Controls

José Antônio da Cunha

IFRN

Controles

Você já teve um contato com alguns controles do Silverlight: Os layouts. Nos slides a seguir veremos alguns dos controles fundamentais do Silverlight.

Controles

TextBlock: ele fornece uma propriedade **Tex**t, a qual aceita uma string com o texto que você deseja exibir.

```
<Grid x:Name="LayoutRoot" Background="White">
        <StackPanel Orientation="Vertical">
            <TextBlock Text="Este é o conteúdo a ser exibido."></TextBlock>
            <TextBlock x:Name="lblTexto" Text="Outra linha de texto."
Foreground="Red"></TextBlock>
            <TextBlock x:Name="lblSeg" FontFamily="Times New Roman" FontSize="18">Algum
texto</TextBlock>
            <TextBlock TextDecorations="Underline">Texto sublinhado</TextBlock>
            <TextBlock FontFamily="Georgia" FontSize="20">
                Texto neste ponto <Run FontStyle="Italic" Foreground="YellowGreen">É isto aí</Run>
a
                <Run FontFamily="Comic Sans MS" Foreground="Red"
FontSize="40">Testando</Run>
             </TextBlock>
        </StackPanel>
    </Grid>
```

Controles

Images: Mostrar uma imagem é uma tarefa simples no silverlight. Você simplesmente necessita adicionar um controle Image e configurar a propriedade Source. Entratnto, há algumas limitações que vcoê precisa saber.

A limitação mais óbvia é que o controle Image suporta justamente dois tipos de formatos. Ele tem total suporte para JPEG e PNG, mas não tem suporte para arquivos PNG que usa 64-bit color ou grayscale.

Controles

```
<Grid x:Name="LayoutRoot" Background="White">
        <StackPanel Orientation="Vertical">
            <TextBlock Text="Este é o conteúdo a ser exibido."></TextBlock>
            <TextBlock x:Name="lblTexto" Text="Outra linha de texto."
Foreground="Red"></TextBlock>
            <TextBlock x:Name="lblSeg" FontFamily="Times New Roman"
FontSize="18">Algum texto</TextBlock>
            <TextBlock TextDecorations="Underline">Texto sublinhado</TextBlock>
            <TextBlock FontFamily="Georgia" FontSize="20">
                Texto neste ponto <Run FontStyle="Italic" Foreground="YellowGreen">É
isto aí</Run> a
                <Run FontFamily="Comic Sans MS" Foreground="Red"
FontSize="40">Testando</Run>
            </TextBlock>
            <Button Margin="3" Height="50">
                <Image Source="Imagem 119.jpg"></Image>
            </Button>
            <Image Source="http://www.mysite.com/Images/grandpiano.jpg"></Image>
        </StackPanel>
    </Grid>
```

Controles

Button: O Silverlight reconhece três tipo de botões. O botão familiar (Buttom), o CheckBox e o RadionButtom. Todos derivam da classe ButtonBase.

```
<Button x:Name="btn1" Margin="3" Height="50">
            <Image Source="Imagem 119.jpg"></Image>
 </Button>

<Button Content="Absoluto Padding"></Button>

<Button Padding="3" Content="Well Padding"></Button>
```

Controles

Button: Outro exemplo

```
<Grid x:Name="LayoutRoot" Background="White">
    <Button x:Name="btn2" Margin="3" Height="100" Width="200">
        <StackPanel>
            <TextBlock Margin="3" Text="Imagem e texto"></TextBlock>
            <Image Source="Imagem 119.jpg"></Image>
        <TextBlock Margin="3" Text="Cortezia do StackPanel"></TextBlock>
        </StackPanel>
     </Button>
  </Grid>
```

Controles

Button: Outro exemplo

```
<Grid x:Name="LayoutRoot" Background="White">
        <Button Margin="3" Height="70" Width="215">
            <Grid Margin="5">
                <Polygon Points="100,25,125,0,200,25,125,50"
Fill="LightSteelBlue" />
                <Polygon Points="100,25,75,0,0,25,75,50" Fill="LightGray" />
            </Grid>
        </Button>
    </Grid>
```

Controles

CheckBox e RadioButtom: Ambos CheckBox e RadioButtom são botões de uma espécie diferente. Elas derivam ToggleButtom, o que significa que pode ser ligado ou desligado pelo usuário, daí o seu comportamento de alternância.

```
<Border Margin="5" Padding="5" BorderBrush="Red" BorderThickness="1"
CornerRadius="5">
            <StackPanel>
                <CheckBox IsChecked="{x:Null}" Content="Checkbox em um
determinado estado"></CheckBox>
                <CheckBox Content="Checkbox em um determinado
estado"></CheckBox>
                <CheckBox IsChecked="True" Content="Checkbox em um determinado
estado"></CheckBox>
            </StackPanel>
        </Border>
```

Controles

Exemplo com RadioButtom

```
 <Border Margin="5" Padding="5" BorderBrush="Yellow" BorderThickness="1"
CornerRadius="5">
            <StackPanel>
               <RadioButton Content="Grupo 1"></RadioButton>
               <RadioButton Content="Grupo 1"></RadioButton>
               <RadioButton Content="Grupo 1"></RadioButton>
               <RadioButton GroupName="Grupo3" Content="Grupo
3"></RadioButton>
            </StackPanel>
         </Border>
```

Controles

Tooltips: você usa o ToolTipService para configurar um tooltip para um existente elemento.

```
 <Border Margin="5" Padding="5" BorderBrush="Bisque" BorderThickness="1"
CornerRadius="5">
            <StackPanel>
               <Button ToolTipService.ToolTip="Isto é um tooltip"
Content="OK"></Button>
            </StackPanel>
         </Border>
```

Controles

Tooltips: outro exemplo

```
 <Border Margin="5" Padding="5" BorderBrush="AliceBlue" BorderThickness="1"
CornerRadius="5">
        <StackPanel>
            <Button Content="Eu tenho uma fantasia tooltip">
                <ToolTipService.ToolTip>
                 <StackPanel>
                     <TextBlock Margin="3" Text="Imagem e texto"></TextBlock>
                     <Image Source="Imagem 119.jpg"></Image>
                     <TextBlock Margin="3" Text="Imagem e texto"></TextBlock>
                 </StackPanel>
                 </ToolTipService.ToolTip>
            </Button>
       </StackPanel>
</Border>
```

Controles

Popup: O controle Popup tem muito em comum com o controle ToolTip, embora nem uma deriva da outra.

```
<Grid x:Name="LayoutRoot" Background="White">
        <StackPanel Margin="20">
            <TextBlock TextWrapping="Wrap"
MouseLeftButtonDown="TextBlock_MouseLeftButtonDown"
                       Text="Clique aqui para abrir o Popup."></TextBlock>
            <Popup x:Name="popUp" MaxWidth="200">
                <Border Background="Linen"
MouseLeftButtonDown="Border_MouseLeftButtonDown">
                    <TextBlock Margin="10" Text="Isto é um Popup."></TextBlock>
                </Border>
            </Popup>
        </StackPanel>
    </Grid>
```

Controles

Popup: código C#

```
public partial class Popup : UserControl
   {
      ...

       private void TextBlock_MouseLeftButtonDown(object sender,
MouseButtonEventArgs e)
       {
           popUp.IsOpen = true;
       }

       private void Border_MouseLeftButtonDown(object sender,
MouseButtonEventArgs e)
       {
           popUp.IsOpen = false;
       }
   }
```

Controles

ListBox: Para adicionar itens para uma lista, você pode aninhar elementos dentro do controle ListBox. Por exemplo, aqui temos uma lista de cores:

```
<Grid x:Name="LayoutRoot" Background="White">
        <ListBox>
            <ListBoxItem Content="Green"></ListBoxItem>
            <ListBoxItem Content="Blue"></ListBoxItem>
            <ListBoxItem Content="Yellow"></ListBoxItem>
            <ListBoxItem Content="Red"></ListBoxItem>
        </ListBox>
    </Grid>
```

Controles

ListBox: outro exemplo

```
<ListBox>
        <ListBoxItem>
            <Image Source="happyface.jpg"></Image>
        </ListBoxItem>
        <ListBoxItem>
            <Image Source="happyface.jpg"></Image>
        </ListBoxItem>
</ListBox>
```

Controles

ListBox: outro exemplo

```
<Grid x:Name="LayoutRoot" Background="White">
        <ListBox>
            <StackPanel Orientation="Horizontal">
                <Image Source="Imagem 119.jpg" Width="30" Height="30"></Image>
                <TextBlock VerticalAlignment="Center" Text="Um cão feliz"></TextBlock>
            </StackPanel>
            <StackPanel Orientation="Horizontal">
                <Image Source="Imagem 119.jpg" Width="30" Height="30"></Image>
                <TextBlock VerticalAlignment="Center" Text="Um cão feliz"></TextBlock>
            </StackPanel>
            <StackPanel Orientation="Horizontal">
                <Image Source="Imagem 119.jpg" Width="30" Height="30"></Image>
                <TextBlock VerticalAlignment="Center" Text="Um cão feliz"></TextBlock>
            </StackPanel>
        </ListBox>

    </Grid>
```

Controles

ComboBox: é similar ao ListBox

```
<Grid x:Name="LayoutRoot" Background="White">
        <ComboBox Height="50" Width="200">
            <ComboBoxItem Content="Opção 1"></ComboBoxItem>
            <ComboBoxItem Content="Opção 2"></ComboBoxItem>
            <ComboBoxItem Content="Opção 3"></ComboBoxItem>
            <ComboBoxItem Content="Opção 4"></ComboBoxItem>
        </ComboBox>
    </Grid>
```

Controles

TabControl: No Silverlight, o TabControl é um controle de itens que contém um ou mais elementos TabItem.

```
<sdk:TabControl>
        <sdk:TabItem>
           <sdk:TabItem.Header>
              <StackPanel>
                 <TextBlock Margin="3">Image and Text Title</TextBlock>
                 <Image Source="Imagem 119.jpg" Stretch="None" Height="30" Width="30"/>
              </StackPanel>
           </sdk:TabItem.Header>
           <StackPanel Margin="3">
              <CheckBox Margin="3" Content="Setting 1"></CheckBox>
              <CheckBox Margin="3" Content="Setting 2"></CheckBox>
              <CheckBox Margin="3" Content="Setting 3"></CheckBox>
           </StackPanel>
        </sdk:TabItem>
        <sdk:TabItem Header="Tab dois">
           ....
           </StackPanel>
        </sdk:TabItem>
     </sdk:TabControl>
```

Controles

Text Controls: Silverlight inclui um controle TextBox padrão que suporta muitos dos recursos do mundo Windows, incluindo o deslocamento, quebra automática de texto, corte-e-colar, e seleção.

```
<Canvas>
        <TextBlock x:Name="userNameLabel" Canvas.Left="0" Canvas.Top="52"
Text="User Name:" FontFamily="Arial" />
        <TextBlock x:Name="passwordLabel" Canvas.Left="0" Canvas.Top="82"
Text="Password:" FontFamily="Arial" />
        <TextBox x:Name="userName" Canvas.Top="50" Canvas.Left="80"
Width="150" MaxLength="20" TabIndex="0" />
        <PasswordBox x:Name="password" Canvas.Top="80" Canvas.Left="80"
Width="150" TabIndex="1" />
    </Canvas>
```

Controles

Text Controls: outro exemplo com TextBox autocompletar

```
<UserControl x:Class="ControlesSilverlightApplication.TextBox"
   xlns:input="clr-
Namespace:System.Windows.Controls;assembly=System.Windows.Controls.Input"
>

    <Canvas>
        <TextBlock x:Name="userNameLabel" Canvas.Left="0" Canvas.Top="52" Text="User Name:"
FontFamily="Arial" />
        <TextBlock x:Name="passwordLabel" Canvas.Left="0" Canvas.Top="82" Text="Password:"
FontFamily="Arial" />
        <TextBlock x:Name="autoCompletar" Canvas.Left="0" Canvas.Top="115"
Text="AutoCompletar:" FontFamily="Arial" />
        <TextBox x:Name="userName" Canvas.Top="50" Canvas.Left="80" Width="150"
MaxLength="20" TabIndex="0" />
        <PasswordBox x:Name="password" Canvas.Top="80" Canvas.Left="80" Width="150"
TabIndex="1" />
        <input:AutoCompleteBox x:Name="txtMonth" Canvas.Left="80" Canvas.Top="112"
Width="100"></input:AutoCompleteBox>
    </Canvas>
</UserControl>
```

Controles

Controles baseados em intervalo: Silverlight inclui três controles que usam o conceito de um intervalo. Estes controles tomam um valor numérico que fica entre um mínimo específico e um valor máximo. esses controles - ScrollBar, Slider, e ProgressBar – Derivam da classe RangeBase (que derivam da classe Controle).

Controles

Slider: exemplo

```
<Canvas x:Name="LayoutRoot" Background="LightGray" >

      <Ellipse x:Name="mcCircle" Width="200" Height="200" Canvas.Left="60" Canvas.Top="20"
            Fill="Gray" Stroke="Black" StrokeThickness="2">
       </Ellipse>

       <Slider x:Name="RedSlider" Width="300" Height="20" Background="Red" Maximum="255"
Minimum="0"
            Canvas.Left="30" Canvas.Top="240" ValueChanged="RedSlider_ValueChanged"/>

       <Slider x:Name="GreenSlider" Width="300" Height="20" Background="Green"
Maximum="255" Minimum="0"
            Canvas.Left="30" Canvas.Top="270" ValueChanged="GreenSlider_ValueChanged"/>

       <Slider x:Name="BlueSlider" Width="300" Height="20" Background="Blue" Maximum="255"
Minimum="0"
             Canvas.Left="30" Canvas.Top="300" ValueChanged="BlueSlider_ValueChanged"/>
   </Canvas>
```

Controles

Slider: código C#

```
private void RedSlider_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)
    {
        UpdateCircleWithColors();
    }

    private void GreenSlider_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)
    {
        UpdateCircleWithColors();
    }

    private void BlueSlider_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)
    {
        UpdateCircleWithColors();
    }

    private void UpdateCircleWithColors()
    {
        Color clr = Color.FromArgb(255, Convert.ToByte(RedSlider.Value),

            Convert.ToByte(GreenSlider.Value), Convert.ToByte(BlueSlider.Value));

        mcCircle.Fill = new SolidColorBrush(clr);
    }
```