

Criando o menu opções

Transcrição

[00:00] Continuando o projeto, agora vamos implementar novas features nativas do iOS. A primeira que nós vamos trabalhar é o envio de SMS a partir do nosso aplicativo. Aquela telinha de SMS nativa que nós temos no iPhone e no iPad, nós vamos utilizá-la agora no nosso app.

[00:17] Mas vamos pensar, como nós vamos abrir esse componente nativo de mensagens do iOS? Um bom caminho seria o que nós já utilizamos, lembra quando nós criamos aqui o formulário no botão de foto? Nós utilizamos um action sheet pra exibir essas informações. Nós poderíamos também utilizá-lo de alguma forma aqui na table.

[00:40] Mas como nós podemos chamar esse menu de opções a partir da table view? Porque se nós clicamos nele, ele leva pra tela de formulário, onde podemos editar um aluno. Tem um recurso interessante no iOS que se chama long press, que nada mais é do que um clique que nós damos na célula, longo. Então se eu segurar por uns três, quatro segundos, ele dispara uma ação desde que eu crie esse gesture, gesture recognizer, na verdade. Nós vamos utilizá-lo aqui no nosso aplicativo. Vamos lá.

[01:12] Para criarmos esse long press, precisamos ir no método onde nós criamos a célula da tabela, que é o método Cell For Row. Aqui embaixo de onde nós criamos a célula, eu vou criar esse long press, então “let longPress”, ele é do tipo “UILongPressGestureRecognizer”. No método construtor dele nós temos que passar o target e a action que queremos disparar quando esse long press ocorrer.

[01:45] Aqui no “target” eu vou passar “self” e na “action”, como já sabemos, é de costume, nós colocamos o selector, que é o “#selector”, e aqui dentro nós colocamos o método que vamos criar agora, ou seja, quando eu fizer o long press vai disparar um método, que é esse que nós vamos criar agora.

[02:02] Então aqui no métodos, vamos vir aqui por último e vamos criar mais um, “func abrirActionSheet” e eu vou passar aqui por parâmetro, então vai ter o “longPress”, que é do tipo “UILongPressGestureRecognizer”. Aqui já podemos utilizar o método, então o método vai ser esse, “abrirActionSheet”.

[02:33] Como estamos usando o selector, nós temos que usar aquela anotação, que é esse “@objc”, essa anotação. Agora ele parou até de dar o erro que estava dando.

[02:46] Depois que nós criamos o long press, nós precisamos adicionar esse evento na célula que nós criamos, então como nós fazemos isso? Vou vir aqui no final, “célula.addGestureRecognizer”, esse método aqui. Esse cara que nós vamos utilizar. Aqui eu passo o long press que nós criamos.

[03:11] Pra ver se está funcionando mesmo, podemos fazer uma verificação. Vamos colocar aqui um “print”, aqui nós colocamos alguma coisa, por exemplo, “LONG PRESS AÇIONADO”. Vamos rodar o app, para ver se esse long press está funcionando? Vou gerar um build aqui.

[03:34] Com o simulador aberto, eu vou até apagar já o console.log que tem aqui, então vou dar um excluir e agora eu vou fazer o long press. Cliquei e segurei um pouco. Ele apareceu duas vezes, uma quando eu cliquei e segurei, e outra quando eu soltei o clique. Não é exatamente o que nós queremos, nós queremos pegar só o começo, quando o usuário clicar na célula, depois de alguns segundos é disparado esse método. Quando ele soltar, nós não queremos fazer mais nada.

[04:03] Pra configurar esse evento, nós vamos aqui, apagamos esse print e podemos fazer o seguinte, “if longPress”, que é o que recebemos por parâmetro, nós conseguimos ver o estado dele. Se ele for, por exemplo, esse “.began”, que é o começo dele, sim nós fazemos alguma coisa, que é exibir o menu que nós vamos criar. Então legal, está funcionando o long press que nós criamos.

[04:26] O próximo passo é criarmos esse menu de opções, parecido com o que nós fizemos mesmo lá no foto. Aqui na pasta “Componentes” nós vamos criar uma nova classe pra deixar isso mais organizado, não deixar tudo só no View Controller, porque imagine se a aplicação cresce, nós queremos tirar alguma coisa, está tudo dentro do View Controller, então fica muito poluído.

[04:49] Isolando esse menu e algumas outras coisas que vamos ver, fica mais limpo o nosso códigos e se precisarmos tirar ou colocar mais coisas, também fica mais fácil. Por isso nós vamos criar uma classe que vai montar esse menu pra nós. Eu vou pôr aqui “MenuOpcoesAlunos”, essa vai ser a classe. Vou dar um create. Então nós podemos começar aqui criando o nosso menu.

[05:21] Vou vir aqui, “func configuraMenuDeOpcoesDoAluno” e ele vai retornar, um que nós na verdade já vimos, que é esse “UIAlertController”, nós já trabalhamos com ele. Então vamos criar esse UIAlertController. “let menu = UIAlertController”, vou inicializá-lo passando o título, o título pode ser atenção mesmo, “Atencao” e a mensagem vai ser “escolha uma das opcoes abaixo”. O tipo dele vai ser esse “actionSheet” mesmo e nós já podemos dar um return, para ele parar de apontar erro no nosso código, então vamos dar um “return menu”.

[06:16] O menu nós já temos, agora nós precisamos criar as opções para pôr dentro do menu, então vamos criar. “let”, qual vai ser a primeira opção? Vai ser o envio de SMS, então vamos criar o “sms”, que é do tipo “UIAlertAction”. Nós vamos inicializá-la passando o título, o título pode ser “enviar SMS”. O tipo dela vai ser o “default” e aqui vai ser o que nós vamos fazer quando o usuário clicar em cima dessa opção do SMS, então nós vamos configurar.

[06:47] Como é uma closure aqui, vou apertar o enter e eu vou dar um nome pra ele, pode ser “acao”, então aqui nós vamos implementar já. Vou deixar aqui comentado, “implementar”.

[06:58] Depois que nós criamos a action, ação do nosso menu, nós precisamos adicioná-la. Então vamos chamar o “menu.addAction” e vamos passar aqui o “sms”. O menu está ok.

[07:11] Agora nós precisamos configurar a ação do botão SMS. Vamos criar um menu, para deixar essas opções mais organizadas. Vou criar aqui o menu, nós podemos chamá-lo de “MenuOpcoesAlunos”, cuidado para não confundir com aquele outro nome que nós já criamos, que é o menu de opções. Aqui o primeiro case vai ser o sms.

[07:45] Pra configurarmos isso, pra não termos que ficar colocando protocolo no nosso View Controller nem nada, nós vamos utilizar uma closure também. Aqui ele está dando erro, porque o nome do enum está igual o nome da classe, então nós vamos ter que mudar o seguinte, vamos colocar aqui “MenuActionSheetAluno”.

[08:05] Então nós podemos criar uma closure, que vai nos devolver a opção que o usuário escolher, assim nós não precisamos ficar colocando muitos protocolos no View Controller, colocando muitas dependências nele, na verdade.

[08:19] Nós fazer o seguinte, aqui nós vamos passar um bloco de completion, então “completion”, eu vou passar a opção que o usuário vai selecionar. A opção é do tipo do enum que nós criamos, que é esse cara. E aqui eu dou um “Void”. Então quando ele clicar em SMS eu tenho que passar para o meu bloco de completion qual é o tipo, qual foi a ação que ele clicou, no caso aqui foi o SMS.

[08:55] Agora que nós já criamos o menu, nós temos que utilizá-lo lá no View Controller, então vou abrir o arquivo “HomeTableViewController” e no método “abrirActionSheet” é que nós vamos utilizar toda essa lógica que nós fizemos.

[09:08] Vou criar aqui o menu, “let menu”. Pra nós o inicializarmos, tem que ser o “MenuOpcoesAlunos”, que foi o que nós criamos, então “MenuOpcoesAlunos” e vou chamar aqui esse método “configuraMenuDeOpcoesDoAluno”. Ele é uma closure, então eu vou dar um enter aqui e ele vai me dar alguma opção que o usuário escolher. E aqui eu vou exibir “self”, como eu estou dentro de uma closure eu preciso do self, e vou exibir o menu. Animado sim, e na closure de completion eu não vou querer fazer nada.

[09:46] Aqui dentro, nós podemos tratar qual foi a opção que o usuário escolheu, então eu posso criar aqui um switch e eu vou fazer um switch da opção que o usuário escolheu, então “switch opcao”, “case”, se for SMS o que eu vou fazer? Eu vou tratar colocando um “print” aqui, por exemplo, então “print(“SMS”)”. Nós vamos testar isso agora.

[10:14] Depois que eu coloquei o meu case, vamos rodar o aplicativo, e vamos ver se está funcionando realmente o menu que nós fizemos. Vamos lá. Agora o aplicativo vai abrir aqui no simulador e com os alunos aqui na tabela, podemos fazer o long press que nós criamos, clica e segura. Apareceu aqui a opção “enviar SMS”. Se eu clicar aqui, repara que ele devolveu no console para nós qual foi a opção que nós escolhemos, que foi SMS. Agora, a partir disso, eu posso começar a criar a lógica da chamada realmente do SMS.

[10:50] Nós só precisamos colocar mais uma opção aqui, o cancelar, que é importante, lembra? E se o usuário fizer um long press, por exemplo, por engano, e ele não quiser fazer nada dessas opções? Então eu tenho que criar aqui, “cancelar” é um “UIAlertAction” e eu vou inicializá-lo passando o título, vamos lá, “cancelar”. O estilo dela vai ser o “default” e aqui eu não vou querer fazer nada. Na verdade o estilo vai ser o “cancel”.

[11:26] E por default ele já some com o menu, então eu não vou precisar configurar nenhuma ação nele, e é só eu adicionar agora. Então “menu.addAction” e eu vou passar aqui o “cancelar”. Vamos rodar o app, para conferir?

[11:44] Com o simulador aberto nós podemos fazer novamente o long press pra conferir o menzinho que nós criamos. Então já temos a opção de enviar SMS e o cancelar, quando eu clico no “cancelar”, ele dispensa o menu, quando eu clico no “sms”, por enquanto ele está exibindo para nós no console o SMS, e a seguir nós vamos começar a implementação realmente do SMS. Então até já.