10

Separando as responsabilidades do ViewController

Transcrição

[00:00] Olá, pessoal! Tudo bem? Agora a gente já terminou toda a implementação do nosso aplicativo. Porém, vamos dar uma revisadadinha nos View Controllers que a gente implementou para ver se tem algum código que a gente pode melhorar, ou separar as responsabilidades do nossos View Controllers que não diz respeito a eles.

[00:18] Então, vamos começar pela primeira tela. Olha só, a gente tem aqui a tela que a gente fez no primeiro curso de layout, que é a tela onde lista as viagens. Vamos abrir o View Controller para dar uma analisada.

[00:27] Então vou vir em "Home", "Controller" e abrir esse primeiro arquivo. Então a gente tem uma implementaçãozinha que a gente já fez. Repare que, no método aqui, no "cellForRow" da table, a gente tem toda a parte visual da célula dentro do nosso Controller.

[00:44] Então, isso daí não está muito legal. Não que vá prejudicar o funcionamento do nosso app, nem nada, mas pensa só, a gente tem uma classe, que essa "TableViewCell". É uma classe específica para cuidar dessa parte visual da célula.

[00:58] Então, não tem necessidade da gente deixar, por exemplo, a implementação, a customização das células dentro do nosso View Controller, porque não é ele o responsável por fazer essa customização e sim a célula.

[01:09] Então, vamos começar a refatorar esse View Controller. Qual é o primeiro passo que a gente deve fazer aqui? A gente pode tirar toda essa parte visual de dentro do método cellForRow e colocar dentro da classe que cuida da parte visual da nossa célula. Então, como é que eu faço?

[01:25] Vou segurar aqui a tecla "Command" e clicar em cima da classe "TableViewCell". Cliquei, ele vai abrir, e a gente pode criar uma funçãozinha para ele customizar essa celula pra gente. Então vamos criar essa função.

[01:38] "func", por exemplo, "configuraCelula", então a gente pode passar aqui como parâmetro o nosso objeto viagem, que a gente usa para setar os nossos Outlets. Então, "Viagem" e a gente cola toda a implementação do método CellForItem aqui dentro. Eu vou dar um "Command + V" e vou apagar todos esses Cells aqui.

[02:02] Por quê? Porque, como a gente já está dentro do arquivo da célula, a gente não precisa, a gente não tem mais referência desse arquivo. A gente já pode acessar direto os Outlets. Então, aqui a gente está pegando, por exemplo, labelTitulo, mas ela já está aqui.

[02:15] Ela já está aqui no Scope da classe, então a gente não precisa mais desse Cell, então a gente pode remover todos eles. Então a gente vai selecionar e apagar todos eles, como a gente já viu, a gente está no mesmo escopo, então a gente consegue acessar diretamente aqui os nossos elementos.

[02:32] Como agora a gente está passando aqui a viagem por parametro, a gente vai substituir esse "viagemAtual" por "viagem", que é o nome que está na assinatura do método. Então a gente vai colar, onde está a "viagemAtual", a gente troca por "viagem". Vamos trocar aqui.

[02:48] Agora aqui, pessoal, repara que a gente tem essa "labelQuantidadeDeDias", mas ela está fixa. Então imagine que uma viagem, por exemplo, por acaso tenha um dia só. A gente vai exibir como? A gente vai exibir dias, da mesma forma.

[03:01] Então, aqui na primeira tela, por exemplo, Paraíba. Vamos supor que dure só um dia, por algum motivo, a gente não sabe. E a gente vai continuar apresentando 1 e a Lable "dias", ao invés de "1 dia". Então, vamos tratar esse caso.

[03:16] Então, o que a gente pode fazer aqui? A gente pode fazer um if ternário. Então, por exemplo: "viagem.quantidadeDeDias = 1"? Se for, a gente vai fazer o seguinte, a gente vai colocar "1 dia".

[03:33] Se não for, a gente vai interpolar a nossa string. A gente vai pôr a quantidade de dias que tem o nosso objeto, " (viagem.quantidadeDeDias) dias". Então, se for um dia só, a gente exibir a string dia, se for mais que um, dias. Então a gente consegue tratar esse caso já aqui.

[03:55] Feito isso, pessoal, vamos rodar o app para ver se não tem nenhum erro de sintaxe e ver se está tudo funcionando da forma estava? Então, olha só, antes da gente rodar o app, a gente precisa chamar o método.

[04:05] Então, aqui embaixo, vamos chamar já. A gente vai pegar "cell.configuraCelula" e a gente passa o nosso objeto "viagemAtual" aqui. Beleza, vamos rodar o aplicativo.

[04:17] Então, com o simulador aberto, pessoal, repara que deu um probleminha aqui. Está vendo que, em alguns casos, a imagem está sobrepondo as Labels? Então, em alguns ela aparece, em alguns ela está sobrepondo.

[04:29] Vamos setar uma Constraint de altura nessa Label, que já vai resolver esse problema. Então, vamos selecionar aqui essa Label e colocar uma Constraint aqui de Top. A gente pode rodar o app de novo, que a imagem vai respeitar já esse espaço, então não vai ter problema.

[04:46] Então vamos rodar o app e conferir para ver se está aparecendo a imagem certinho. Então, olha só, agora ele aparece a Label. Ele está restringindo o espaço da imagem, então ela não está mais sobrepondo.

[05:00] Então a gente conseguiu tratar aquele caso. Olha só, agora aqui está um dia. Primeiro View Controller a gente já conseguiu refatorar esse código. Vamos para o de Pacotes, pessoal?

[05:11] Então eu vou abrir aqui o View Controller de Pacotes, a gente tem o mesmo caso. A gente tem toda a parte visual da célular dentro do nosso Controller. Vamos deixar isso dentro da cell, para ficar mais enxuto o nosso Controller e separar essa responsabilidade que diz respeito à célula?

[05:26] Então, mesmo esquema, pessoal. Eu vou recortar toda essa parte visual aqui da célula, recorto e a gente pode colocar ela dentro da classe da célula, que é esse "PacoteViagemCollectionViewCell".

[05:39] Cliquei, a gente pode criar um método aqui embaixo. "func configuraCelula" e, como parâmetro, a gente passa o "pacote". Nesse caso, o objeto é o "pacoteViagem", ele é do tipo "PacoteViagem". A gente cola toda implementação aqui.

[05:58] Como a gente também não vai precisar da referência "celulaPacote", a gente nem tem aqui e a gente está dentro do mesmo escopo, a gente não vai precisar, então, desse "celulaPacote". Então, a gente pode apagar ele de todos os elementos. Então, vamos selecionar aqui e apagar.

[06:17] Aqui embaixo é a mesma coisa. Onde a gente está arredondando e colocando a borda, vamos apagar essa referência do "celulaPacote". Agora que a gente apagou, a gente pode renomear aqui.

[06:28] Onde está "pacoteAtual", vamos pegar esse um "pacoteViagem", que está na assinatura do método, e colar em cima dele, porque é esse o nome que a gente está utilizando na assinatura do método.

[06:39] Agora a gente já pegou a parte visual que estava no nosso Controller e colocou dentro da classe da célula. Então já está um pouquinho mais organizado o nosso código.

[06:51] Agora, pessoal, a gente pode voltar lá e chamar o nosso método "configuraCelula". Então, vamos chamar ele aqui. Vamos pegar o "cell.configura", "celulaPacote", aqui no nosso caso, "celulaPacote.configuraCelula". E a gente passa para o parâmetro, o "pacoteAtual". Então, passamos aqui para o parâmetro.

[07:13] Agora, pessoal, a gente tem que refaturar o nosso Search. Quando a gente implementou, a gente estava utilizando o objeto Viagem. Então, a gente estava mostrando, ao invés dos pacotes, as viagens.

[07:24] Porém, depois, a gente refaturou e implementou, realmente, os pacotes de viagem. E aqui no nosso Predicate, a gente está passando o formato "título" direto. Só que, esse título, ele está dentro do nosso objeto Pacote.

[07:39] Então, vamos relembrar. Vamos vir aqui em "Model". Então, olha só, em "PacoteViagem", a gente tem objeto Viagem. Então, dentro do objeto Viagem, a gente tem título. Então, a gente precisa mudar lá, a gente não pode ter acessar o título direto, a gente tem que primeiro acessar o objeto Viagem.

[07:56] Então, "viagem.titulo", aí sim a gente vai verificar. Vamos rodar aqui de novo? Legal, pessoal, então vamos testar. Vamos vir aqui em "Pacotes", olha só a nossa implementação, não mexemos em nada, a gente só mudou o código de um lugar para o outro. E vamos testar o Search.

[08:13] Vamos digitar aqui de novo, "Natal". Então, ele está funcionando. Búzios, legal, Porto de Galinhas. Então, o nosso Search já está funcionando, porque a gente estava utilizando um objeto, depois a gente teve que trocar. Então, por isso a gente refaturou.

[08:31] Com isso, a gente conseguiu isolar as responsabilidades que não eram do nosso View Controller, para os arquivos específicos das células. Então, a gente deixou nosso Controller um pouquinho mais enxuto. Então, agora cada um tem sua devida responsabilidade.

[08:45] Agora, pessoal, para finalizar, vamos fazer o seguinte, vamos buildar aqui o nosso app no iPad Air 2, por exemplo. Vamos clicar aqui e vamos rodar o app. Beleza, galera, agora com o simulador do iPad, vamos entrar nas outras telas, porque, até então, a gente implementou tudo no tamanho do iPhone 5, do iPhone SE e a gente não rodou mais nosso app no iPad.

[09:07] Então, agora vamos dar uma olhada para ver como é que está a nossa interface. Então, vamos mudar aqui para o Pacote. Então, olha só, as imagens você viu que já está bem destorcida.

[09:18] Vamos clicar em cima de alguma aqui, "Buzios", por exemplo. Aqui ele está mostrando a imagem, apesar de estar meio estranho a resolução dela. As Labels estão com mesmo tamanho do iPhone, então isso não está muito legal.

[09:36] Imagina um usuário com o iPad, que tem uma tela grandona, vendo essas Labels pequenininhas. A gente pode melhorar a interface, melhorar a experiência do usuário. Nosso app está meio feinho assim.

[09:48] E, além disso, pessoal, está vendo que os campos de texto está bem apertadinho e tudo mais? Vamos clicar aqui em "finalizar compra". Cliquei aqui, beleza. Aqui também, a mesma coisa.

[10:00] Aqui está torto, a Label aqui do nosso header. As Labels aqui estão bem pequenininhas, a imagem está distorcida. Então, a gente tem aí um trabalhinho para fazer. A única tela que está okay é a primeira, que a gente tinha feito na primeira parte do curso, já configurado ela.

[10:17] Agora é hora da gente continuar customizando os elementos para visualização de iPad. Então a gente já vai começar a fazer isso.