

Page Title, parâmetro de construtor da página e AgendamentoView

Transcrição

Continuaremos implementando a navegação das páginas. Na segunda delas, teremos como título o nome do modelo do veículo selecionado e, no fim da página, temos um botão com o texto "Próximo".

Faremos alterações no arquivo `DetalheView.xaml.cs` para recebermos um objeto contendo o nome do veículo a ser exibido no topo do layout, como título da página. O construtor do arquivo receberá um parâmetro chamado `veiculo`.

```
namespace TestDrive.Views
{
    public partial class DetalheView : ContentPage
    {
        public DetalheView (Veiculo veiculo)
        {
            InitializeComponent();
            this.Title = veiculo.Nome;
        }
    }
}
```

Com a alteração do construtor, faremos o mesmo em relação a quem o chama também, para passar o veículo selecionado pelo usuário como parâmetro. Entraremos em `ListagemView.xaml.cs`, em que fazemos esta chamada, e instanciamos `DetalheView`, em que passaremos o veículo que está sendo selecionado:

```
Navigation.PushAsync(new DetalheView(veiculo));
```

Rodaremos a aplicação para verificar o que acontece. Clicando em "HB20 S", entramos na página de detalhes, com apenas o nome do veículo como título. Ainda falta colocarmos o botão que permitirá a navegação para a página seguinte.

Para isto, clicaremos em `DetalheView.xaml`, acrescentando depois do `Label` o controle `Button`, o qual precisa de um nome, que será `buttonProximo`.

```
<Button x:Name="buttonProximo" Text="Próximo"></Button>
```

Criaremos a próxima página, a terceira, para onde o usuário navegará após o clique neste botão que acabamos de criar. Isto será feito clicando-se com o lado direito do mouse em `Views` e selecionando "*Add > New Item*" e depois "Forms Xaml Page".

Esta última página terá dados cadastrais do usuário para o agendamento do *Test Drive*, portanto, iremos nomeá-la "AgendamentoView". Agora, faremos a navegação da página de detalhe para a página de agendamento utilizando um evento no botão (`Clicked`), em `DetalheView.xaml`:

```
<Button x:Name="buttonProximo" Text="Próximo" Clicked="buttonProximo_Clicked"></Button>
```

Assim, no *code behind* da classe que representa esta página cria-se o método que podemos acessar, programando-se uma navegação para a terceira página, de agendamento do *Test Drive*.

Como foi visto anteriormente, existe um objeto de navegação disponível no Xamarin Forms chamado *Navigation*, o qual chamará o método que permitirá a navegação para a próxima página. Em *DetalheView.xaml.cs* teremos:

```
namespace TestDrive.Views
{
    public partial class DetalheView : ContentPage
    {
        public DetalheView(Veiculo veiculo)
        {
            InitializeComponent();
            this.Title = veiculo.Nome;
        }

        private void buttonProximo_Clicked(object sender, EventArgs e)
        {
            Navigation.PushAsync(new AgendamentoView());
        }
    }
}
```

Feito isto, executaremos a app para verificar seu funcionamento. Com isto, esperamos poder navegar entre as três páginas da nossa aplicação. Abrindo-se a app, clicamos em um veículo qualquer, "Azera V6", por exemplo, e aparece o botão "Próximo" expandido pela tela toda, teremos que corrigir isto mexendo no layout da página.

Entraremos em *DetalheView.xaml* e modificaremos o texto de *Label* :

```
<Label Text="Esta é a página de detalhe" VerticalOptions="Center" HorizontalOptions="Center" />
```

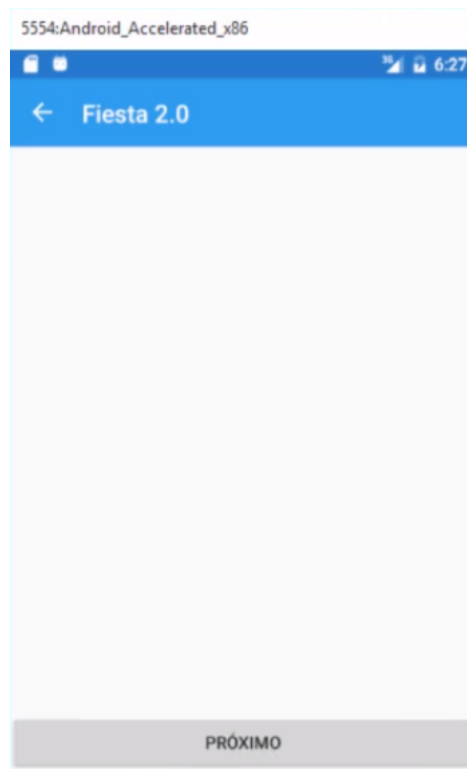
Temos dois *Options*, vertical e horizontal, que definem a posição do *Label* na página, portanto deixaremos o código assim:

```
<Label Text="Esta é a página de detalhe" VerticalOptions="CenterAndExpand" HorizontalOptions="C
```

Queremos também que o botão ocupe apenas o fim da página, portanto seu código ficará assim:

```
<Button x:Name="buttonProximo" Text="Próximo" Clicked="buttonProximo_Clicked" VerticalOptions="I
```

Feito isto, rodaremos a aplicação e selecionaremos "Fiesta 2.0" para navegarmos à próxima página. Desta vez, o botão aparece onde gostaríamos:



Clicando nele, somos redirecionados a uma terceira página, ainda sem título ou conteúdo, de agendamento. Vamos abrir `AgendamentoView.xaml.cs` para alterar o construtor para que ele também receba `veiculo` como parâmetro. Settaremos o título da página de agendamento para *Test Drive*.

```
public partial class AgendamentoView : ContentPage
{
    public AgendamentoView(Veiculo veiculo)
    {
        InitializeComponent();
        this.Title = veiculo.Nome;
    }
}
```

No entanto, isto não é o suficiente, porque precisaremos passar o parâmetro novo para chamada que está iniciando esta página. Iremos então acessar `DetalheView.xaml.cs`, em que programamos o evento de clique no botão "Próximo", passando o `veiculo`.

Porém, ainda não temos como acessá-lo neste momento, então, modificaremos a classe `DetalheView` para inclusão de uma propriedade chamada `veiculo`, o qual receberá o parâmetro que está chegando a esta página por meio do parâmetro no construtor.

```
public partial class DetalheView : ContentPage
{
    public Veiculo Veiculo { get; set; }

    public DetalheView(Veiculo veiculo)
    {
        InitializeComponent();
        this.Title = veiculo.Nome;
        this.Veiculo = veiculo;
    }
}
```

```
private void buttonProximo_Clicked(object sender, EventArgs e)
{
    Navigation.PushAsync(new AgendamentoView(this.Veiculo));
}
}
```

Rodando a aplicação e selecionando "HB20 S", depois, clicaremos em "Próximo", verificamos que fomos à terceira página, sendo possível voltarmos às anteriores por meio da seta localizada no topo da página.

