

Novo fluxo

Transcrição

[00:00] Agora vamos ter que repensar um pouco a navegação do nosso aplicativo, assim que eu faço login eu caio nessa tela, que é a tela de listagem de veículos disponíveis para agendamento do TestDrive, se eu entrar no menu eu tenho a opção nova que é “Meus Agendamentos”.

[00:20] Se eu clicar em meus agendamentos eu vou para tela de agendamentos já realizados pelos usuários, eu já agendei para três datas e 3 horários diferentes, três veículos, então eu vou fazer o “TestDrive” desses três veículos, em três datas e horários diferentes.

[00:39] Agora, se eu quisesse adicionar um quarto veículo, um quarto test drive, como eu faria? Se eu clico em uma das opções lógico que não está acontecendo nada, eu não vou para lugar nenhum, se eu voltar para o menu inicial eu não tenho um opção para voltar para listagem de veículos disponíveis para agendar o TestDrive, então eu fiquei em um beco sem saída, eu não consigo voltar para nossa listagem inicial, a listagem de veículos que foi a primeira tela que desenvolvemos nesse curso.

[01:14] Eu tenho que repensar a navegação do aplicativo, eu vou entrar aqui no desenho e aqui temos o fluxo da aplicação, primeiramente eu vou cair aqui na tela de listagem dos veículos para teste drive e depois eu posso ir navegando, eu vou para tela de detalhe, onde eu escolho aqui os acessórios, depois eu vou lá para tela de agendamento e faço o agendamento.

[01:39] Depois disso eu caio de volta aqui na tela de listagem, porém quando eu clico aqui no ícone de menu, eu vou lá para o menu inicial onde eu tenho a opção aqui de Meus agendamentos, esse botão Meus Agendamentos vai me levar para uma outra tela, eu vou iniciar um outro fluxo de navegação a partir de meus agendamentos, eu vou cair aqui nessa tela, cliquei no botão, ele vai me levar aqui para meus agendamentos, só que a partir daqui eu tenho quê?

[02:14] Eu tenho um botão menu de novo, que vai me trazer de volta para o menu, só que aqui no menu eu não tenho uma opção ainda para eu poder voltar para listagem de veículos, para eu agendar um novo teste drive, se olharmos na especificação, vamos ter um outro botão, um botão novo que ainda não implementamos e vamos ter que criar na MasterView, esse botão “Novo Agendamento” vai ter que nos levar de volta para o fluxo de navegação que começa com a ListagemView.

[02:51] O primeiro passo é criar o botão, eu vou lá na MasterView, aqui no projeto, vou parar a aplicação, MasterView.xaml, logo abaixo do botão Meus Agendamentos eu vou criar um outro botão chamado “Novo Agendamento”, para criar o botão é o componente Button e o texto vai ser “Novo Agendamento”.

[03:20] O que eu preciso fazer também é criar um comando para associar a ação desse botão com alguma tarefa que vai ser realizada, que no caso vai ser a de trocar a página de detalhe do fluxo de navegação, eu vou criar aqui um novo comando, “Command” e aqui dentro eu faço o quê?

[03:41] Eu vou associar com um comando que eu vou ter que criar lá no ViewModel, eu faço, para associar, para amarrar, eu utilizo o “binding” para amarrar e eu coloco aqui o nome do comando que eu vou chamar de “NovoAgendamentoCommand”, eu vou ter que criar ele lá na View Model e qual é a ViewModel?

[04:10] Se eu estou na MasterView eu tenho que ir para MasterViewModel, que é onde eu vou criar um novo comando, eu vou colocar aqui junto com os outros comandos, eu vou criar uma propriedade que vai ter um “get” privado, porque

eu não quero que esse comando seja alterado de fora da classe, eu coloco aqui, ele vai ser uma interface “ICommand” e eu vou colar aqui o nome do comando, “NovoAgendamentoCommand”.

[04:42] Agora eu tenho que instanciar esse comando, eu instancio esse comando junto com os outros comandos que já existem, aqui no método que eu já criei que se chama DefinirComandos, dentro de DefinirComandos eu vou aqui no final do código e vou fazer a associação, vou instanciar esse novo comando.

[05:07] Ele vai ser “new Command” e o “new Command” aceita qual parâmetro? Ele aceita uma Action, ele aceita que no caso vai ser uma expressão lambda que vai ser um método anônimo, eu faço aqui a expressão lambda e coloco aqui dentro o quê? Eu coloco o código que vai ser executado quando o usuário clicar no botão Novo Agendamento e o que eu tenho que fazer na Action desse comando?

[05:36] Se voltarmos lá para o nosso desenho, para o nosso esquema, temos aqui a MasterView, estamos dentro da ViewModel da MasterView, esse componente aqui dentro e temos que trocar o outro componente que é o detalhe, o detalhe não faz parte da MasterView, ele não faz parte da Master, o Master fica junto com a Detail dentro de uma outra página, que é a página MasterDetailView, que é uma página que está englobando os dois.

[06:12] A MasterView, assim como a Detail são filhas da MasterDetailView e ela que tem o poder de trocar a página que é atribuída como detalhe desses dois componentes, o que eu preciso fazer é comunicar a página pai, que é a MasterDetailView que eu quero trocar a página de detalhe.

[06:35] Para fazer isso utilizamos o sistema de mensagem do Xamarin Forms, utilizamos o MessagingCenter para enviar uma nova mensagem e enviamos uma mensagem com o método “send”, para enviar e aqui dentro eu passo qual é o tipo do objeto que está anexo, que está associada a essa mensagem, que no caso vai ser o “usuário” e os parâmetros que estão dentro desse método “send” são o quê?

[07:04] A Instância do usuário que é “usuário” mesmo minúsculo e aqui tipo da mensagem que vai identificar esse tipo de mensagem quando ela precisar ser interceptada, eu vou chamar essa mensagem de “NovoAgendamento”.

[07:23] Agora o que precisamos fazer é interceptar a mensagem, é assinar a mensagem, vamos assinar a mensagem no componente que tem o poder de trocar a página de detalhe, esse componente é a MasterDetailView, eu vou lá no Code Behind da MasterDetailView, onde eu tenho o código CSharp e aqui eu vou fazer a assinatura da mensagem NovoAgendamento.

[07:48] Dentro do “OnAppearing” eu crio uma nova assinatura, eu assino utilizando o componente de mensagem, que é o “MessagingCenter.Subscribe” e aqui o tipo da mensagem que é “Usuario”, aqui eu passo a instância que está assinando que é a instância “this”, aqui eu passo o tipo da mensagem que eu quero assinar que é “NovoAgendamento”.

[08:20] O terceiro e último parâmetro desse método de assinatura é uma Action que vai receber um parâmetro da mensagem e o parâmetro é no caso o “usuário”, então eu passo aqui “usuário” dentro dessa expressão lambda e aqui dentro do corpo da expressão lambda eu quero trocar efetivamente qual é a página de detalhe que vai ser exibida na tela.

[08:45] Fizemos alguma coisa similar aqui em cima, quando clicamos no botão Meus Agendamentos, eu estou trocando o “this.Detail” para uma nova NavigationPage que vai iniciar na página AgendamentosUsuarioView, vamos fazer uma coisa similar aqui, eu quero colocar “this.Detail”, eu quero trocar a página detalhe da MasterDetail, eu vou associar esse detalhe com uma nova NavigationPage, porque NavigationPage?

[09:20] Porque ele vai permitir que eu navegue para as próximas páginas, que são as páginas detalhe, a página de agendamento, todo aquele fluxo que já tínhamos desde o começo do projeto no início do curso.

[09:32] Aqui dentro eu passo a página raiz, a página inicial desse fluxo de navegação, que vai voltar a ser a `ListagemView`, eu passo aqui dentro uma nova instância de `"ListagemView"`, aqui dentro ele está exigindo um usuário, o parâmetro `"usuário"` que eu posso colocar a mesma instância que eu recebo aqui nesse método anônimo, que é o `"usuario"`, eu passo esse parâmetro aqui também para dentro da `ListagemView`, estou chamando a `ListagemView` com o parâmetro `"usuario"`.

[10:14] Agora para finalizar, eu faço o quê? Eu vou fazer com que o menu, a página Master fique invisível assim que essa página for trocada, aqui em cima colocamos `"this.IsPresented"`, vamos fazer a mesma coisa aqui embaixo, `"this.IsPresented = false"`, isso vai fazer com que esse componente, essa parte Master, do `MasterDetail` volte, ela suma da tela e nós vamos ficar vendo só a página de detalhe.

[10:55] Voltando lá, ainda temos que fazer o que? Temos que cancelar a assinatura, uma vez que assinamos, por boa prática temos que a cancelar assinatura também, vamos utilizar aqui `MessagingCenter`, para cancelar a assinatura `"Unsubscribe"`, eu estou `"desassinando"`, estou cancelando uma assinatura, aqui dentro eu passo o usuário e aqui dentro a instância que está cancelando a assinatura, que é a instância, que é a referência `"this"` e aqui o nome da mensagem, que é `"NovoAgendamento"`.

[11:31] Feito isso temos aqui o método `"OnAppearing"` assinando duas mensagens e para deixarmos o código um pouco melhor vamos extrair um método para facilitar a leitura, deixar o nosso código mais limpo, eu vou clicar com o botão direito, `"Quick Actions & Refactorings"` para refatorar, eu vou extrair o método que eu vou chamar de `"AssinarMensagens"`, apliquei, aqui embaixo `"OnDisappearing"` eu vou criar um novo método, eu vou refatorar e extrair aqui para um novo método chamado `"CancelarAssinaturas"`.

[12:17] Agora vamos fazer o quê? Vamos rodar a aplicação e ver como está o fluxo do nosso teste, eu vou entrar com o usuário `"joao@alura.com.br"`, senha `"alura123"` e vou entrar, aqui eu vou cair na página de listagem de veículos, agora eu clico no botão menu e clico em meus agendamentos, com isso vou exibir aqueles três agendamentos que já foram realizados, se eu clico neles não acontece nada, como é esperado e eu volto para o menu onde eu vou clicar agora no botão Novo Agendamento.

[12:57] Com isso, eu voltei para tela inicial que é a tela de listagem de agendamentos, conseguimos sair daquele beco sem saída, conseguimos voltar para o nosso fluxo natural da aplicação.