" Background="Lime">Right=0, Bottom=0</Button>
    <Button Canvas.Right="0" Canvas.Top="0" Background="Aqua">Right=0, Top=0</Button>
    <Button Canvas.Left="0" Canvas.Bottom="0" Background="Magenta">Left=0, Bottom=0</Button>
  </Canvas>
</Window>
```

Canvas



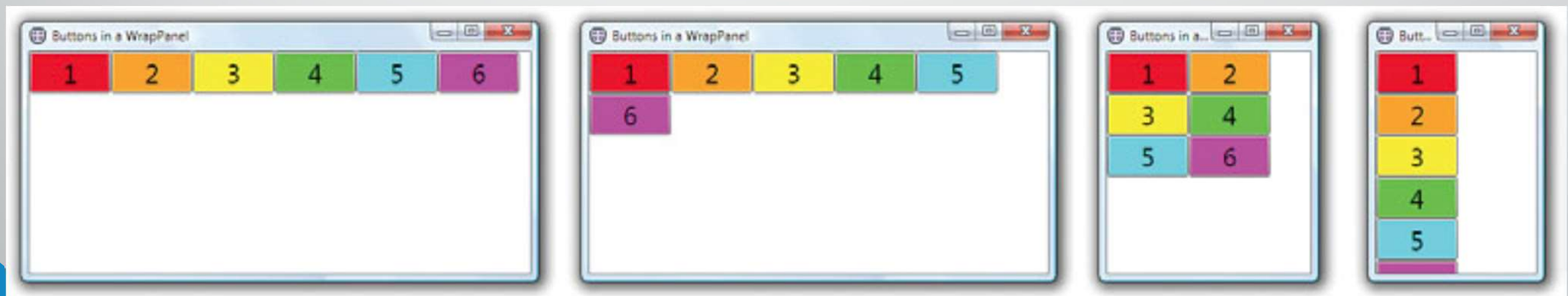
StackPanel

- É um tipo de painel que empilha os componentes.
- Pode ter orientação horizontal ou vertical.



WrapPanel

- Funciona como um *StackPanel*.
- Permite controlar o transbordo de elementos para uma nova linha/coluna.

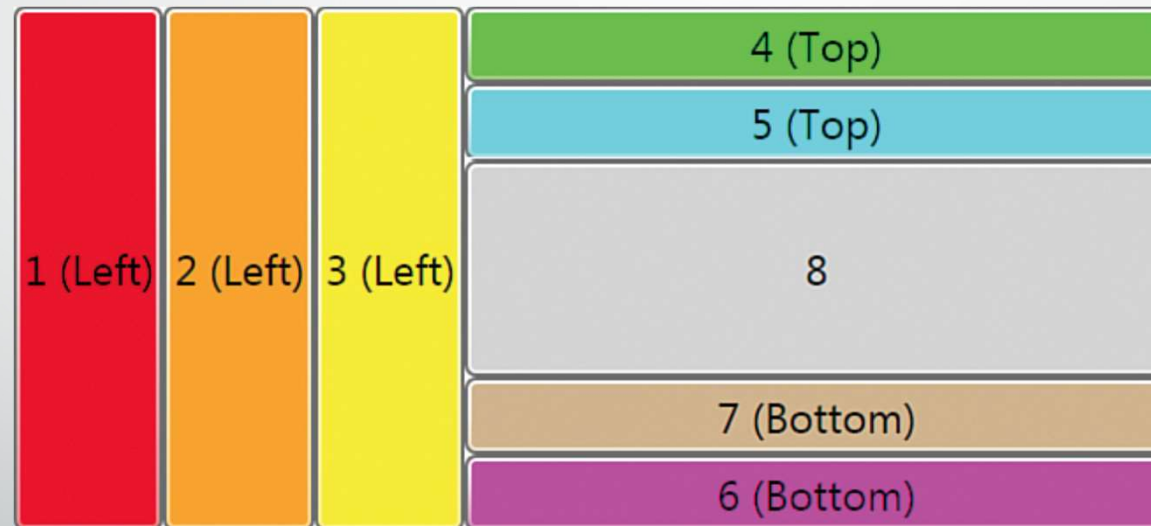


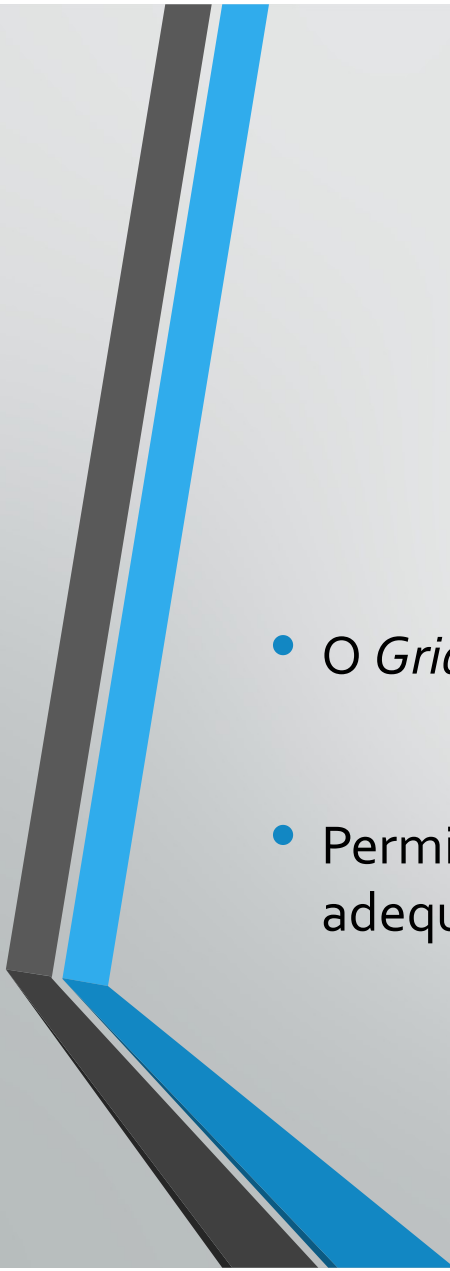
DockPanel

- Esse controle permite o ancoramento de elementos ou mesmo o preenchimento do controle.
- Exemplo: mudança dos botões em um *DockPanel*.



DockPanel





Grid

- O *Grid* é um dos controles mais versáteis do WPF.
- Permite dividir em linhas e colunas e colocar os elementos nas posições adequadas.

Grid

```
<Grid Background="LightBlue">
  <!-- Define four rows: -->
  <Grid.RowDefinitions>
    <RowDefinition/>
    <RowDefinition/>
    <RowDefinition/>
    <RowDefinition/>
  </Grid.RowDefinitions>
  <!-- Define two columns: -->
  <Grid.ColumnDefinitions>
    <ColumnDefinition/>
    <ColumnDefinition/>
  </Grid.ColumnDefinitions>
  <!-- Arrange the children: -->
```

```
    <Label Grid.Row="0" Grid.Column="0" Background="Blue" Foreground="White"
      HorizontalContentAlignment="Center">Start Page</Label>
    <GroupBox Grid.Row="1" Grid.Column="0" Background="White" Header="Recent Projects">...</GroupBox>
    <GroupBox Grid.Row="2" Grid.Column="0" Background="White" Header="Getting Started">...</GroupBox>
    <GroupBox Grid.Row="3" Grid.Column="0" Background="White" Header="Headlines">...</GroupBox>
    <GroupBox Grid.Row="1" Grid.Column="1" Background="White" Header="Online Articles">
      <ListBox>
        <ListBoxItem>Article #1</ListBoxItem>
        <ListBoxItem>Article #2</ListBoxItem>
        <ListBoxItem>Article #3</ListBoxItem>
        <ListBoxItem>Article #4</ListBoxItem>
      </ListBox>
    </GroupBox>
  </Grid>
```

Grid

Start Page	
Recent Projects [el]	Online Articles Article #1 Article #2
Getting Started [el]	
Headlines [el]	

Grid

- É possível posicionar os elementos e fazer com que eles se distribua em mais de uma linha ou coluna:

Grid.RowSpan="3"

Grid.ColumnSpan="2"

Grid

```
<!-- Define four rows: -->
<Grid.RowDefinitions>
  <RowDefinition Height="Auto"/>
  <RowDefinition/>
  <RowDefinition/>
  <RowDefinition/>
</Grid.RowDefinitions>
<!-- Define two columns: -->
<Grid.ColumnDefinitions>
  <ColumnDefinition Width="Auto"/>
  <ColumnDefinition/>
</Grid.ColumnDefinitions>
```

