

Aula 4 - Data Binding

Resumo:

- Exemplo mais completo de *data binding*
- Data binding a um ficheiro xml
- Validação em WPF

Este guião pressupõe a resolução dos guiões das últimas duas aulas. Se não tiver concluído a aula 03, pode começar na secção 4.2 utilizando o código da aula 2 até a alínea 2.5 (está disponível uma possível solução no site da disciplina).

4.1. DataGrid

Substitua na interface (se estiver a usar o exemplo fornecido) os dois `labels` “Disciplina” e “ECTS” por uma `DataGrid`. Para criar a `DataGrid` pode usar o código seguinte:

```
<DataGrid Grid.Column="1" Grid.Row="2" >
    <DataGrid.Columns>
        <DataGridTextColumn Header="Disciplina"/>
        <DataGridTextColumn Header="ECTS"/>
    </DataGrid.Columns>
</DataGrid>
```

Aplique a `DataGrid` o estilo `columnHeaderStyle` (associe o estilo ao atributo `ColumnHeaderStyle` e não `Style`) que segue:

```
<!-- DataGrid header style -->
<Style x:Key="columnHeaderStyle" TargetType="{x:Type
DataGridColumnHeader}">
    <Setter Property="Height" Value="35" />
    <Setter Property="Padding" Value="5" />
    <Setter Property="Background" Value="#4E87D4" />
    <Setter Property="Foreground" Value="White" />
</Style>
```

4.2. Data Binding a dados em formato xml

Na página `DetiHome.xaml`, adicione no elemento `<Grid>` o código seguinte que define a informação associada a cada curso. Neste caso é usado um `XmlDataProvider` para fornecer a informação que poderia ser carregada de um ficheiro `xml` ou de uma

Base de Dados. Se estiver a usar o data binding da última aula, retire o código associado dado que o binding vai agora ser feito a dados em XML.

```
<Grid.Resources>
<!--Dados cursos 1º ano 1º semestre-->
  <XmlDataProvider x:Key="FonteDadosDeti" XPath="Cursos">
    <x:XData>
      <Cursos xmlns="">
        <Curso Nome="Computadores e Telemática"Codigo="8240">
          <UC Disciplina="Álgebra Linear e Geometria Analítica" ECTS="6" />
          <UC Disciplina="Cálculo I" ECTS="6" />
          <UC Disciplina="Programação I" ECTS="8" />
          <UC Disciplina="Introdução aos Sistemas Digitais" ECTS="6" />
        </Curso>
        <Curso Nome="Electrónica e Telecomunicações"Codigo="8204">
          <UC Disciplina="Álgebra Linear e Geometria Analítica" ECTS="6" />
          <UC Disciplina="Cálculo I" ECTS="6" />
          <UC Disciplina="Programação I" ECTS="8" />
          <UC Disciplina="Introdução aos Sistemas Digitais" ECTS="6" />
        </Curso>
        <Curso Nome="Engenharia Informática"Codigo="8295">
          <UC Disciplina="Álgebra Linear e Geometria Analítica" ECTS="6" />
          <UC Disciplina="Cálculo I" ECTS="6" />
          <UC Disciplina="Elementos de Física" ECTS="8" />
          <UC Disciplina="Introdução às Tecnologias Web" ECTS="6" />
          <UC Disciplina="Fundamentos de programação" ECTS="6" />
        </Curso>
      </Cursos>
    </x:XData>
  </XmlDataProvider>
</Grid.Resources>
```

Utilize um data template para indicar que informação usar para mapear na listbox. Pode definir este DataTemplate também nos Grid Resources da página.

```
<DataTemplate x:Key="nameItemTemplate">
  <Label Content="{Binding XPath=@Nome}"/>
</DataTemplate>
```

Modifique a listbox no DETI-Home para considerar os dados fornecidos através do código seguinte que indica a fonte de dados a usar (itemSource) e o formato a aplicar aos dados (ItemTemplate).

```
<ListBox Name="cursosListBox" Grid.Column="1" Grid.Row="2"
ItemsSource="{Binding Source={StaticResource FonteDadosDeti}, XPath=Curso}"
ItemTemplate="{StaticResource nameItemTemplate}">
</ListBox>
```

Este código define a fonte de dados (FonteDadosDeti) e usa um (nameItemTemplate) para garantir que o nome é apresentado para todos os elementos (cursos) disponíveis.

4.3. Associação entre dados e controlos (Data binding)

É agora necessário usar a informação para mapear as disciplinas corretas para cada curso. Para tal, é necessário passar os dados a página DETI-Cursos. Isso pode ser feito através do construtor usando o código seguinte que indica que o DataContext da página contém todos os dados definidos no Resource.

Note que se tiver algum código associado ao clique no botão deve retirar o mesmo.

```
// Construtor para a classe Deti-Cursos
public DETI_Cursos(object data) : this()
{
    // Associa os dados ao contexto da nova página.
    this.DataContext = data;
}
```

A passagem dos dados a esta página pode ser feita agora ao clicar no botão “ver” com o código seguinte:

```
// Vê página Disciplinas
DETI_Cursos PaginaCursos= new DETI_Cursos
(this.cursosListBox.SelectedItem);
this.NavigationService.Navigate(PaginaCursos);
```

Falta agora fazer o mapeamento entre os dados e os controlos na página dos cursos. Comece por associar nome do curso seleccionado ao label Disciplina através de Data Binding:

```
<Label Style="{StaticResource labelStyle}" Content="{Binding
XPath=@Nome}"></Label>
```

Finalmente pode associar a informação das disciplinas e dos ECTS a DataGrid note que se não colocar a opção AutoGenerate a false, as colunas são geradas automaticamente de acordo com a fonte de dados.

```
<DataGrid ItemsSource="{Binding XPath=UC}" AutoGenerateColumns="False" >
    <DataGrid.Columns>
        <DataGridTextColumn Header="Disciplina" Binding="{Binding
XPath=@Disciplina}" />
        <DataGridTextColumn Header="ECTS" Binding="{Binding XPath=@ECTS}" />
    </DataGrid.Columns>
</DataGrid>
```

Mais informação sobre o binding em wpf pode ser encontrada em:

[http://msdn.microsoft.com/en-us/library/ms752347\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/ms752347(v=vs.110).aspx)
<http://msdn.microsoft.com/en-us/library/aa480224.aspx>

4.4. Leitura de um ficheiro xml

Modifique o código para ler informação directamente de um ficheiro xml (use por exemplo o ficheiro deti.xml). Para tal modifique o `xmlDataProvider` para usar o código seguinte:

```
<XmlDataProvider x:Key="FonteDadosDeti"
Source="Deti.xml" XPath="Cursos"/>
```

Adicione o ficheiro Deti.xml ao projeto e execute o programa. Modifique alguns parâmetros do ficheiro e veja o resultado.

Modifique o `Datatemplate nameItemTemplate` para mostrar na listbox o código do curso seguindo do nome. Para tal pode adicionar um `StackPanel` com orientação horizontal colocando no mesmo vários `textblock` com o texto associado ao binding correto. Adiciona dessa forma também o código no label disciplina na página DETI-Cursos.

4.5. Validação em WPF

Acrescente, numa posição a sua escolha da grelha da página DETI_Home, uma textbox para especificar o nº do curso a seleccionar. Vamos adicionar o código para validar o conteúdo garantindo que a textbox tem o código de um dos 3 cursos.

Comece por adicionar no código a função de validação herdando a classe ValidationRule. Pode usar o código seguinte:

```
public class NameValidator : ValidationRule
{
    public override ValidationResult Validate (object value,
        System.Globalization.CultureInfo cultureInfo)
    {
        if (value.ToString() != "8240" && value.ToString() != "8204" &&
            value.ToString() != "8295")
            return new ValidationResult(false, "Tem que usar um código
válido");

        return ValidationResult.ValidResult;
    }
}
```

Adicione um estilo e um controlTemplate no resources da grid xaml para indicar a ocorrência de um erro, por exemplo usando o código seguinte:

```
<Style x:Key="textBoxInError" TargetType="{x:Type TextBox}">
    <Style.Triggers>
        <Trigger Property="Validation.HasError" Value="true">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource.Self},
                    Path=(Validation.Errors)[0].ErrorContent}"/>
        </Trigger>
    </Style.Triggers>
</Style>

<ControlTemplate x:Key="validationTemplate">
    <DockPanel>
        <TextBlock Foreground="Red" FontSize="20">!</TextBlock>
        <AdornedElementPlaceholder/>
    </DockPanel>
</ControlTemplate>
```

É agora possível fazer o binding da caixa de texto a regra de validação com código seguinte na textbox:

```
<TextBox Width="100" HorizontalAlignment="Left"
    Validation.ErrorTemplate="{StaticResource validationTemplate}"
    Style="{StaticResource textBoxInError}">
    <TextBox.Text>
        <Binding Source="{StaticResource myCurso}" Path="CodCurso"
UpdateSourceTrigger="PropertyChanged">
        <Binding.ValidationRules>
            <local:NameValidator/>
        </Binding.ValidationRules>
    </Binding>
    </TextBox.Text>
</TextBox>
```

Repare que o binding é feito ao código do curso da classe Curso definida na última aula. Para usar esse código deve instanciar a classe curso no xaml usando:

```
<local:Curso x:Key="myCurso"/>
```