

07

ActivityResult

Transcrição

[00:00] Agora esse nosso usuário, no momento de tirar foto, se ele mudar de ideia, se ele clicar, em vez de clicar aqui no “Ok” ele clicar no cancelar, nesse botão “x”, o que acontece? Se clicarmos aqui ele caiu no “OnActivityResult”, de qualquer maneira se ele tiver aceitando ou rejeitando, ou cancelando a foto ele vai cair aqui, não podemos simplesmente pegar essa imagem e aceitar ela como imagem do perfil, por quê?

[00:33] Porque ele cancelou a foto então temos que verificar qual é o código, qual é o resultado, o código do resultado dessa atividade, fazemos isso como? Verificamos o “resultCode”, verificamos qual é o valor dele, nesse caso é “Android.App.Result.Canceled”, o “Canceled” está dizendo que o usuário cancelou essa foto, vamos fazer o quê?

[01:03] Vamos colocar uma condição, “resultCode” tem que ser “Ok”, ele tem que ser “Ok” para podermos prosseguir e enviar essa imagem lá para aplicação, para o outro projeto do TestDrive.Portable, se o resultado for “Ok”, vou colocar uma condição aqui e vamos enviar a mensagem de sucesso na hora de tirar foto.

[01:33] Aqui eu vou modificar essa mensagem também para que tenhamos um nome melhor, porque “TirarFoto” não é um nome muito apropriado, eu vou colocar aqui “FotoTirada”, que vai ser o identificador da mensagem que indica que a aplicação está pronta para receber uma foto a partir do Android, agora o que vou fazer?

[02:05] Eu preciso, agora que eu tenho a mensagem enviada aqui pela aplicação Android, eu preciso interceptar essa mensagem lá no projeto Portable, como eu faço isso? Eu vou ter que ir lá no MasterViewModel, que é onde tudo começa, onde o usuário clica no botão para tirar a foto e vou ter que criar também aqui uma nova assinatura também, para essa mensagem, como eu faço a assinatura da mensagem?

[02:35] Eu uso “MessagingCenter.Subscribe” e passo aqui qual é o tipo da mensagem que está chegando, qual é o dado, o tipo do dado da mensagem que está chegando, que nesse caso vai ser o quê? Vai ser o “Java.IO.File”, eu copio esse “Java.IO.File” para cima, mas espera, “Java.IO.File”, eu tô aqui no código CSharp, eu tô no projeto Portable.

[03:04] O problema é o seguinte, no nosso projeto Portable eu tenho uma quantidade de Assemblies limitado, eu não tenho aqui por exemplo referência para nenhum Assembly do Java, do Xamarin do Java, como eu tenho aqui no “TestDrive.Droid”, que eu tenho por exemplo, eu tenho “Mono.Android”, eu tenho “Xamarin.Droid”, eu tenho uma série de referências para códigos que são relacionados à arquitetura do Java.

[03:34] Já no meu projeto Portable eu não tenho nada disso, tenho simplesmente o mais básico, tudo que é mais básico para poder rodar em todo tipo de plataforma, o nosso projeto TestDrive (Portable) é muito básico, ele é muito simples, eu não posso de maneira nenhuma, eu nem consigo adicionar uma referência aqui para classe “File” do “Java.IO”, que eu tenho que fazer?

[04:03] Eu tenho que descobrir uma outra maneira de obter essa informação do arquivo sem utilizar uma classe que é específica, uma classe que é nativa do Java, como que eu faço isso? Podemos imaginar uma imagem, um arquivo de imagem como uma Array de bytes, a vantagem dele é que ele é um tipo primitivo, eu consigo utilizar um Array de bytes para trocar mensagens entre uma aplicação, ou melhor, um projeto “TestDrive (Portable)” e um projeto Android, vamos fazer isso, vamos tentar converter um arquivo de imagem, em uma Array de bytes.

[04:45] Eu venho aqui na classe da atividade, onde estamos recebendo o resultado da atividade, está vindo lá da câmera, eu vou trocar o tipo da mensagem, em vez de ser um “Java.IO.File” vai ser uma Array de bytes, eu coloco “byte” e Array aqui, abre e fecha chaves, para indicar que é um Array.

[05:06] Agora eu tenho um problema, porque eu estou recebendo aqui nesse método, eu tô passando aqui um “arquivoImagem” que é do tipo “Java.IO.File”, eu preciso converter esse “Java.IO.File” em um Array de bytes, eu pego esse “arquivoImagem”, eu vou remover ele daqui e vou colocar no lugar uma nova variável chamada “bytes”, que vai ser um Array de bytes.

[05:32] Eu vou declarar ela aqui em cima, “var bytes = new byte[]” e dentro dessas chaves eu tenho que colocar o tamanho desse Array de bytes e o tamanho desse Array de bytes é exatamente o tamanho do meu “arquivoImagem”, eu coloco “arquivoImagem.” e eu coloco o comprimento dele, que é a propriedade “length” que em inglês significa comprimento, faltou aqui parênteses, muito bem, agora está certinho, agora eu posso utilizar esse Array de bytes?

[06:08] Ainda não, porque esse Array de bytes é simplesmente um Array vazio, eu não coloquei nenhum dado lá dentro ainda, eu preciso preencher esse Array com valores que vão vir lá do nosso “arquivoImagem”.

[06:24] Para fazer isso eu tenho que declarar uma string, eu tenho que declarar um novo stream do arquivo de entrada, um novo “Java.IO.FileInputStream” que vai receber o quê?

[06:40] Vai receber o nosso arquivo de imagem, “arquivoImagem”, isso aqui vai retornar para mim um Stream de onde eu vou poder ler os dados do arquivo, eu declaro uma variável chamada “stream” e ele é um disposable, ele implementa a interface disposable, é bom eu colocar, uma boa prática é eu colocar um “using” porque eu vou ter certeza que depois que eu utilizar esse arquivo, esse Stream, ele vai fazer o Dispose, ele vai liberar todos os recursos que esse Stream vai consumir.

[07:18] Agora eu crio aqui o corpo do “using” que vai até aqui essa linha, agora eu tenho que declarar a variável “bytes” fora do escopo do “using”, senão eu não consigo usar o “bytes” fora no momento de enviar a mensagem lá para o outro lado da aplicação, eu declaro “byte[] bytes”, tiro esse “var”, muito bem, agora posso consumir esse “bytes” depois que eu sair do “using”, depois que o Stream for destruindo.

[07:59] Agora o que precisamos fazer é transferir para o nosso Array de Bytes o conteúdo do arquivo de imagem, fazemos isso obrigando o Stream a ler os dados do arquivo, fazemos o Stream ler os dados do arquivo “stream.Read” e vamos passar essa leitura para onde?

[08:22] Vamos passar para o nosso Array de Bytes que no momento estava vazio, eu passo aqui “bytes”, que é o nosso Array de Bytes, nesse momento vamos conseguir transferir todo o conteúdo do arquivo de imagem, que é um formato Java.IO, ou melhor, “Java.IO.File” para um “byte[]” que vai poder ser enviado lá para aplicação, ou melhor, para o nosso projeto TestDrive (Portable).

[08:51] Estamos utilizando um tipo primitivo para fazer essa troca de informações, tem um chave a mais aqui, estou tirando agora.

[08:59] Agora vamos lá para o ViewModel, onde estamos interceptando a mensagem, onde tem a assinatura da mensagem e vamos trocar aqui, em vez de “Java.IO.File” vamos utilizar um “byte[]” e aqui eu vou passar a instância “this” que é quem está assinando, é a própria classe e aqui a mensagem, o nome da mensagem, o identificador da mensagem, que agora vai ser o quê? “FotoTirada” e agora vamos ter que passar aqui, o quê?

[09:34] Um método anônimo que via ser o Callback, no momento que a mensagem é interpretada vamos receber aqui o arquivo em formato de Array de bytes e com ele vamos poder recompor, recriar a imagem original da foto, que foi tirada lá no Android.

[09:53] Eu crio aqui o Callback como um método anônimo, uma expressão lambda que vai receber o quê? Vai receber um Array de bytes que eu vou chamar “bytes” também, agora eu vou colocar um breakpoint aqui para ver se conseguimos interceptar essa mensagem corretamente e se conseguimos ler os bytes dessa imagem.

