05

Binding complexo

Transcrição

Para continuarmos, vamos utilizar o `Binding` nas páginas de detalhe e agendamento da nossa aplicação. Na primeira, definimos o título diretamente no código C#, na linha `this.Title = veiculo.Nome;` , como sendo `veiculo.Nome` .

Definiremos o título desta página no XAML e não no código C#. Então, abrindo `DetalheView.xaml` , incluiremos a propriedade `Title` do `ContentPage` , definindo-o como um `Binding` que irá trazer uma propriedade para o *code behind*. Na classe que define a página, existe a propriedade pública `Veiculo` , a qual definiremos como sendo o `Binding` do título:

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms" xmlns:x="http://schemas.microsoft.cor
```

O `Veiculo` não é um texto, e sim uma classe que representa um objeto, o veículo selecionado para *Test Drive*, não sendo propriamente um nome. Assim, passaremos no `Binding` , além de `Veiculo` , a propriedade a ser acessada para informar ao título - o nome do veículo propriamente dito.

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms" xmlns:x="http://schemas.microsoft.cor
```

Rodaremos a aplicação vendo o que acontece agora. Ao clicarmos no nome de um veículo, somos direcionados a uma tela em que nada aparece, indicando que ocorreu algum problema. Isto acontece pois definimos o `Binding` no `Title` , porém esta página ainda não sabe qual o contexto de `Binding` no qual será utilizado, ou a partir de que objeto deve pegar as propriedades para montar o `Binding` .

Assim como fizemos na página inicial, de listagem, definiremos o contexto de `Binding` , em `DetalheView.xaml.cs` :

```
InitializeComponent();
this.Veiculo = veiculo;
this.BindingContext = this;
```

Rodaremos novamente a aplicação, clicando em "Fiesta 2.0" para testarmos seu funcionamento, e agora conseguimos exibir o nome do modelo do veículo utilizando simplesmente o `Binding` . É interessante como podemos acessá-lo a partir de um "caminho": há o nome de uma instância, que possui o objeto `Veiculo` e, a partir dela, acessamos também sua propriedade.

Faremos o mesmo na página de agendamento (`AgendamentoView.xaml`), acrescentaremos `Title="{Binding Veiculo.Nome}"` na tag `ContentPage` . Modificaremos também o *code behind* (`AgendamentoView.xaml.cs`) e, em vez de passarmos `this.Title` , passaremos `BindingContext` , informando que seu contexto é ele mesmo:

```
InitializeComponent();
this.BindingContext = this;
```

Desta forma, conseguimos passar o contexto do `Binding` à nossa página. Tentamos acessar `veiculo`, mas ainda não temos uma propriedade pública com este nome nesta página. Precisamos criá-la, em `AgendamentoView.xaml.cs`, com o tipo `Veiculo` e mesmo nome.

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

    public AgendamentoView(Veiculo veiculo)
    {
        InitializeComponent();
        this.BindingContext = this;
    }
}
```

Vamos rodar a aplicação selecionando uma das opções de veículos e clicando depois no botão "Próximo". Mais uma vez chegamos a uma página em que nada é mostrado, pois quando passamos `veiculo` ao construtor (em `AgendamentoView.xaml.cs`), que vem de outra página, a propriedade `veiculo` não está sendo preenchida com este objeto.

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

    public AgendamentoView(Veiculo veiculo)
    {
        InitializeComponent();
        this.Veiculo = veiculo;
        this.BindingContext = this;
    }
}
```

Verificaremos estas alterações no emulador, indo à página de detalhe, e depois, acessaremos a página de agendamento. Agora, tudo funciona corretamente.

Nós aprendemos a utilizar o título para ajudar o usuário a não se perder durante a navegação, e vimos como fazer as navegações entre as páginas usando o objeto `Navigation`, que já vem com o Xamarin Forms. Vimos também o `NavigationPage` para encadeamento entre as páginas da app, e aprendemos como utilizar um `Binding` mais complexo, acessando-se não apenas uma instância de objeto, mas também sua propriedade.

A partir do próximo vídeo, começaremos a ver como preencher as páginas seguintes, deixando a aplicação cada vez mais completa. Até!