03

Propriedades

Transcrição

Começaremos a trabalhar com a página `MasterDetailPage` de nome `MasterDetailView`, alterando sua estrutura para podermos colocar as duas propriedades desta `MasterDetailPage`, a página mestra e de detalhe. O primeiro ponto a ser alterado é o `Label`, que está "sobrando", pois ele não pode ser incluído diretamente dentro de uma estrutura de um `MasterDetailPage`. Uma das maneiras de se definir uma propriedade no XAML é inclui-la dentro de uma tag, algo como:

```
<MasterDetailPage xmlns="http://xamarin.com/schemas/2014/forms"
                  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
                  x:Class="TestDrive.Views.MasterDetailView" Master="" Detail="">
</MasterDetailPage>
```

Porém, existem maneiras melhores de se fazer isto, trabalhando com estruturas dentro do XAML, a partir da cópia do nome da tag mãe (ou pai, como preferir), acrescentando-se a propriedade `Master`, dentro da qual colocaremos a definição do que será a instância do mestre da página `MasterDetailPage`. Faremos o mesmo com `Detail`.

```
<MasterDetailPage xmlns="http://xamarin.com/schemas/2014/forms"
                  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
                  x:Class="TestDrive.Views.MasterDetailView">
  <MasterDetailPage.Master>
    <!--perfil do usuário-->
  </MasterDetailPage.Master>
  <MasterDetailPage.Detail>
    <!--listagem de veículos (navegação)-->
  </MasterDetailPage.Detail>
</MasterDetailPage>
```

Para definirmos a página mestra (de perfil do usuário), normalmente criaríamos uma página de perfil, fazendo a referência dentro dela mesma. Para facilitar nosso trabalho, como ainda não temos esta página de perfil, vamos criá-la *inline* (dentro do próprio código XAML), declarando uma página como se fosse outro arquivo, porém com a estrutura dentro do próprio arquivo.

Ao observarmos os tipos de visualização (*views*) que temos no Xamarin, a maneira mais fácil para se criar uma página é justamente o `ContentPage`, que colocaremos na parte de perfil do usuário:

```
<MasterDetailPage xmlns="http://xamarin.com/schemas/2014/forms"
                  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
                  x:Class="TestDrive.Views.MasterDetailView">
  <MasterDetailPage.Master>
    <ContentPage>
      <Label Text="Este é o perfil do usuário"></Label>
    </ContentPage>
  </MasterDetailPage.Master>
  <MasterDetailPage.Detail>
    <!--listagem de veículos (navegação)-->
```

```
  </MasterDetailPage.Detail>
</MasterDetailPage>
```

A definição da página de detalhe normalmente seria feita no *code behind* também, no código C# de `MasterDetailView.xaml.cs`, assim:

```
using Xamarin.Forms;

namespace TestDrive.Views
{
  public partial class MasterDetailView : MasterDetailPage
  {
    public MasterDetailView()
    {
      InitializeComponent();

      this.Detail = new ListagemView();
    }
  }
}
```

No entanto, declararemos a `ListagemView` dentro do próprio XAML, deixando o código assim:

```
using Xamarin.Forms;

namespace TestDrive.Views
{
  public partial class MasterDetailView : MasterDetailPage
  {
    public MasterDetailView()
    {
      InitializeComponent();
    }
  }
}
```

Em `MasterDetailView.xaml`, criaremos uma instância nova da `ListagemView`, inserindo uma referência para ela... Porém, falta o Intellisense mostrar essa opção. Este objeto faz parte do projeto, no entanto, ainda está inacessível, porque o Xamarin Forms desconhece onde o objeto deve ser procurado, esta classe. Por isso, definiremos seu *namespace*, fazendo uma declaração `xmlns` (*namespace* do XML), determinando seu prefixo, que será `view`, onde se declara o `ListagemView`.

```
<MasterDetailPage xmlns="http://xamarin.com/schemas/2014/forms"
                  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
                  x:Class="TestDrive.Views.MasterDetailView">
  <MasterDetailPage.Master>
    <ContentPage>
      <Label Text="Este é o perfil do usuário"></Label>
    </ContentPage>
  </MasterDetailPage.Master>
  <MasterDetailPage.Detail>
    <view:ListagemView></view:ListagemView>
```

```
  </MasterDetailPage.Detail>
</MasterDetailPage>
```

Após fazermos as alterações, ao rodarmos a aplicação, veremos que ocorrerá um erro: `ContentPage` é uma página (dentro do `Master`) que exige um título. Faremos isto voltando ao arquivo `MasterDetailView` :

```
<MasterDetailPage xmlns="http://xamarin.com/schemas/2014/forms"
                  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
                  x:Class="TestDrive.Views.MasterDetailView">
  <MasterDetailPage.Master>
    <ContentPage Title="Perfil">
      <Label Text="Este é o perfil do usuário"></Label>
    </ContentPage>
  </MasterDetailPage.Master>
  <MasterDetailPage.Detail>
    <view:ListagemView></view:ListagemView>
  </MasterDetailPage.Detail>
</MasterDetailPage>
```

Rodaremos a aplicação mais uma vez. Desta vez, como definimos o detalhe do `MasterDetailPage` como `ListagemView` , fomos redirecionados à página de listagem de veículos. No entanto, é difícil visualizar o perfil do usuário, porque falta um ícone que indique a navegação até ele.

Se dermos uma olhada mais de perto, podemos puxar a borda esquerda para o centro, deixando a página de listagem mais escura, como se estivéssemos puxando uma cortina. Assim, exibiremos a página de perfil (a mestra). No topo, lê-se: "Esta é a página do usuário", conforme determinamos anteriormente.

Conseguimos implementar de forma bem simples e rudimentar a página `MasterDetailPage` .