03

ICamera

Transcrição

[00:00] Exibir a imagem padrão do usuário na tela foi fácil, agora vamos para a parte realmente difícil que é tirar foto, capturar a imagem do usuário com a câmera do Android. Para isso vamos ter que criar um botão para o usuário clicar e vamos chamar as funções do Android que vão permitir essa captura da foto, então vamos fazer o quê?

[00:26] Vamos no código "xaml" que está na "MasterView" e aqui na página de editar o perfil do usuário vamos colocar logo abaixo da imagem do perfil um botão. E esse botão vai ficar dentro desse "StackLayout", vamos exibir ele numa sequência, vai ser a imagem e embaixo logo o botão para tirar foto.

[00:53] Vamos criar um elemento, um controle "Button" que vai ter um texto que vai ser "Tirar Foto", quando o usuário clicar nesse "Tirar Foto", nesse botão o que vai acontecer? Vamos chamar um comando no "ViewModel", fazemos isso como?

[01:16] Criamos uma propriedade "Command" e fazemos o "Binding" para uma propriedade do "ViewModel" que eu vou chamar de "TirarFotoCommand", vou copiar esse nome "TirarFotoCommand" e vamos para o "ViewModel", "MasterViewModel" e junto com as outras, junto com os outros comandos eu vou colocar aqui "public ICommand" e o "TirarFotoCommand ", vou colocar com "get" público e um "set" privado, "private set".

[02:03] Agora vamos ter que definir esse comando, vamos ter que falar para o Xamarin Forms o que fazer quando esse comando for executado. Eu vou definir a Action declarando "TirarFotoCommand" como um novo comando, "new Command" e aqui dentro passa o quê? Eu passo uma Action que vai executada quando o usuário clicar no botão para tirar foto.

[02:34] Ele vai executar esse código e agora eu coloco um breakpoint aqui no final dessa Action que está vazia, por enquanto, vamos rodar a aplicação, eu vou colocar o usuário "joao@alura.com.br", senha "alura123", vou entrar e no perfil apareceu o botão tira foto, quando eu clico em tirar foto ele executa a Action do comando "TirarFotoCommand", agora vamos começar a implementar realmente o código que vai fazer a captura da foto pelo Android.

[03:21] Agora que estamos nesse código, estamos no projeto PCL, no projeto Portable, que é o projeto portátil aqui da nossa solução test-drive. Estamos nesse projeto Portable, então vamos ter que acionar métodos para tirar foto no projeto Droid. Agora se eu for executar um código no projeto Droid eu tenho que chamar esse código a partir do projeto Portable que é o projeto que é multiplataforma, então o que eu tenho que fazer?

[04:02] Eu preciso referenciar no meu projeto Portable, esse projeto Droid, como eu faria isso? Eu teria que adicionar uma referência ao projeto "TestDrive.Droid" e vou clicar em "Ok". Apareceu um erro aqui, uma referência ao projeto "TestDrive.Droid" não pode ser adicionado, por quê? Adicionar esse projeto como referência causaria uma dependência circular. Por que que ele está dando esse erro?

[04:40] Porque aqui dentro do projeto Droid eu já tenho uma referência ao meu projeto Portable aqui em "TestDrive", se eu estou referenciando o "TestDrive(Portable)" a partir de Droid eu não posso referenciar o Droid a partir do "TestDrive". Eu não posso fazer essa dependência circular, o que eu posso fazer? Qual caminho eu tenho que seguir para poder executar o código que está nesse projeto Droid a partir do meu projeto o "TestDrive (Portable)"?

[05:17] Eu preciso arrumar uma solução para poder executar esse código que está no projeto Droid. Felizmente o Xamarin Forms fornece para gente uma solução para resolver esse problema.

[05:32] Para resolvermos esse problema utilizando o Xamarin Forms primeiro é necessário, é obrigatório que utilizemos uma interface, precisamos criar uma interface que vai conter um ou mais métodos que vamos chamar a partir do projeto TestDrive (Portable) e a partir dele vamos chamar esse método que vai ficar numa classe, que vai implementar essa interface, essa classe vai ficar no "TestDrive.Droid".

[06:06] Dentro do TestDrive (Portable) temos que criar essa interface que vai ser comum entre os dois projetos, eu vou criar uma interface que vai conter um método chamado "TirarFoto", eu tenho que definir, adicionar o novo item e adicionar uma interface. Ou melhor eu vou criar uma pasta nova, vou criar uma pasta chamada "Media" e agora eu vou adicionar uma interface chamada "ICamera".

[06:46] Criei e vou ter que definir um método que, no caso, vai ser o "TirarFoto". Vou deixá-la pública porque vai ser utilizada pelos dois projetos. Agora que temos essa interface vamos ter que criar uma classe ou fazer alguma classe, implementar essa interface no projeto Droid. Vamos para o projeto Droid e vamos procurar. Dentro do Droid temos essa classe que é a atividade principal ou MainActivity, essa classe faz o quê?

[07:26] Essa classe é a responsável por dar a partida na aplicação assim que rodamos, o executável da aplicação no caso do Android começa por aqui, essa atividade principal é que vai instanciar o nosso app, o nosso App e vai rodar o Xamarin Forms a partir daqui. O que eu vou fazer?

[07:53] Em vez de criar uma nova classe aqui no projeto Droid eu vou aproveitar o MainActivity para implementar, para fazer a implementação da interface ICamera, eu coloco aqui "ICamera", eu vou resolver o caminho, resolver a referência, e implementar o único método que é o tirar a foto.

[08:19] Implementando a interface apareceu para mim o método "TirarFoto" e agora eu vou colocar um breakpoint, que eu tenho esse método e não tem nenhuma implementação eu vou colocar um breakpoint para ver se conseguimos chegar nesse momento.

[08:38] Agora voltando ao nosso projeto "TestDrive (Portable)" precisamos fazer ele acionar o esse método "TirarFoto" que está aqui na nossa classe "MainActivity", que está no projeto Droid, como fazemos isso?

[08:54] Vamos no "MasterViewModel" e no comando "TirarFotoCommand" vamos ter que chamar aquele método que está no projeto Droid, e como fazemos isso? Vamos ter que chamar agora o método de uma classe especial do Xamarin Forms que é o serviço de dependência. Esse serviço de dependência é um contêiner, ele contém todas as dependências para essas interfaces que resolvemos em tempo de execução, utilizando o Xamarin Forms.

[09:31] O ICamera que é a nossa interface que criamos para poder tirar foto é implementada por uma classe no Droid que é o MainActivity. Aqui no outro projeto que é o TestDrive PCL, o "TestDrive (Portable)" vamos chamar um método do DependencyService, que é o serviço de dependência do Xamarin Forms para poder obter a instância que implementa o ICamera, como fazemos isso?

[10:01] Vamos ter que chamar a instância, o método, ou melhor a classe DependencyService, que vai ter um método chamado "get" e ele vai fazer o quê? Vai pegar uma instância, vai criar ou vai obter uma instância já criada dessa interface, uma instância que implementa essa interface que é a interface que tem o método "TirarFoto". É a interface ICamera, eu coloco aqui "ICamera", vou resolver a referência e eu vou fazer o quê?

[10:42] Eu vou ter que instanciar, estou criando o objeto e vou ter que chamar o método dessa interface que vai permitir eu tirar a foto. Eu vou chamar o "TirarFoto", fazendo a chamada e aqui eu deveria conseguir chamar o método para tirar foto no Android, eu vou rodar a aplicação e vamos ver se conseguimos chegar até o método tirar foto que está implementado no Android.

[11:12] Eu vou colocar o usuário “joao@alura.com.br”, a senha “alura123”, vou para o perfil do usuário e vou clicar no tirar foto para vermos se conseguimos chegar no método do Android, Tirar Foto, chegou aqui no DependencyService e vai tentar resolver essa referência para esta interface ICamera, está dando um erro aqui, está dando uma exceção de referência nula, ele não conseguiu criar uma Instância ou obter uma Instância do objeto que implementa a interface ICamera, por que isso está acontecendo?

[11:58] O problema é que estamos tentando obter uma Instância de um objeto que implementa o ICamera antes de registrar esse objeto, o DependencyService que é o serviço de dependência do Xamarin Forms exige que você, antes de obter instâncias, registre essa instância, o que precisamos fazer?

[12:24] Na mesma classe que implementa a interface que é comum aos dois projetos, que no caso é o ICamera, eu preciso definir o registro, preciso registrar aquela classe como sendo parte do serviço de dependência do Xamarin Forms.

[12:43] Eu vou lá na classe que implementa a interface ICamera que é essa interface MainActivity e eu preciso marcar, eu preciso fazer uma anotação, colocar um atributo de “Namespace” indicando que ela implementa uma interface dentro do DependencyService, porque o DependencyService vai conseguir localizar essa classe. E como que é a sintaxe para fazer esse atributo, essa anotação?

[13:17] Colocamos logo acima do “Namespace”, colocamos [assembly: Xamarin.Forms. Dependency()]. E, dentro do Dependency, temos que adicionar o tipo classe que vai ser registrada e qual é a classe que vai ser registrada?

[13:42] É a classe que tem o tipo, eu tô colocando o “typeof” para pegar o tipo dessa classe. E a classe é justamente o MainActivity, então eu coloco (typeof(MainActivity))”. Eu tenho que resolver essa referência ainda, clico com o botão direito, “Quick Actions” e eu adiciono o “using TestDrive.Droid”. Com isso ele referenciou corretamente o “Namespace” dessa classe. E agora vamos rodar a aplicação de novo para ver se consegue chegar finalmente ao método tirar foto, que está no projeto Droid.

[14:25] Agora eu vou colocar o usuário “joao@alura.com.br”, senha “alura123”, vou entrar clicando no perfil do usuário, e aqui eu vou tirar foto, clica no Tirar Foto, ele vai resolver a dependência do ICamera. Agora eu vou rodar de novo, conseguimos chegar no MainActivity que está no “Namespace”, no projeto “TestDrive.Droid”.

[15:00] Vimos como implementar utilizando DependencyService, o serviço de dependência do Xamarin Forma, vimos como fazer uma chamada de um projeto Portable para um projeto Droid, poderíamos fazer isso para qualquer outro tipo de plataforma, qualquer outro tipo de sistema operacional. Vocês aprenderam uma coisa muito importante que é fazer uma chamada de um projeto para outro, evitando uma dependência circular.