

ViewModel, ListagemViewModel e BindingContext no xaml

Transcrição

[00:00] Então, até agora, a gente conseguiu desacoplar algumas coisas da nossa view, do nosso code behind, utilizando classes de modelo. A gente fez isso para as três views, para a de listagem de veículos, para a view de detalhe de veículo e para a view de agendamento.

[00:21] A gente moveu, tirou aqui do nosso code behind e moveu para o model. Só que a gente ainda tem algumas coisas que estão acopladas no nosso code behind. Como a gente pode ver aqui, que a view, o xaml faz uma referência ao code behind para fazer, por exemplo, o data binding.

[00:44] Então, a gente pode melhorar essa arquitetura, utilizando uma outra camada intermediária, que vai fazer essa função do code behind, para fazer esse data binding. Então, a gente vai remover esse databinding aqui, que está no code behind e vai passar ele para outra camada, para outras classes, que são os viewmodels.

[01:14] Vou colocar aqui viewmodels. Então, a.viewmodel vai ser uma intermediária entre o modelo e a camada de visualização e entre o modelo e a view. Então aqui, a gente tem a.viewmodel como intermediária entre a view e o modelo. Então, perceba que entre a view e a model...

[01:43] A gente tem aqui as notificações, o data binding e os comandos. O que a gente vai ver a partir de agora, inicialmente, só os data bindings. Então, a gente vai ver como realizar o data binding, fazendo não mais entre a view code behind, mas entre a view e a.viewmodel.

[02:05] Então, a gente vai acabar movendo o data binding que existia com o code behind para fora desse esquema. A gente vai mover o data binding lá para o.viewmodel. Então, a gente vai fazer isso agora. Então, a gente vai criar uma pasta nova dentro do nosso projeto Test Drive.

[02:26] A gente vai criar aqui, adicionar, nova pasta, vou colocar viewmodels. E aí, eu vou começar pela nossa primeira página, que é a página de listagem. Então, a página de listagem tem esse codebehind aqui e eu vou utilizar invés da listagem view do code behind para fazer o data binding, eu vou utilizar um.viewmodel.

[02:53] Então, para isso, eu vou criar uma nova classe dentro da pasta viewmodels, eu vou criar uma classe nova aqui em adicionar classe, eu vou chamar ela de: ListagemViewModel. Dentro da listagem.viewmodel, eu vou ter algumas propriedades que vão permitir que eu faça o data binding, entre a view e ela.

[03:23] Agora, vamos olhar lá na listagem view, lá no code behind, para ver o que a gente tem. A gente tem aqui a propriedade veículos, que é uma lista de veículos e a gente vai mover isso lá para o nosso.viewmodel. Então, vamos remover essa propriedade veículos. Vamos lá para o.viewmodels.

[03:47] Vou colocar aqui no.viewmodel essa nova propriedade. Vou resolver a referência aqui do veículo, resolvido. E aí, eu vou ter que pegar essa atribuição da propriedade de veículos que estava no code behind, eu vou ter que mover isso lá para o construtor do.viewmodel.

[04:07] Então, eu vou recortar essa linha e aqui eu vou criar um construtor dentro de listagem.viewmodel. Então, criando um construtor e fazendo a referência, fazendo essa atribuição do veículo. Agora, o que a gente vai fazer? A gente vai ter que modificar a listagem view.

[04:26] Para entender, a origem, a fonte do binding, a fonte dessa amarração de dados, não vai ser mais o code behind, agora ela passa a ser a nossa.viewmodel, que vai ser aquela camada intermediária. Então, para isso, o que a gente pode fazer?

[04:45] A gente pode entrar aqui no xaml e adicionar aqui diretamente no xaml, eu posso colocar um ContentPage.BindingContext e aqui na linha de baixo, eu vou colocar qual vai ser a minha nova origem de dados para o binding. Eu vou colocar aqui...

[05:07] Eu tenho local, eu tenho esse local aqui, test drive, que referenciava o meu code behind. O que eu vou ter que fazer agora é criar um novo namespace, porque isso aqui é o namespace. Então, eu tenho que criar um novo namespace, para dizer onde estão os meus viewmodels.

[05:26] Então, eu coloco aqui: xmlns, que é o prefixo do namespace e vou colocar aqui o nome de um namespace, vou colocar vm, o namespace dos view models. Então, eu vou procurar aqui o nosso código, onde está o namespace do viewmodels. Então está aqui: TestDrive.ViewModels.

[05:52] Então fica: xmlns:vm="clr-namespace:TestDrive.ViewModels". Então agora, eu vou colocar aqui como prefixo dessa teg, eu vou colocar "vm:" e ele já encontrou para mim o listagem viewmodels. Então, com isso, eu vou ter que remover também, esse binding context, que a gente tinha colocado antes...

[06:13] Que estava referenciando o próprio code behind, como origem de dados do binding. Então, removendo aqui, vamos rodar a aplicação e ver o que acontece. Estou rodando a aplicação. Vamos lá, clicando aqui, legal. Então, a nossa lista foi preenchida agora.

[06:37] Só que a origem de dados passou do code behind lá para o.viewmodel. Então, a gente conseguiu mudar o binding context, a gente conseguiu mudar o contexto dessa amarração, dessa ligação entre o xaml, entre a nossa view e os nossos dados.

[06:54] Então, olha só como ficou o nosso código agora no code behind da listagem view, então ele ficou muito mais enxuto, por quê? Porque a gente está movendo lógica de apresentação para fora do nosso code behind, com isso a gente consegue mais clareza, mais limpeza.

[07:12] A gente vai começar agora a incrementar os outros viewmodels com a lógica de apresentação das outras páginas também.