

07

Ligação telefônica

Transcrição

[00:00] Com o componente de mensagem já configurado e funcionando no nosso app, é a hora de seguirmos com as próximas implementações.

[00:07] No formulário, quando nós vamos cadastrar um novo aluno, nós podemos perceber que temos a opção de colocar o telefone, em que nós já estamos utilizando pra mandar o SMS pro aluno. Porém, nós também poderíamos utilizar esse próprio telefone pra fazer uma ligação pro aluno, porque imagine só, eu abri o formulário, eu quero ligar pro João, o que eu tenho que fazer? Eu vou ter que ficar trocando de app toda hora, o número dele é 999, então vou vir pra cá, 999, eu volto pra cá, vejo o resto do número, 8989, volto pra cá, 8989.

[00:42] Pro usuário isso é um estorvo, ter que ficar toda hora trocando de app pra conseguir fazer uma funcionalidade em que nós já poderíamos ter deixado pra ele mais fácil, ou seja, o que nós temos que fazer agora é implementar a chamada telefônica a partir do nosso app.

[00:58] Se eu fizer um long press aqui, nós vamos criar uma nova opção ligação, em que o usuário não vai precisar ficar trocando de app pra poder fazer a ligação. É nisso que nós vamos mexer agora.

[01:10] De volta no nosso projeto, o primeiro passo é criarmos um novo case no nosso menu, então nós temos aqui a classe menu de opções e nós precisamos adicionar um novo case, nós só temos o SMS por enquanto. Então vamos lá. Vou criar um novo case. “case ligacao”.

[01:32] Agora nós precisamos adicionar um novo botão lá no menu, então vamos criá-lo, “let ligacao = UIAlertAction” e nós vamos inicializá-lo passando o título. O título dele vai ser “ligar”, então vou colocar aqui. O estilo dele vai ser o padrão e aqui na closure de completion nós vamos dar um enter e vamos dar um nome pra essa variável, então vou colocar “acao” também.

[02:00] O que vai acontecer quando o usuário clicar em ligação? Vai cair dentro dessa closure, então eu preciso chamar aqui o completion e vou passar qual é o case do enum que nós criamos, que no caso é ligação. Então vou dar um “ligacao”. Agora que eu criei o botão, eu preciso adicioná-lo no menu. Então vou chamar o “menu.addAction” e vou passar a “ligacao”. O menu já está configurado.

[02:29] Agora nós temos voltar no “HomeTableViewController” e implementar esse novo case que nós fizemos aqui. Aqui eu vou dar um “break”, que é o do SMS ainda e aqui eu vou chamar o case da ligação, que é o que nós criamos. Aqui eu já vou colocar um “break” também e nós vamos implementar a nossa lógica aqui dentro.

[02:53] No iOS, tem uma classe que é instanciada logo quando o app é inicializado, que é uma classe que se chama UIApplication. Na verdade ela se transforma em um singleton, que nós podemos utilizar em todo o ciclo de vida do nosso aplicativo.

[03:08] Essa classe, é responsável por vários eventos que acontecem no iOS, por exemplo, se o nosso app está consumindo muita memória a ponto de dar crash no app, é disparado um método que chama “Did Receive Memory Warning”, em que nós podemos desalocar alguns objetos de memória. Ela também é responsável por abrir alguns aplicativos externos, então, por exemplo, através do meu aplicativo eu posso abrir o Waze, através desse UIApplication, ele faz essa ponte para nós.

[03:39] E é ele que nós vamos utilizar também para conseguirmos fazer a ligação telefônica. Nós vamos criar uma URL, que nós já vamos entender como que funciona, e através dessa URL nós conseguimos chamar a funcionalidade de ligação a partir do nosso app. É através dessa classe UIApplication que nós vamos conseguir todas essas coisas. Então vamos lá.

[04:00] Primeiro nós temos que criar uma URL, para conseguirmos acessar. A URL nós vamos passar qual é a funcionalidade que nós queremos que o iOS abra, seguido do número do telefone do aluno, que, no caso, a funcionalidade que nós queremos é a de ligação.

[04:14] Vamos lá, “let url”. Eu vou inicializar a classe URL e ela tem vários construtores. O que nós precisamos é um que nós passemos uma string, então vamos achar aqui. Esse daqui, esse é o que nós precisamos.

[04:33] Aqui, o que nós vamos fazer? Nós precisamos passar para o iOS qual é a funcionalidade que nós queremos que ele abra. No caso, a funcionalidade que nós queremos que ele abra é a de telefone, então nós temos que pôr esse “tel://” seguido do número do aluno.

[04:48] Como o número do aluno é um optional, vamos criar aqui um “guard let” para nós conseguirmos desembalar esse telefone. Então “guard let numeroDoAluno” é igual ao “alunoSelecionado”, que nós já temos lá em cima ““telefone”. Repara que ele é um optional mesmo, então nós temos que extrair esse valor. Se ele não conseguir, eu dou um “return”. Não dá pra extrair, não tem, deu algum problema, ele nem vai entrar no resto aqui.

[05:20] Agora que eu tenho o número do aluno, eu vou interpolar essa string e vou chamar aqui o “numeroDoAluno”. Então eu tenho a URL que nós precisamos.

[05:33] Só que além disso, a própria documentação da Apple diz para fazermos uma verificação para ver se é possível o iOS fazer esse tipo de recurso. Como nós vimos, a câmera nós não conseguimos testar no simulador, o envio de SMS também nós não conseguimos, e a ligação telefônica também não conseguimos. Então é só em device mesmo. Por isso nós vamos criar o “if let”, para fazermos essa verificação.

[05:58] Então aqui eu tenho a URL que eu estou pegando e se eu conseguir fazer isso, o que eu vou fazer agora? Eu vou chamar a classe UIApplication e verificar se é possível ela abrir a opção da ligação telefônica, então eu posso dar um “.canOpenURL”. Eu passo a própria URL que eu criei no meu “if let”, então vou passar a URL.

[06:24] Se for possível ele fazer isso, ele vai entrar no “if”, ou seja, eu estou utilizando um device de verdade para poder fazer essa ação, então fica fácil. Aí nós chamamos a classe “UIApplication” e damos um “.open url”, então aqui nós passamos a URL que nós criamos no “if let”, então “url”.

[06:45] Aqui ele espera um dicionário, onde aqui é uma string e o valor nós não sabemos qual é. Em opções eu não vou passar nada, então pra eu declarar em um dicionário vazio eu abro e fecho o esquema e coloco os dois pontos. E ele tem uma closure de completion, também não vou querer fazer nada.

[07:06] Então nós já verificamos, já criamos a URL, agora já podemos testar no nosso device de verdade. Vamos fazer isso? Eu estou com meu iPhone aqui e vou rodar o nosso app para testar essa funcionalidade. Vamos lá.

[07:24] Eu estou rodando o aplicativo no meu device mesmo, no meu iPhone de verdade, para nós conseguirmos testar. Vou fazer um long press na célula do João e aqui nós já temos a opção “ligar”, que é a que nós adicionamos no nosso componente de menu. Vou clicar em “ligar”, aí ele vai me dar um popover, falando se eu realmente quero fazer a ligação ou se eu quero cancelar. Eu vou clicar em “Call” pra ele fazer a ligação. E repara que ele vai abrir pra nós realmente a função de chamada telefônica no nosso app.

[07:57] Com isso nós otimizamos a experiência do usuário, ele não precisa ficar trocando entre a função de telefone do iPhone e o nosso app pra ficar vendo o número, e nós implementamos mais um recurso nativo no nosso aplicativo.

Daqui a pouco nós vamos continuar implementando outros recursos.