

05

Binding de Text e SwitchCell

Transcrição

Até o momento, nossa tela se encontra muito parecida com o que foi desenhado no início do projeto. Porém, ainda precisamos fazer a página de detalhes funcionar corretamente; ao selecionarmos uma das opções de acessórios, o valor total não muda, pois o deixamos como um texto fixo.

É preciso definirmos os valores dos acessórios, modificando a aplicação para colocá-los em constantes, a serem somados ao valor do veículo selecionado, totalizando o valor final.

Vamos abrir o *code behind*, `DetalheView.xaml.cs`, para declarar uma constante privada e inteira para cada acessório:

```
public partial class DetalheView : ContentPage
{
    private const int FREIO_ABS = 800;
    private const int AR_CONDICIONADO = 1000;
    private const int MP3_PLAYER = 500;
}
```

Feito isto, estes valores serão exibidos na tela, usando as constantes que acabamos de declarar. Como já vimos antes, é possível utilizar o `Binding` para exibir no XAML algumas propriedades que estão no *code behind*.

Criaremos, então, três propriedades em `DetalheView.xaml.cs`, que irão armazenar o texto a ser mostrado em nossa página:

```
public Veiculo Veiculo { get; set; }

public string TextoFreioABS
{
    get
    {
        return string.Format("Freio ABS - R$ {0}", FREIO_ABS);
    }
}
```

A propriedade utilizada acima (`TextoFreioABS`) precisa ser colocada em nosso *view*, no XAML.

```
<TableSection Title="Acessórios">
    <SwitchCell Text="{Binding TextoFreioABS}" On="True"></SwitchCell>
//...
</TableSection>
```

Vamos rodar a aplicação e ver o que acontece. O acessório "Freio ABS" custa R\$ 800,00, um valor de propriedade proveniente da classe correspondente no *code behind*, pelo `Binding`. Faremos o mesmo para os outros `SwitchCell`s, em `DetalheView.xaml`.

E também tiraremos o `On` do primeiro acessório, para que o usuário possa escolher de fato o que quer ativar ou não.

```
<TableSection Title="Acessórios">
    <SwitchCell Text="{Binding TextoFreioABS}"></SwitchCell>
    <SwitchCell Text="{Binding TextoArCondicionado}"></SwitchCell>
    <SwitchCell Text="{Binding TextoMP3Player}"></SwitchCell>
    <TextCell Text="Total: R$ 50000"></TextCell>
</TableSection>
```

No *code behind*, declararemos as propriedades para o ar-condicionado e para o mp3 player:

```
public Veiculo Veiculo { get; set; }

public string TextoFreioABS
{
    get
    {
        return string.Format("Freio ABS - R$ {0}", FREIO_ABS);
    }
}

public string TextoArCondicionado
{
    get
    {
        return string.Format("Ar Condicionado - R$ {0}", AR_CONDICIONADO);
    }
}

public string TextoMP3Player
{
    get
    {
        return string.Format("MP3 Player - R$ {0}", MP3_PLAYER);
    }
}
```

Com isto, podemos rodar a aplicação mais uma vez, conferindo as alterações. O layout fica assim, com todas as configurações feitas a partir do `Binding`:

