

09

Deletando aluno

Transcrição

[00:00] Agora falta nós implementarmos a funcionalidade pra excluirmos um aluno. Na table view, se eu fizer um swype, ou seja, se eu clicar numa célula e arrastar um pouco pra esquerda, está vendo que aparece um botão escrito “Delete”?

[00:16] Esse botão é nativo do iOS, ele aparece porque esse método já vem implementado, esse editing style, quando eu criei esse View Controller, eu escolhi o tipo Table View Controller, então por default ele já veio implementado para nós, por isso que quando eu faço swype aparece esse botão do “Delete”.

[00:40] Porém, se eu clicar nele agora ele vai dar crash, porque nós ainda não implementamos nada, então nós precisamos fazer a implementação. É o seguinte, quando clico em “Delete”, vai dar o crash, mas ele cai nesse caso aqui, “.delete”. Aqui então que nós devemos implementar a nossa lógica.

[01:04] Quando nós vamos deletar um aluno, por exemplo, primeiro nós precisamos resgatar qual foi o aluno que o usuário fez o swype e clicou no delete, então vamos lá? “let alunoSelecionado”, nós vamos usar novamente “gerenciadorDeResultados.fetchedObjects”, ele retorna um array de alunos. O que nós queremos pegar está aqui, no “indexPath.row”, então nós conseguimos capturar esse aluno. Nós já pegamos então esse aluno.

[01:37] Na verdade, nós podemos colocá-lo num guard let, “guard let” e nós damos um “else”, assim ele já vem desembrulhado para nós, e dá um “return”. Nós já temos acesso ao aluno.

[01:50] Como nós fazemos então para apagar um aluno? É simples. Na verdade nós vamos aqui. Vamos chamar o “contexto.delete”, ele pede que nós passemos um objeto pra ele, então eu vou passar o “alunoSelecionado”. Será que só isso daqui já faz o aluno ser excluído? Nós já estamos conseguindo deletar esse aluno? Ainda não.

[02:16] Sempre quando nós fazemos alguma operação utilizando o Core Data, nós precisamos salvar, então o que nós precisamos fazer aqui? Precisamos vir no “contexto.save”. Só que ele é um throw function, nós não podemos chamá-lo direto, nós precisamos colocá-lo num “try” e podemos colocar aqui um “do”, pra ele fazer isso. Se der algum problema, “catch”, nós vamos querer descobrir qual foi o erro, então “print”, vamos pôr “(error.localizedDescription)”. Agora sim nós estamos deletando o nosso aluno.

[02:59] Depois disso, essa linha, que é onde eu deleto a célula da table view, eu vou passá-lo pra esse método do delete, que nós tínhamos falado que íamos implementar, eu vou passá-lo pra cá.

[03:12] Pra eu ter acesso ao Index Path, eu preciso primeiro tirá-lo do optional, eu preciso desembrulhar esse Index Path, então eu vou criar aqui um guard let. “indexPath = indexPath”, está vendo que ele é optional? Então nós vamos desembrulhá-lo. Se não der certo eu vou dar um “return”. Agora eu o uso aqui dentro, do Index Path.

[03:43] Vamos rodar o aplicativo e ver se nós estamos conseguindo apagar os alunos que nós já cadastramos? Vou rodar o app e nós já vamos visualizar isso. Então eu vou excluir o Henrique, eu vou fazer um swype e vou clicar aqui em “Delete”. Olha só, apagou. Visualmente apagou. Vamos ver se ele está realmente funcionando? Eu vou rodar o aplicativo de novo pra ele recarregar todas as opções do Core Data e ver se ele realmente apagou do nosso banco. Não está mais aparecendo.

[04:15] Com isso nós fizemos as operações de create, read, update e delete, que seria o crude, utilizando o Core Data. Repara que nós não mexemos em nenhuma linha de código der comando SQL, o Core Data fez todo esse gerenciamento

para nós através do Core Data Stack, que são aqueles caras que nós estudamos.

[04:34] E agora, algumas informações importantes, nós não podemos deixar muitas informações salvas no banco de dados local do nosso app. Ele, na verdade, auxilia o nosso desenvolvimento. É legal trabalhar com Core Data para utilizar pra cache ou pra salvar algumas informações realmente locais.

[04:52] Se forem muitos dados, o ideal é que você trabalhe com web service, com uma API que salve esses registros. Então o Core Data é mais pra esse sentido de nos ajuda a salvar alguns dados.

[05:05] Na próxima aula nós vamos continuar implementando novos recursos nativos do iOS. Nós vamos ver, por exemplo, como que nós chamamos a tela de SMS nativa do iOS a partir do nosso app, então como nós cadastramos aqui o telefone de alguns alunos, nós vamos aprender a utilizar o envio de SMS a partir do nosso app. Então nós já vamos ver isso.