01

# MVVM, ListView e SelectedItem

### Transcrição

[00:00] Oi, pessoal, tudo bem com vocês? Eu sou Marcelo Oliveira. Bem-vindos de volta ao nosso curso do Xamarin com Visual Studio. Nas aulas anteriores, a gente viu como fazer um desacoplamento da nossa solução utilizando o design pattern, chamado Model View ViewModel.

[00:20] Então a gente tem aqui o diagrama, que a gente separou as partes da nossa solução, os componentes que formam a nossa aplicação. A gente separou em model, que é o M, o view, que é o V e o viewmodel, que é o VM, então, isso forma o padrão MVVM.

[00:47] A gente começou desacoplando, movendo aqui da view, a gente tirou tudo o que era modelo e a gente passou aqui para a nossa camada model e mais para frente, a gente também moveu o data binding que estava aqui no code behind. Então, a nossa view referenciava aquela classe C Sharp, que ficava atrás, que é o code behind.

[01:13] Então, a gente removeu esse data binding, que fazia no próprio code behind e a gente moveu aqui para o viewmodel. Então, aqui o data binding, a gente conseguiu fazer com sucesso acessando o viewmodel, invés do code behind. E ainda falta fazer aqui para completar o nosso padrão MVVM os comandos.

[01:37] Agora, o que são comandos? A gente vai ver agora como implementar comandos para remover principalmente os eventos que a nossa view, o nosso código xaml, ele aciona para chamar eventos que estão lá no code behind. Então, a gente vai começar a mover esses eventos de code behind...

[01:58] E passar eles para o nosso viewmodel e para isso, a gente vai utilizar a técnica dos comandos. Então, vamos lá. Então, agora a gente vai entrar aqui na solução, o nosso projeto Xamarin. A gente vai entrar aqui na primeira view que a gente criou, que é a view: listagem view.

[02:20] Nessa página, vocês se lembram que a gente acessa... quando um elemento da listagem é acionado, quando o usuário toca no item, no veículo da nossa listagem, a gente está chamando um evento aqui, que é o ItemTapped. Então, olha só, esse evento é próprio da list view.

[02:46] Então a list view tem um evento ItemTapped, isso aqui aciona um evento lá no nosso code behind. Então, eu vou abrir aqui o code behind, vamos ver. Então, aqui está evento ItemTapped do list view veículos. Então, quando o usuário toca no item da list view, a gente está acionando esse código aqui.

[03:12] Esse código é acionado para que ele possa mover o usuário para uma outra página, então o usuário navega quando ele toca no item da lista, quando ele escolhe um veículo, ele vai navegar para outra página, que é a página de detalhes para fazer a seleção dos acessórios.

[03:30] Então, o que a gente vai fazer, para continuar o desacoplamento do nosso projeto, a gente vai mover esse código que está aqui e vai passar lá para o nosso view model. Agora, por que esse desacoplamento? Por que essa preocupação? Porque com o desacoplamento, a gente reduz a dependência entre essas camadas, entre esses componentes.

[03:54] O MVVM foi criado justamente para desacoplar as coisas, para que um componente não dependa do outro. Então a gente vai começar aqui, primeiro, lá na nossa listagem view, a gente vai remover esse evento ItemTappet. Então, eu vou acertar aqui, vou remover esse ItemTapped...

[04:22] E vou, invés disso, acionar a nossa listagem, eu vou permitir que o usuário navegue para uma outra página da nossa aplicação através de um outro evento, esse evento é chamado de SelectedItem. Agora, SelectedItem não é exatamente um evento. SelectedItem é uma propriedade que está amarrada com algum objeto, com algum valor.

[04:54] E esse valor, a gente vai colocar esse valor lá no nosso viewmodel, ou seja, para a gente ligar um viewmodel com o view, a gente utiliza o quê? Como a gente já viu antes, a gente tem que usar um binding. Então, eu coloco aqui binding e eu vou colocar aqui VeiculoSelecionado.

[05:21] Agora, note que essa propriedade VeiculoSelecionado não existe ainda, eu estou criando aqui um nome qualquer, que eu vou ter que utilizar lá no nosso viewmodel. Então, agora a gente entra no viewmodel da listagem, que é o ListagemViewModel, aqui no nosso projeto.

[05:41] Então, eu estou abrindo aqui e eu vou criar uma propriedade chamada veículo selecionado. Essa propriedade, ela é do tipo... do próprio item que está sendo usado para popular essa lista, que é o veículo. Então, ela é do tipo veículo. Eu deixo essa propriedade pública, então public veiculo veiculoSelecionado.

[06:10] E aí, eu vou utilizar o assessor get e agora eu coloco aqui para ele retornar, return veiculoSelecionado, só que aí, eu vou retornar aqui um valor privado, uma propriedade privada, que é o veiculo veiculoSelecionado. Aí, eu faço o assessor set, coloco aqui um set para ele (setar) o valor do veículo selecionado, conforme o valor de entrada.

[06:53] Então agora, o que eu vou fazer? Eu vou colocar um breakpoint aqui no set, porque quando a gente selecionar o item na lista, eu quero que ele pare nessa linha. Então, agora a gente vai rodar a aplicação. Então, agora ele parou no nosso breakpoint, vamos ver o que ele jogou aqui como valor do nosso veículo selecionado.

[07:23] Ele jogou aqui um valor selecionado null, porque ele está inicializando primeiro a nossa página, então ele não tem ainda o valor selecionado. Então, eu vou rodar de novo, continua sendo null. Agora vai rodar a aplicação. Eu quero selecionar esse elemento do meio, eu quero selecionar o Fiesta 2.0.

[07:45] Agora eu vou tocar aqui. Olha só, então agora o nosso velho, ele tem aqui um veículo já selecionado. Vamos ver, vou expandir aqui, ele tem lá o Fiesta 2.0, tem o preço, tem o preço formatado, o preço total formatado e as outras propriedades desse objeto.

[08:06] Então, com isso, a gente conseguiu ver como substituir um evento do ItemTappet por uma propriedade chamada SelectedItem, que a gente pode mover lá para o nosso viewmodel. Só que ainda falta navegar, levar o usuário para outra página, como que a gente faria isso?

[08:29] A gente utilizaria o objeto do Xamarin Forms chamado navigation, mas o navigation não existe no nosso viewmodel, porque ele faz parte lá da nossa página, ele faz parte da view. Então aqui no nosso viewmodel, a gente não tem acesso ao navigation. Então, como a gente vai resolver isso?

[08:55] É o que a gente vai ver daqui a pouco.