

Exibindo a média dos alunos

Transcrição

[00:00] Pra finalizar, nós precisamos exibir a média dos alunos na tela, mas antes disso repara que nós deixamos aqui o nosso dicionário, onde nós montamos o JSON que nós enviamos pra requisição, fixo, então não faz sentido deixarmos dessa forma, nós só fizemos isso pra testar a chamada da nossa requisição.

[00:19] Agora precisamos refatorar e deixar isso de forma dinâmica, ou seja, a quantidade de alunos que tiver na tabela é a que nós vamos utilizar pra montar esses parâmetros. Então vamos fazer isso.

[00:30] O primeiro passo é termos acesso a esses alunos, por enquanto nós não temos. Então na assinatura do método eu vou pedir essa lista de alunos, então vamos criar um “Array” de “Aluno”. Agora nós precisamos percorrer essa lista pra montar, de fato, o nosso dicionário. Vamos fazer isso através de um “for”, então “for” pra cada “aluno” dentro do nosso array de “alunos”, o que nós vamos fazer? Vamos pegar esse dicionário e colocar dentro do “for”.

[01:03] Antes de montarmos efetivamente o dicionário, nós precisamos tratar algumas coisas como, por exemplo, se eu vier e colocar o “aluno.nome” direto, ele é uma variável do tipo optional, então pode ser que ela não tenha valor.

[01:17] Pra ficar legal, nós precisamos extrair o valor desses objetos que nós estamos percorrendo no “for”, então vamos fazer isso através de um “guard let”, então “guard let nome” é igual ao “aluno” que ele está percorrendo no momento no “for” “nome”. Se ele não conseguir, ele vai vir no “else” e eu vou dar um “break”, pra ele parar o “for”.

[01:40] A mesma coisa com os próximos, então “guard let endereco = aluno.endereco”, se não der certo, “break”. O próximo é o telefone, então “guard let telefone = aluno.telefone else { break }”. Depois do telefone temos o site, então “guard let site = aluno.site” e se não der certo, damos um “break” aqui novamente.

[02:21] Então nós extraímos esses valores. O nota nós não precisamos, porque como ele é double, nós podemos fazer o seguinte, passamos uma “String” e o mandamos aqui dentro, então o “aluno” mesmo que está lá no “for”, “.nota”.

[02:37] Agora que já extraímos esses valores, nós podemos utilizá-lo dentro do nosso dicionário, vamos fazer isso? Aqui onde está “id”, nós vamos fazer o seguinte, vou interpolar, vou colocar o “aluno.objectID”, que é o próprio Core Data que gera pra nós. Aqui dentro nós podemos utilizar esses guard lets que nós criamos, então aqui eu vou colocar o “nome”, aqui eu vou colocar o “endereco”, aqui eu vou colocar o “telefone”, aqui eu vou colocar o “site”. E nós montamos o nosso dicionário de forma dinâmica.

[03:21] Ele vai percorrer todos os alunos e aqui ele vai chamar o nosso array e dar um “append”, ou seja, ele vai adicionar esse dicionário na lista. Quando ele vier pra cá, ele já vai criar todo o dicionário. Então nós deixamos isso de forma dinâmica.

[03:36] O próximo passo é criarmos o nosso alerta, e como nós vamos fazer isso? Porque nós estamos chamando essa função a partir da action do botão calcular média, aqui ele já está dando até algum erro porque falta passarmos a lista de alunos, mas já nós vamos ver isso.

[03:54] Nós precisamos devolver esse dicionário pro View Controller exibir essa média. Então vamos criar uma closure pra fazermos isso. Aqui vou colocar uma vírgula, se der “sucesso”, o que nós vamos fazer? Vamos passar o dicionário com a média, então a média dos alunos nós passamos pro View Controller.

[04:16] Aqui nós vamos passar o “dicionarioDeMedias”, ele é do tipo “Dictionary” de “String, Any”. E se der falha também nós precisamos saber, é importante tratarmos isso. Se der “falha”, o que nós vamos fazer? Vamos passar o erro pra lá, pra conseguirmos tratar. Se der falha nós vamos passar o “error”, que é do tipo “Error”.

[04:53] Agora nós precisamos passar pra nossa closure esses parâmetros que nós criamos, então vamos vir pra baixo e onde que nós temos acesso ao dicionário? Aqui embaixo, onde colocamos esse “print” pra testar. Então vou tirá-lo daqui e vou chamar o bloco de sucesso”.

[05:11] Repara que ele espera um dicionário de “String, Any”. Pra não termos que fazer o casting aqui dentro eu vou fazer o seguinte, “as! Dictionary <String, Any>”. Agora eu chamo o “sucesso” e passo, de fato, o nosso dicionário, que nós recebemos por parâmetro, que é esse carinha aqui.

[05:32] A mesma coisa aqui embaixo. Então vou tirar esse “print”, se der falha ele espera um erro, então eu vou passar aqui um “error”. Nós estamos passando já os parâmetros que nós precisamos, de volta pro nosso View Controller. Agora é só mexermos na chamada do View Controller.

[05:53] Vou apagar esse cara e vou chamá-lo de novo, então o que nós vamos fazer aqui? “CalculaMediaAPI”, instancio, chamo o método “calculaMediaGeralDosAlunos”. Ele espera uma lista de alunos. Nós temos uma lista de alunos que o Core Data nos dá, inclusive temos aqui uma variável que chama “gerenciadorDeResultados”, que nós já utilizamos. Vamos usá-la agora.

[06:16] Como a lista é optional, eu vou criar um “guard let” pra extrairmos esse valor da lista, então “guard let listaDeAlunos = gerenciadorDeResultados.fetchedObjects”, é esse cara aqui. Se ele não conseguir pegar a lista, não faz sentido nós chamarmos o calcula média, então eu já posso dar um “return”.

[06:36] Agora que eu extraí o valor da lista, eu passo esse “listaDeAlunos”. Se der “sucesso”, ele vai devolver um “dicionario”, onde vamos ter essas informações. Vou colocar aqui um “print” e nós vamos chamar um “dicionario”, só pra testarmos. E a falha é a mesma coisa, ele vai devolver um “error”, e nós podemos colocar um “print” do “error.localizedDescription”.

[07:05] A única coisa que nós fizemos foi mexer no nosso JSON e deixar de forma dinâmica e alterar a chamada para recuperarmos essas informações de média do aluno aqui dentro do nosso View Controller, assim nós conseguimos criar o alerta. Então vamos testar pra ver se está funcionando? Eu já estou com o simulador do iPhone 8 e vou gerar um build.

[07:25] Com o simulador aberto, vamos testar. Vou clicar aqui no livro, ele está fazendo requisição e ele devolveu para nós, então a quantidade agora é 2 e a média é 6.5, ou seja, nós estamos passando a lista de forma dinâmica e montando os parâmetros corretamente. Agora, para terminar, nós precisamos finalmente exibir essas informações com o UIAlertController, que é o alerta padrão do iOS. Então vamos fazer isso.

[07:51] Vou criar uma nova classe pra montar esse alerta, então “New File” e, inclusive, nós podemos deixar aqui em “Componentes”, porque se nós precisarmos usar de outros pontos do app, nós utilizamos essa classe, então “Notificacoes”. Vou dar um next e vou criar. Então vamos criar um método aqui, “func exibeNotificacaoDeMediaDosAlunos”. Eu vou devolver um “UIAlertController”, porque é ele que nós vamos utilizar para exibir no View Controller.

[08:35] E eu vou pedir como parâmetro o dicionário, para eu conseguir extrair o valor, então “dicionarioDeMedia” é do tipo “Dictionary<String, Any>”, é isso que nós temos lá. Vamos começar criando aqui o alerta, “let alerta = UIAlertController” e eu vou inicializá-la passando um título, então “Atenção”, “a media geral dos alunos é: “. E aqui eu vou concatenar com alguma variável que eu preciso, que é a média.

[09:17] Só que eu tenho um dicionário aqui, então eu preciso extrair o valor desse dicionário. Eu vou fazer o seguinte, “if let media” é igual ao “dicionarioDeMedia” que eu tenho. Qual que ele está? Se nós virmos aqui, a chave desse dicionário é “media”, então vou passar “media”. Eu vou tentar convertê-la pra “String”, porque é isso que nós vamos utilizar lá.

[09:43] Então aqui eu vou colocar esse alerta dentro do “if”, e aqui eu passo a variável que nós criamos temporária, no “if let”, pra minha mensagem, então vou colocar aqui “media”. Qual vai ser o estilo do alerta? “.alert” mesmo.

[10:00] Agora eu preciso criar um botão pra dar um ok nesse alerta, pra ele fechar, então vamos fazer isso. “let botao = UIAlertAction”, inicializo-a passando um título, qual vai ser o título? Eu vou passar “Ok”, esse vai ser o nome do botão. Ele vai ser do tipo “default” e aqui eu passo “nil”, porque eu não vou fazer nada nessa closure.

[10:23] Agora que eu criei o botão, eu preciso adicioná-lo no alerta, então “alerta.addAction”, aqui eu vou passar o “botao”. E agora dou um “return” do “alerta” que nós criamos. Só que, pensa só, se ele não entrar nesse “if let” que eu criei, significa que ele não conseguiu extrair esse valor “media” de dentro do dicionário, então eu não posso dar um “return nil” aqui direto, porque não faz sentido, já que eu estou falando no retorno que eu vou devolver um “UIAlertController”.

[10:56] Pra eu conseguir dar um “return nil” eu tenho que deixar esse retorno como optional, eu faço isso colocando “?”. A consequência disso é que na hora que eu for implementar, eu preciso tratar para verificar se tem valor ou não, para não dar nenhum crash no nosso app.

[11:14] Criei o nosso alerta, agora vamos voltar no View Controller e implementar esse alerta. Vou fazer o seguinte, “if let alerta” é igual a classe que nós criamos, “Notificacoes.exibeNotificacaoMediaGeralDosAlunos”, aí ele pede que eu passe o dicionário, então vou passar o “dicionario”. E dentro do “if” eu tenho acesso ao alerta, se ele conseguir criar. Vou fazer o seguinte agora, como eu estou dentro de uma closure, eu preciso passar o self pra chamar o método present, que é do View Controller mesmo.

[11:52] Vou passar aqui “self.present”, eu passo o “alerta” do “if let”. Animado, sim, e no bloco de completion eu não vou fazer nada. Vamos testar então? Vou rodar aqui o aplicativo.

[12:07] Com o simulador aberto, eu vou clicar aqui no livro. Cliquei, olha que legal, nós estamos exibindo aqui um alerta mostrando a média geral dos alunos. Cliquei em “Ok”, ele desaparece com o alerta. Com isso nós fechamos a primeira parte do curso, onde vimos vários conceitos e aplicamos na prática de recursos nativos do iOS.

[12:28] Nessa última aula nós vimos como nós fazemos uma requisição de forma nativa, ficou um pouco extenso, mas é importante sabermos como é feita a requisição antes de utilizarmos essas bibliotecas que facilitam pra nós, como o Alamofire, ela é uma biblioteca bem legal, mas é importante sabermos como nós utilizamos esses recursos também de forma nativa.