

CameraError

Transcrição

[00:00] Agora vamos fazer um teste com a câmera do emulador de Android. No emulador tem esse ícone na tela inicial, câmera, vou clicar nele, “Camera error”, não pode se conectar com a câmera. Temos um problema da conexão com a câmera do nosso emulador. O que a gente pode fazer para resolver isso?

[00:26] Primeiro lugar temos que fechar o emulador. E agora eu vou reabrir o emulador só que mudando uma configuração, como que eu faço isso? Aqui no menu do Visual Studio eu tenho essa caixa com o robô do Android e do lado dela tem esse ícone que é um smartphone, que é o AVD. Está assim “Open Android Emulator Manager” então ele vai abrir o gerenciador de emuladores de Android.

[01:00] Eu vou abrir esse gerenciador de emulador e vou escolher qual emulador que eu estou trabalhando que é o emulador acelerado “Android_Accelerated_x86”. E vou editar, então cliço no “Edit” e modifico o quê?

[01:18] Eu tenho que modificar qual é a câmera que ele está acessando. Eu tenho uma configuração da câmera frontal, “Configures camera facing front” que está com valor nenhum e vou modificar ele para emulado. Eu tenho uma outra opção que é “webcam0” só que nessa máquina não está funcionando. Eu tenho que escolher uma opção viável, que no caso é a câmera do emulador. Também vai ser uma câmera emulada, ou seja, não vai ser a imagem de verdade.

[01:43] Aqui eu configuro também a imagem traseira da câmera, eu vou escolher a câmera traseira, valor emulado também. Para salvar essas configurações eu cliço em “Edit AVD”, ele está salvando agora para mim, ele vai salvar as informações desse simulador para eu utilizar a minha câmera do jeito que quero na aplicação.

[02:20] Agora que ele está salvando ele exibe o resultado do que conseguiu salvar e tem duas opções, “camera.back” emulado, “camera.front” emulado também. Agora temos a câmera emulada no emulador do Android.

[02:36] Agora feito isso vou iniciar o emulador, vou lançar o emulador. Ele está iniciando, agora está subindo o sistema operacional e daqui a pouco vai aparecer a tela inicial do Android. Aqui no aplicativo de câmera, quando eu cliço nessa câmera o que aparece para mim?

[02:59] Aparece uma imagem rolando, aparece esse grid preto e branco com o quadrado verde, ele fica se movimentando na tela e eu tenho também a opção de pegar a câmera traseira, essa aqui é a câmera frontal, eu cliço na câmera traseira e aparece um quadrado verde no mesmo fundo quadriculado. Isso daqui é para termos uma câmera para trabalhar nesse aparelho.

[03:29] Agora eu consigo acessar a câmera a partir da aplicação do TesteDrive. O que eu vou fazer agora? Eu vou começar a invocar, eu vou começar a chamar a atividade da câmera, porque quando abrimos a câmera temos o Activity da foto. Eu quero fazer isso, eu quero invocar essa atividade da câmera.

[03:58] Agora o que eu preciso fazer também é passar para essa atividade da câmera qual é o local do arquivo onde deve ser gravado a imagem, porque não basta só eu abrir a câmera, eu tenho que falar para câmera. Depois que o usuário clicar no botão salvar de capturar a imagem, por favor grave essa imagem nesse arquivo aqui, e vou ter que passar essas informações para atividade de câmera e como que eu passo a informação da aplicação dos Xamarin Forms, do nosso TestDrive para o aplicativo de câmera?

[04:37] Fazemos isso através do “intent” e ele não só descreve o que você quer fazer, mas também com quais dados você quer fazer isso, eu tenho o intent que está associado a uma ação de capturar imagem, só que eu tenho que passar nesse

“intent” também qual é o local do arquivo onde eu quero gravar os dados.

[05:03] Agora vamos começar a acessar a câmera do Android a partir da aplicação de TestDrive, só que antes disso vamos garantir que a nossa aplicação tenha permissões para fazer isso, como definimos permissões numa aplicação em um projeto Android, de uma aplicação Xamarin Forms?

[05:26] Vamos no projeto Droid, “TestDrive.Droid”, vamos procurar em propriedades e em “Android Manifest” temos em baixo uma lista com vários checkbox para as permissões exigidas, isso aqui é um manifesto, isso é uma lista, estamos declarando tudo que a nossa aplicação exige com permissão, por que isso? Por que existe esse tipo de coisa?

[05:59] Primeiro porque o usuário de uma aplicação Android, IOS, Windows Phone, a pessoa, o usuário, o dono do celular, do smartphone precisa saber o que aquele aplicativo está fazendo. Ele não quer, por exemplo, que todo aplicativo acesse a câmera porque isso pode resultar em um problema de segurança e privacidade. Você já imaginou se todo aplicativo que você abrisse pudesse acessar a câmera, tirar foto, gravar o áudio, o que você tá falando, gravar suas conversas? Temos o quê?

[06:44] Temos um manifesto para declarar tudo que o aplicativo está exigindo, tudo que ele precisa fazer, no nosso caso vamos dizer para o sistema operacional que a nossa aplicação exige o uso da câmera, temos que procurar aqui câmera. Permissão exigida, a câmera e o que mais?

[07:09] Ele também precisa, no final das contas, gravar a imagem que foi capturada na câmera em um arquivo, como a gente faz isso? Vamos para o final e procuramos a opção que é para escrever em um meio de armazenamento externo, “Write_External_Storage”, estamos falando, nossa aplicação também precisa gravar um arquivo em uma mídia externa, então o que acontece?

[07:42] Quando o nosso aplicativo estiver em uma loja e o usuário for instalar essa aplicação, além de estar exibindo na loja, essa aplicação exige o quê? Câmera e gravação de dados externos. Da mesma quando você vai instalar você recebe um aviso da loja de aplicativos falando que esse aplicativo exige câmera, exige gravação de arquivo em uma mídia externa, você tem certeza que você quer instalar? Quando você permite, você consegue instalar, isso deixa claro, deixa muito transparente para o usuário o que a sua aplicação vai fazer.

[08:22] Com isso temos um manifesto já declarado do que a nossa aplicação exige do sistema operacional, com esse código que não é muita coisa, que acabamos de criar vamos rodar a aplicação e ver como que isso se comporta no aparelho, ou melhor, no emulador.

[08:44] Rodando a aplicação, eu vou colocar o usuário “joao@alura.com.br”, a senha “alura123”, vou entrar agora. Clico no menu, vou no perfil e em tirar foto. Agora apareceu aquela imagem engraçada daquele Grid rolando, com esse quadrado bobo, vermelho, batendo de um lado para o outro. Isso aqui é uma emulação, uma simulação do que seria a imagem da câmera, em vez de aparecer o meu rosto está aparecendo uma imagem aleatória na tela.

[09:29] O que estamos vendo agora? Estamos vendo exatamente o aplicativo de câmera. Temos uma aplicação externa fazendo um trabalho que deveria ser da nossa aplicação de TestDrive, aproveitamos essa colaboração entre as aplicações para poder ter esse poder todo que tem o smartphone.

[09:57] Quando eu clico no botão de tirar foto ele captura a imagem, aquilo que estávamos vendo é uma imagem, como se fosse uma simulação do mundo real, estamos mexendo a câmera então a imagem está se movendo e quando eu capturo a câmera, quando eu capturo a imagem aparece a imagem estática que é a foto que eu tirei naquele momento. Eu posso tanto descartar com esse botão “x”, como também posso aceitar essa imagem com esse botão de check aqui, eu consigo aceitar, para quê?

[10:36] Para que eu possa voltar para nossa aplicação. O que acabamos de vir aqui é uma Activity, é a atividade da aplicação de câmera e essa atividade foi acionada pelo quê? Pelo nosso “intent”, o nosso intent tem essa descrição, tem o objetivo de abrir uma imagem para captura e isso que a câmera mostrou.

[11:02] Se a gente tivesse por exemplo pedido para o “intent” chamar uma ação de capturar um filme, uma gravação, a câmera abriria em outro modo, abriria no modo de gravação de vídeo, por isso que é importante você definir corretamente o seu “intent”.

[11:22] Quando eu clico agora no Check, eu quero que ele volte para aplicação, para nossa aplicação de teste drive, por quê? Porque eu quero pegar a imagem e trabalhar com ela para colocar no perfil do usuário, quando eu clico aqui no Check, ele fechou. Só que ainda falta algumas coisas para definirmos, porque precisamos definir onde gravar, porque se não informarmos para ele onde gravar não vamos conseguir obter a imagem e colocar no perfil do usuário.