

Utilizando a biblioteca de fotos

Transcrição

[00:00] Agora que acabamos de implementar a funcionalidade da câmera no nosso app, é hora de aprendermos como que nós utilizamos a biblioteca de imagens do iOS, porque, pensando bem, se eu vir aqui no “+” e clicar em “foto” e tiver com o app buildado em um device de verdade, ele vai abrir a câmera.

[00:18] Mas não é toda hora que nós queremos tirar foto naquele momento e usar em algum aplicativo, geralmente nós já temos alguma foto na nossa biblioteca que nós gostaríamos de utilizar. Então o melhor caminho é deixarmos essas duas opções, o usuário ter a possibilidade de tirar a foto na hora e também a possibilidade de ele conseguir acessar a biblioteca de fotos do iOS. É isso que nós vamos ver agora.

[00:43] O primeiro passo é criarmos um menu para aparecer e mostrar essas opções para o usuário. Então quando ele clicar no botão “foto” não vai abrir a câmera direto, vai aparecer um menu e vai ter a opção tirar foto ou acessar a biblioteca. Para fazer isso tem um componente que nós usamos no iOS que se chama action sheet, é ele que nós vamos implementar.

[01:06] Abrindo a pasta “Componentes” nós vamos no arquivo “ImagePicker”, e vamos criar um método pra montar esse menu. Deixa eu dar alguns espaços pra ficar com a visualização melhor. Vou criar um método pra montar o menu, então “func menuDeOpcoes()” ele vai retornar um UI Alert Controller, que é o componente que nós vamos usar pra mostrar o menu. Então vou colocar aqui o retorno “UIAlertController”, beleza.

[01:35] Agora nós precisamos criar um Alert Controller, então vou declarar aqui uma constante, que vai se chamar “menu” e ela vai ser igual a “UIAlertController”. No método construtor dele eu tenho que passar algumas informações, como título, então aqui eu posso pôr, por exemplo, “Atenção”. A mensagem eu vou colocar “escolha uma das opções abaixo” e o estilo dele eu vou selecionar o estilo “actionSheet”, que é o que nós vamos utilizar.

[02:09] O action sheet é o que tem mais opções, o alert é um alerta normal, só tem opção de cancelar e ok, então o que nós vamos utilizar é o action sheet. Como o menu já está criado, eu vou dar um “return” e vou retornar já o menu nessa função, pra ele não ficar apontando nenhum erro no nosso código.

[02:30] O que nós precisamos fazer? Agora nós já criamos o menu, precisamos adicionar quais opções o menu vai ter. Então ele vai ter a opção de tirar foto e a opção de acessar a biblioteca. Pra criar essas opções, como que eu faço? Vou criar outra constante, a primeira vai ser a câmera, então “camera”. A câmera vai ser um “UIAlertAction”, que é um botão que vai ter no nosso menu, então eu vou inicializá-la passando o título, vou pôr, por exemplo, “tirar foto”. O estilo dela vai ser um estilo “default”.

[03:05] E aqui repara que é uma closure, então se eu apertar o “enter” eu posso dar um nome, eu vou pôr, por exemplo, “acao”, e aqui dentro é o que nós vamos fazer quando o usuário clicar na opção câmera no nosso menu, então aqui nós ainda vamos implementar, vou deixar aqui um comentário.

[03:24] Depois que nós criamos a opção, nós precisamos adicionar essa opção no menu, como nós fazemos isso? Vamos em “menu.addAction” e nós passamos a opção que nós criamos, que no caso foi a câmera, foi essa opção que nós criamos.

[03:42] Além da câmera, agora nós temos também a biblioteca, que é o que nós precisamos. Então eu vou criar outro Alert Action e vou instanciá-lo. O nome dele vai ser “biblioteca”, vai ser a opção biblioteca, o estilo dele também vai ser o default e aqui também tem uma closure que nós vamos implementar alguma ação, que vamos ver logo. Então

“implementar”, deixa aqui para nós não esquecermos. Eu vou pegar o menu e vou dar um “addAction” e aqui nós passamos a biblioteca.

[04:20] Feito isso, o menu já está ok. Nós criamos já o action sheet, agora nós precisamos tratar os eventos de quando o usuário escolhe uma das opções, se ele escolher câmera, eu preciso de alguma forma falar pro View Controller que o usuário quer exibir a câmera. Se eu escolher a biblioteca, eu tenho que falar pra esse View Controller que o usuário quer acessar a biblioteca.

[04:44] Pra fazer isso nós podemos utilizar uma closure na assinatura do nosso método, então nós passamos a opção que o usuário escolheu e no View Controller nós tratamos qual foi a opção. Então nós vamos fazer isso.

[04:56] Pra ficar ainda mais organizado, nós vamos criar um “enum” aqui em cima pra colocar essas opções. Vou digitar “enum” e aqui nós vamos dar um nome, eu vou pôr por exemplo, “MenuOpcoes”. Qual vai ser a primeira? Vai ser a “camera”, e depois nós temos a “biblioteca”, então vão ser essas duas opções possíveis aqui.

[05:24] Agora na assinatura do método nós podemos fazer um bloco de completion, ou seja, quando nós implementarmos essa função eu vou conseguir saber qual é o botão que o usuário vai clicar. Então aqui eu vou pôr um “completion”, nós vamos passar qual foi a opção que o usuário vai escolher, vou colocar aqui “opcao:”.

[05:47] A opção vai ser do tipo do enum que nós criamos aqui, então eu vou copiar ele, vou descer no nosso método e vou colá-lo aqui, então essa variável opção vai ser algum desses dois cases, ou vai ser a câmera ou vai ser a biblioteca. Então vamos tratar isso lá. Aqui eu tenho que colocar um “Void” na nossa closure.

[06:11] Feito isso, agora nós precisamos falar qual vai ser a opção que nós queremos que acione quando o usuário clicar em câmera. Então vou pegar aqui o “completion”, menu de opções, então se ele clicar em câmera eu vou mandar a opção câmera, que é o que faz sentido, e se ele clicar em biblioteca eu vou chamar o completion e vou passar a opção da biblioteca.

[06:41] Além dessas duas opções, é legal também nós criarmos um botão para o usuário conseguir cancelar, porque imagine só que ele clicou sem querer no botão de tirar foto e vai aparecer o menu, então é interessante termos essa opção de cancelar. Então nós vamos criá-la aqui, “cancelar”, vai ser uma Alert Action também, vamos inicializá-la aqui, e damos o nome de “cancelar” mesmo, e o estilo não vai ser mais o default, o estilo que nós vamos pegar vai ser o “cancel”.

[07:14] E nessa closure nós não vamos querer fazer nada, quando o usuário clicar em cancelar nós só queremos que o menu desapareça e isso ele já faz por padrão. Então aqui nós vamos passar o “nil”. Agora pegamos o menu, damos um “addAction” e passamos o botão cancelar que nós criamos.

[07:31] Nós terminamos a implementação do nosso menu, agora nós precisamos utilizá-lo no View Controller, então vamos voltar pra lá? Vou clicar no “AlunoViewController”. Aqui é a implementação que nós fizemos da câmera. Eu vou criar então o nosso menu, “let menu = UIImagePickerController”, que é o cara que nós criamos, “.menuDeOpcoes”.

[07:59] Como é uma closure que nós fizemos o de completion, nós podemos colocar, por exemplo, “opcao”, agora nós conseguimos tratar qual foi a opção que o usuário escolheu. E aqui nós vamos apresentar para o usuário o nosso menu, então vou dar um “present” e aqui eu passo o menu. Menu, animado verdadeiro, e aqui eu passo o “nil”.

[08:26] Agora eu vou criar um método pra nós tratarmos se é a biblioteca ou a câmera que nós vamos ter que exibir. Vou criar aqui outro método, “mostrarMultimedia()”, o que eu vou receber como parâmetro? Eu vou receber uma opção. Então vou colocar aqui “opcao”, que ela é do tipo do enum que nós criamos no Image Picker.

[08:50] E aqui o que eu faço? Eu chamo esse cara. Como eu estou dentro de uma closure eu preciso pôr o “self” e dou um “mostrarMultimedia”, e eu passo a opção que o usuário escolheu. Para não ficar opção, opção, nós podemos ocultar esse

primeiro parâmetro, então se eu colocar um “_” e der um espaço, eu não preciso mais passar de novo, duas vezes opção. Fica mais legível dessa forma.

[09:19] Agora essa implementação da câmera, eu vou pegá-la, vou recortar daqui e vou colá-la dentro desse método, que é aqui que nós vamos tratar. O que nós precisamos? Nós precisamos do Image Picker Controller, então vou tirá-lo daqui, passar pra cá. Nós precisamos setar o seu delegate, independente da opção que estiver. O source nós precisamos tratar, se for câmera nós exibimos a câmera, então eu vou fazer aqui um “if”, a opção que nós recebermos por parâmetro, então vou pegar aqui “if”.

[09:59] Só que não basta ela ser só opção câmera, ela tem que ser opção câmera e tem que estar disponível para o iOS exibir isso, então eu tenho que colocar aqui um “&”. Se for a opção câmera e estiver disponível, nós mudamos o source do multimídia aqui pra câmera, se não, nós exibimos a biblioteca. Deixa-me tirar esse “present” daqui, colar aqui e tirar esse “if” pra fora. Se não for a câmera, eu exibo a biblioteca do iOS.

[10:31] Pra exibir a biblioteca tem uma opção no próprio enum pra isso, que é essa “photoLibrary”, então nós vamos pegar esse cara aqui, e no final nós exibimos qual foi a multimídia que o usuário escolheu. Feito isso, agora nós já podemos testar. Vamos rodar o aplicativo e ver como ficou a nossa implementação.

[10:53] Eu vou dar um rodar aqui. Vou clicar no “+”. Como eu estou rodando o app no simulador, eu não vou ter acesso à câmera, mas a biblioteca nós conseguimos ver. Cliquei no botão “foto”, repara que apareceu um menu com as opções. Então “tirar foto”, eu consigo ver se eu estiver no meu iPhone, a biblioteca eu consigo acessar, ele vai mostrar para mim a biblioteca do iOS, inclusive eu posso pegar alguma imagem padrão. Vou clicar aqui e nós já conseguimos setar a imagem no nosso formulário.

[11:24] Se eu clicar de novo aqui, repara que tem a opção “cancelar”, que nós também criamos. Clico em “cancelar” e ele dispensa o menu.

[11:33] Essa parte do formulário nós já matamos. Nós conseguimos tirar uma foto, conseguimos acessar a biblioteca do iOS. Agora nós precisamos preencher o formulário, colocar uma nota pro aluno e salvar esse aluno na nossa agenda. Como é que salva esse aluno na nossa agenda? Nós vamos ver daqui a pouco.