

## Utilizando a câmera

### Transcrição

[00:00] Tudo bem? Antes de iniciarmos o nosso projeto, eu vou deixar o link pra download, onde já vai ter a parte visual do projeto que nós vamos utilizar.

[00:12] O objetivo desse curso é aprendermos como que nós usamos alguns recursos nativos do iOS, como, por exemplo, a câmera, o envio de SMS, ligação telefônica, o uso do mapa. Tem vários recursos nativos que nós vamos aprender.

[00:28] Nós vamos trabalhar em um projeto da agenda. É muito trabalhoso para qualquer pessoa ficar anotando, por exemplo, endereço, informações em um papel. Então nós vamos fazer um app para automatizar essas coisas. No aplicativo que nós vamos fazer, vamos conseguir cadastrar alunos, vamos colocar a localização dele, vamos colocar o número de telefone e tudo isso nós vamos utilizar pra montar essas features.

[00:54] A primeira delas que nós vamos ter que utilizar é nessa tela de cadastro. Nós temos aqui um formulário, onde temos uma foto, uma image view, tem aqui algumas labels, text fields, tem nome, endereço, telefone, site. Tem o stepper, onde nós conseguimos colocar a nota e o botão salvar.

[01:16] Eu estou com o aplicativo rodando, só pra nós entendermos qual vai ser nossa primeira tarefa. Eu vou clicar no “+”, onde vai abrir o formulário e aqui tem vários campos, onde nós vamos preencher normalmente. Só que repare que aqui nós temos a foto, que é um botão. Eu clico aqui e não está acontecendo nada. Nós vamos iniciar o projeto a partir desse ponto.

[01:37] Qual é o arquivo dele? Vamos entender. Aqui é o Aluno View Controller, eu estou com a pasta “Alunos” aberta e vou abrir esse view controller. Vamos descer até nós acharmos a action, que é o botão foto. Aqui nós vamos ter que implementar a utilização da câmera, então é isso que nós vamos ver agora.

[01:58] Pra utilizar a câmera, nós vamos utilizar uma classe que se chama UIImagePickerController Controller. Então vamos criar uma constante dessa classe. “let”, nós podemos colocar, por exemplo, “multimidia”, que ela serve tanto para acessar a câmera quanto para buscarmos fotos na biblioteca do iOS, que também nós vamos ver. E vou inicializá-la. “UIImagePickerController”, é esse cara que nós vamos utilizar. Vou instanciar.

[02:29] Feito isso nós precisamos falar pro iOS qual é o tipo de source, qual é o tipo de dado que nós vamos querer. Você vai querer exibir a câmera ou você vai querer ver a biblioteca de fotos do usuário? Nós temos que configurar isso.

[02:44] Vou colocar aqui “multimidia.sourceType” e aqui eu vou colocar um ponto e ele tem um menu de opções. Qual é a opção que nós vamos querer agora? A câmera. Então “camera”. Nós já criamos então o controller, que vai servir pra exibir a tela de fotos no nosso app. Como que nós fazemos pra exibir?

[03:05] Nós vamos aqui, “self.present” e ele espera um “UIViewController”. Como o UIImagePickerController herda de UIViewController, nós podemos passá-lo aqui, então “multimidia”, animado eu vou pôr “true”, que eu vou querer que ele apareça animado, e unblocked completion aqui eu não vou querer fazer nada. Nós vamos testar isso agora. Vou rodar o app e nós vamos ver o que vai acontecer.

[03:35] Vou clicar no “+”, vim pro formulário. Eu vou clicar aqui em “foto”, vamos ver o que vai acontecer. Cliquei aqui. No console já deu que ocorreu alguma exception. Vamos entender o que aconteceu?

[03:50] Vou subir um pouco, deixa eu aumentar pra podermos visualizar. Ele fala que terminou o app porque o source, que é a câmera, não está disponível. Por que será que não está disponível?

[04:04] É o seguinte, alguns recursos do iOS, como a câmera, o GPS, o envio de SMS, nós não conseguimos testar no simulador, então nós precisamos realmente de um iPhone ou um iPad de verdade pra testar esses recursos, por isso que ele deu esse exception pra nós.

[04:24] Eu estou aqui com meu iPhone já plugado no Mac, então vamos testar nele. Mas o ideal é nós fazermos uma verificação, para saber se está disponível esse recurso no device que o Xcode que vai instalar o app ou não.

[04:38] Como que nós fazemos isso? Através de um if. Vou colocar aqui um “if UIImagePickerController.isSourceType”, eu passo aqui a câmera, porque eu quero saber se é a câmera que está disponível ou não. Se ela entrar no if significa que a câmera está disponível pra uso, então eu vou passar todo esse código pra dentro do if. Agora sim eu posso testar em um device de verdade.

[05:08] Será que nós precisamos fazer mais alguma implementação antes disso? Nós já vamos ver. Vou clicar aqui e nós vamos rodar em um iPhone agora. Vou vir aqui no “+” e agora nós vamos clicar no botão de foto. Cliquei em foto e olha só, ele deu outro exception.

[05:28] O que acontece é o seguinte, quando nós acessamos algum recurso do usuário, como o GPS, a câmera e alguns outros, nós precisamos de permissão pra acessar esses recursos, eu não posso simplesmente usar o GPS no meu app e conseguir localizar onde o usuário está sem que ele saiba, com a câmera é a mesma coisa, eu não posso acessar a câmera sem que o usuário permita. Então nós precisamos colocar uma permissão, falar pro usuário: “eu preciso acessar a câmera pra utilizar, pra fazer tal coisa”, então é isso que nós vamos fazer agora.

[05:58] Ele está falando que nós precisamos configurar esse arquivo “Info.plist”, está vendo que ele deu essa exception e falou qual arquivo que é? É isso que nós vamos fazer agora.

[06:08] Vou vir aqui e esse arquivo está aqui, “Info.plist”, é ele que nós vamos utilizar. Esse arquivo guarda várias configurações gerais do nosso app, então nós precisamos colocar a nossa permissão aqui. Como nós fazemos?

[06:23] Quando eu coloco o ponteiro do mouse em cima dessa primeira coluna, ele aparece um “+” aqui, está vendo? Então aqui nós conseguimos incluir uma nova permissão, é isso que eu vou fazer. Eu vou clicar aqui, beleza. Aí ele vai dar uma série de opções.

[06:37] Pra nós acessarmos a câmera, eu vou digitar “Privacy” e aqui ele tem vários recursos que nós precisamos pedir permissão, Bluetooth, calendário, câmera, que é o que nós precisamos, contatos, face ID, o app de saúde. Então tem várias features que nós precisamos pedir permissão para o usuário, para que nós possamos acessar. O que eu vou escolher é o da câmera.

[07:05] Fiz isso, agora eu vou dar um enter, só pra você ver como que fica. É essa permissão que eu preciso. Aqui tem uma coluna onde está escrito “Value”, o que nós colocamos aqui? Nós podemos colocar qualquer texto, na verdade nós precisamos explicar pro usuário por que precisamos utilizar esse recurso do device dele.

[07:30] Eu posso colocar, por exemplo, “Este aplicativo precisa utilizar a câmera para preencher o formulário” e vou dar enter. Aqui pode ser qualquer texto, o que você quiser pôr pra explicar pro usuário por que você vai utilizar. Agora sim nós podemos testar a implementação da câmera. Eu vou rodar novamente no meu iPhone e nós vamos testar.

[07:58] Então vamos clicar agora no “+” e em seguida em “foto”. Vou clicar em “foto” e antes de acessar a câmera olha o alerta que ele está mandando, ele fala “Agenda”, que é o nome do app, precisa acessar a câmera, “esse aplicativo precisa utilizar a câmera para preencher o formulário”. O usuário pode escolher em deixar ou não nós acessarmos esse recurso do dispositivo dele. Eu vou clicar em “Ok”.

[08:28] Agora eu já estou conseguindo utilizar a câmera, olha só, a câmera já está funcionando. Eu vou tirar uma foto. Aqui nós temos a opção de usar foto, que é o que nós precisamos. Quando o usuário clicar em usar foto, tem que aparecer no formulário. Então vou clicar em usar e não aconteceu nada, porque nós ainda não implementamos nada nesse callback, ou seja, quando o usuário clicar em tirar a foto, nós temos que utilizar essa foto, então vamos fazer isso agora.

[09:00] Primeiro passo, nós voltarmos nesse “AlunoViewController” e o que nós precisamos fazer? Implementar os métodos de delegate dessa classe aqui, “UIImagePickerController”, ela tem os métodos de delegate, inclusive que é esse quando o usuário seleciona a foto, que é o que nós vamos utilizar agora.

[09:20] Pra não deixar tudo no ViewController, todas as responsabilidades acopladas no ViewController, eu vou criar uma nova classe pra implementar esses métodos, pra nós deixarmos isso isolado.

[09:31] O que eu vou fazer agora? Eu vou criar um novo grupo, vou chamar de “Componentes”, que depois nós vamos pôr mais arquivos aqui dentro. O puxei pra baixo. Eu vou criar um arquivo que vai se chamar “ImagePicker”. Vou dar um next e vou dar um create.

[09:56] Pra nós utilizarmos os métodos de delegate, qual o primeiro passo? Nós importarmos o protocolo desses métodos. Então eu vou pôr uma vírgula, “UIImagePickerControllerDelegate”. Selecionei esse cara. Quando eu faço isso eu já tenho acesso aos métodos. O método que nós precisamos é o seguinte, “didFinishPickingMediaWithInfo”, esse método grandão aqui.

[10:22] Repare que na assinatura do método, ele tem um dictionary, um dicionário onde aqui é uma string e o valor é N, nós ainda não sabemos qual é. Nós precisamos extrair a foto desse dicionário. Então “let foto”. E eu vou pegar esse info, que é o dicionário, e eu vou pegar essa propriedade, “UIImagePickerControllerOriginalImage”, e vou convertê-la pra “UIImage”, que é o que nós precisamos pra setar no nosso outlet.

[10:58] Depois que eu conseguir pegar a foto, eu vou fechar a tela de foto, porque o usuário já tirou a foto, já clicou em use, não tem porquê deixá-la. Eu vou pegar esse “picker”, que está na assinatura do método, “.dismiss” e vou deixá-lo animado. Então aqui vão ficar os nossos métodos, vou criar o marcador para deixar os métodos, e vou criar outro marcador pra nós deixarmos os atributos que nós formos criar. Então “Atributos”, só para deixar mais organizado.

[11:35] Só que como nós fazemos para falar pro nosso image picker, que é essa constante que nós criamos, que quem vai implementar os métodos dessa classe vai ser esse arquivo que nós criamos, como é que nós fazemos isso? Nós precisamos criar uma constante desse arquivo.

[11:54] Vou vir aqui em cima, vou criar um novo marcador também, “Atributos”, e vou criá-lo. Então “let imagePicker” é igual à classe que nós criamos, qual é a classe? “ImagePicker”. Instanciei-a.

[12:23] O que nós precisamos fazer? Como nós temos acesso à constante que nós criamos, que é esse “multimedia”, nós podemos fazer o seguinte, “multimedia.delegate”, ou seja, quem vai implementar esses métodos não sou eu, vai ser essa constante que nós criamos. Qual é a constante? Esse image picker, então eu vou copiá-lo e vou colar aqui.

[12:49] Ele vai pedir para nós implementarmos outro protocolo, que é esse “UINavigationControllerDelegate”, então pra ele fazer isso eu preciso utilizar esse protocolo aqui. Esse nós já implementamos, “ImagePickerControllerDelegate”, falta esse “NavigationController”. Então vamos voltar no arquivo, colocar aqui uma vírgula, “UINavigationControllerDelegate”. Já fizemos o que precisava para ele entender que quem vai implementar os métodos vai ser esse arquivo que nós criamos.

[13:23] Vamos ver se está funcionando? Eu vou deixar, por exemplo, um breakpoint, e vou rodar o aplicativo novamente pra nós testarmos. Vamos lá. Estou com meu iPhone aqui, vou clicar no “+” e agora eu vou clicar na foto. Selecionei,

tirei uma foto e vou clicar em “Use”. Cliquei e caiu exatamente no breakpoint que eu havia colocado. Então está funcionando, nós conseguimos fazer com que o nosso UIImagePickerController entendesse que quem vai implementar os métodos de delegate dele é esse arquivo que nós criamos.

[14:05] Só que nós temos aqui a foto e nós não estamos fazendo nada com ela por enquanto. Se você reparar, nós só estamos extraíndo a foto. Será que é só isso que nós precisamos? Não é. Com a foto, nós precisamos setar no nosso outlet, que é esse outlet que tem aqui em cima, “imageAluno”, é ele que nós vamos utilizar.

[14:25] Como que faz pra nós mandarmos essa foto pra esse arquivo novamente? Uma boa saída seria nós utilizarmos um protocolo. Vamos fazer isso? Vou vir aqui, “protocol”, qual vai ser o nome dele? “UIImagePickerControllerFotoSelecionada”, esse é o nome do meu protocolo. E aqui eu vou criar um método, “func UIImagePickerControllerFotoSelecionada”. Qual é o parâmetro que eu vou mandar? A “foto”. Qual é o tipo dela? É uma “UIImage”.

[15:01] Feito isso nós temos um protocolo em que nós vamos disparar, vamos implementar nesse arquivo e nós vamos conseguir passar foto pra lá. Nós precisamos criar aqui um delegate, do tipo do protocolo que nós criamos. Vou colá-lo aqui e vou deixar optional, porque nós não sabemos se vai ter ou não valor esse cara, por isso nós vamos ter que tratar.

[15:27] Quando que nós vamos chamar esse método que criamos? Quando nós vamos passar essa foto? Quando ele clicar em usar a foto, então nós podemos fazer o seguinte, “delegate.” o nome do método, “UIImagePickerControllerFotoSelecionada”, e nós passamos a foto, que é essa constante que nós extraímos do dicionário que veio na assinatura desse método.

[15:51] Só que pra tudo isso funcionar, nós precisamos implementá-lo aqui, nesse AlunoViewController. O primeiro passo é nós criarmos um novo método onde vamos deixar essas configurações de delegate, então vou criar o método “setup”. Está vendo que nós criamos aqui a constante, que é do tipo da classe que nós criamos o protocolo? Então eu vou pegá-lo agora, “UIImagePickerController.delegate = selfie”.

[16:20] Quando nós fazemos isso, nós precisamos agora utilizar o protocolo, que é esse protocolo aqui, “UIImagePickerControllerFotoSelecionada”. Então vamos vir aqui em cima, colocar uma vírgula, “UIImagePickerControllerFotoSelecionada”.

[16:36] Agora que nós já importamos o protocolo UIImagePickerControllerFotoSelecionada aqui no ViewController, nós podemos utilizar o método que nós criamos lá. Aqui embaixo eu vou criar um novo marcador, “MARK:” e vou chamá-lo de “Delegate”, que vai ser onde vão ficar todos os métodos delegate que nós formos utilizar. Então vou chamar “UIImagePickerControllerFotoSelecionada”, agora pego o outlet. Qual é o nome do outlet? “imageAluno.image = foto”, que vem aqui por parâmetro.

[17:08] Feito isso nós já estamos setando a foto que está vindo na assinatura do método no outlet que nós temos aqui no ViewController. Pra funcionar nós só precisamos chamar agora esse método setup, então vou chamá-lo aqui em cima, “setup”. Agora sim nós vamos testar a nossa implementação.

[17:26] Com o app rodando no meu iPhone eu vou clicar no “+” e agora vou clicar em “foto”. Vou tirar uma foto e vou clicar em usar. Cliquei em usar e agora ele está setando a foto que nós tiramos no nosso formulário. Com isso nós conseguimos utilizar a câmera pra fazer várias outras coisas. Nesse caso nós estamos utilizando pra um formulário.

[17:53] Recapitulando, primeiro nós precisamos fazer uma validação pra verificar se a câmera está disponível ou não. Se for simulador vai dar crash. Então nós precisamos fazer essa validação. Depois nós precisamos vir nesse arquivo, “Info.plist” e pedir autorização do usuário pra conseguirmos utilizar esse recurso do dispositivo dele.

[18:14] Feito isso é só nós utilizarmos os métodos de delegate do UIImagePickerController e setar a imagem que ele devolve no lugar que nós precisarmos utilizar. Na próxima aula nós vamos aprender a utilizar a biblioteca do iOS. Até daqui a pouco.

