

## NavigationPage

### Transcrição

Definimos como detalhe do `MasterDetailPage` uma instância da `ListagemView`. Ao logarmos na app com os dados "joao@alura.com.br" e "alura123" ("Usuário" e "Senha"), iremos à página de detalhe do `MasterDetailPage`, uma instância de `ListagemView`.

Quando chegamos nesta página, estamos acessando `ListagemView` diretamente, diferente de como era feito antes, por meio de um `NavigationPage`. Qual a consequência disto? Quando clicamos em uma opção para irmos à próxima página, a aplicação aponta erro de exceção não tratada.

Isto ocorre porque não temos uma pilha de navegação criada, então, quando vamos à página seguinte a partir da `ListagemView`, o Xamarin Forms indica o erro. Portanto, colocaremos de volta `NavigationPage` dentro da estrutura da aplicação, uma vez que é impossível chamar diretamente a `ListagemView`.

Chamaremos um `NavigationPage`, dentro do qual passaremos como argumento `ListagemView`, a página raiz da navegação neste caso.

Como utilizaremos um argumento dentro de um objeto no XAML? Existe um recurso chamado `Arguments` que faz parte do Xamarin Forms. A partir dele, colocamos os parâmetros do construtor da classe, neste caso, o `NavigationPage`.

```
<MasterDetailPage.Master>
  <ContentPage Title="Perfil">
    <Label Text="Este é o perfil do usuário"></Label>
  </ContentPage>
</MasterDetailPage.Master>
<MasterDetailPage.Detail>
  <NavigationPage>
    <x:Arguments>
      <view:ListagemView></view:ListagemView>
    </x:Arguments>
  </NavigationPage>
</MasterDetailPage.Detail>
```

Feito isto, rodaremos a aplicação verificando se o `NavigationPage` foi reestabelecido. No topo do aplicativo, foi incluída a barra de navegação que tínhamos anteriormente. Ao clicarmos em alguma opção, seguiremos à próxima página, pois recolocamos o `NavigationPage` dentro do `Detail` do `MasterDetailPage`.

O "menu hambúrguer" também aparece, o qual traz por meio do clique a página mestra (de perfil do usuário).



Agora, definiremos uma nova página mestra do `MasterDetailPage`. Removeremos a tag `ContentPage`, deixando o código assim:

```
<MasterDetailPage.Master>
</MasterDetailPage.Master>
<MasterDetailPage.Detail>
  <NavigationPage>
    <x:Arguments>
      <view:ListagemView></view:ListagemView>
    </x:Arguments>
  </NavigationPage>
</MasterDetailPage.Detail>
```

Com o lado direito do mouse, clicaremos em "Views" localizado no lado direito do Visual Studio, selecionando "Add > New Item", chamando a nova página de `MasterView.xaml` e definindo-a como mestra, que aparecerá no início da navegação.

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="TestDrive.Views.MasterView">
  <Label Text="João da Silva" VerticalOptions="Center" HorizontalOptions="Center" />
</ContentPage>
```

Voltaremos ao arquivo `MasterDetailView.xaml`, acrescentando dentro das tags referentes a `Master`, `MasterView`, que não aparecem como opção pelo Intellisense, já que não demos o *namespace* correto para o Xamarin Forms localizar esta classe.

Precisamos definir primeiramente um prefixo:

```

<MasterDetailPage.Master>
  <view:MasterView></view:MasterView>
</MasterDetailPage.Master>
<MasterDetailPage.Detail>
  <NavigationPage>
    <x:Arguments>
      <view:ListagemView></view:ListagemView>
    </x:Arguments>
  </NavigationPage>
</MasterDetailPage.Detail>

```

Feito isto, vamos rodar a aplicação. Mais uma vez, esquecemos de colocar um título na `ContentPage` :

```

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="TestDrive.Views.MasterView" Title="Perfil">
  <Label Text="João da Silva" VerticalOptions="Center" HorizontalOptions="Center" />
</ContentPage>

```

Ao rodarmos a aplicação de novo, veremos que tudo funciona conforme esperado. Clicaremos no menu para tentar entrar na página de perfil. É exibida a informação configurada acerca do usuário: "João da Silva" centralizado. Começaremos a colocar o conteúdo desta página, não bastando colocarmos simplesmente um nome fixo dentro do `Label` .

Consultando a especificação recebida pela Aluracar, temos nome do usuário, e-mail, e botões de comando a serem utilizados pelo usuário: "Perfil", "Meus agendamentos" e "Novo agendamento". Criaremos o `StackLayout` para empilhar verticalmente estes controles. Teremos um `Label` para o nome do usuário e outro para o e-mail:

```

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="TestDrive.Views.MasterView" Title="Perfil">
  <StackLayout>
    <Label Text="João da Silva"></Label>
    <Label Text="joao@alura.com.br"></Label>
  </StackLayout>
</ContentPage>

```

Vamos rodar a aplicação e conferir seu funcionamento. Realizamos login com sucesso, vimos que as informações referentes ao usuário foram colocadas, porém, aparecem no topo do layout. Vamos centralizá-las, modificando o `StackLayout` :

```

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="TestDrive.Views.MasterView" Title="Perfil">
  <StackLayout VerticalOptions="Center" HorizontalOptions="Center">
    <Label Text="João da Silva"></Label>
    <Label Text="joao@alura.com.br"></Label>
  </StackLayout>
</ContentPage>

```

Rodando a aplicação mais uma vez, logamos e navegamos até a página de perfil que acabamos de configurar:



Assim, vimos tudo que é necessário para implementarmos um `MasterDetailPage` em uma aplicação Xamarin Forms, como criar uma página mestra e outra, de detalhe. Mostramos como implementar `NavigationPage` dentro desta que nos permite a navegação da forma como fazíamos anteriormente.