

## Utilizando o protocolo UICollectionViewDelegate

### Transcrição

[00:00] Oi pessoal, tudo bem? Então, a gente acabou de terminar de implementar a tela de pacotes de viagens. A gente já conseguiu fazer o nosso search funcionar, a gente implementou uma collection.

[00:10] Só que pensa só, para um aplicativo de viagem, como você vai saber, por exemplo, quais informações vão ter nesse pacote ou quantos dias? Então, a gente precisa agora continuar a implementação do nosso aplicativo, criando uma próxima tela, a tela onde vai dizer, por exemplo, quais informações da viagem que eu selecionar.

[00:30] Então, olha só, vou selecionar uma viagem, um pacote de viagem, e ele vai mostrar para mim as informações como, o que está incluso, qual é a data, o preço, quantos dias.

[00:41] Então, a gente tem que continuar a implementação do nosso aplicativo com uma nova tela, a tela de detalhes. Então, a gente vai continuar fazendo isso.

[00:51] Aqui no storyboard, pessoal, a gente vai precisar de uma nova tela. Então, quando a gente quer criar uma nova tela na nossa aplicação, como é que a gente faz? A gente vem aqui no Object Library e pode escolher um View Controller, então a gente vai clicar nele e arrastar para dentro do storyboard.

[01:04] Assim como a gente fez nas outras telas que a gente criou, a gente vai continuar seguindo um design proposto. Então, olha só, a próxima tela que a gente vai implementar a gente tem que seguir essas especificações. Então, vamos começar fazendo passo a passo, vamos começar pela parte de cima.

[01:22] Então, olha só, vamos analisar, a gente tem uma imagem, a gente tem várias labels, a gente tem alguns ícones, algumas imagens aqui embaixo, outra label, text field, são elementos já conhecidos.

[01:37] Então, vamos começar fazendo passo a passo, vamos começar arrastando uma imagem do nosso Object Library para dentro do View Controller. Vamos procurar aqui por image e arrastar, vamos posicionar ela aqui no centro.

[01:53] Vamos seguir aqui o gabarito. Olha só, reparem que ela tem uma altura de 185, então vamos deixar essa imagem já com a altura correta, eu vou vir aqui e vou colocar 185 e vou prender ela aos lados, o lado direito e agora o lado esquerdo.

[02:15] Mas, antes disso pessoal, reparem que a gente tem uma lista de objetos, um embaixo do outro, então, a gente não pode continuar aqui arrastando os elementos.

[02:28] Vamos supor, eu vou pegar aqui uma label, eu vou arrastar para cá e vou digitar Porto de Galinhas. Se a gente fizer dessa forma, a gente vai precisar aplicar as constraints, então eu posso posicionar os elementos aqui livremente e depois aplicar as constraints.

[02:48] Só que, no começo do curso, a gente viu que para dar manutenção nas constraints, não é tão simples assim, é um pouco custoso, porque se a gente precisar apagar ou modificar o layout do nosso app, a gente vai ter que ficar descobrindo, qual constraint a gente vai ter que apagar, qual a gente vai ter que pôr. Então, a gente aprendeu uma forma mais fácil da gente implementar isso que é utilizando o Stack View.

[03:11] Então, olha só, como a gente já percebeu a gente tem um elemento embaixo do outro aqui no nosso View Controller. Então, a gente pode utilizar um Stack View que deixa esses elementos empilhados.

[03:22] Vamos procurar aqui no Object Library por Stack View, então olha só, lembra que a gente tem dois tipos de Stack View? Um que ele deixa um elemento ao lado do outro e um que ele empilha um elemento embaixo do outro.

[03:35] Então, a gente vai utilizar um Stack View que empilha um elemento embaixo do outro porque no nosso design a gente tem um elemento embaixo do outro, então vamos pegar aqui um Stack View vertical e arrastar para dentro do nosso View Controller.

[03:49] Vamos apagar a imagem e a label e a gente re-coloca ela dentro do Stack View. Com o Stack View aqui na tela, a gente pode deixar ele do tamanho do nosso View Controller.

[03:59] Então, vamos arrastar ele, vamos dimensionar ele para que ele ocupe todo o espaço, assim, a gente consegue colocar os elementos dentro.

[04:05] Já vamos aplicar as constraints porque, em alguns casos, a gente vai precisar colocar os objetos dentro e aplicar alguma constraint de altura, então a gente precisa que ele tenha constraints nas laterais.

[04:18] Agora que a gente já tem um mecanismo de empilhamento dos nossos elementos, a gente pode continuar com a implementação. Então vamos lá, vamos colocar uma imagem novamente, então vamos procurar por image.

[04:30] Repare que quando a gente arrastou, ele preencheu todo o Stack View. A gente já estudou isso e era por causa que a configuração do Stack View, a distribuição está fill, que significa preencher, então, quando a gente coloca um elemento na tela, ele ocupa todo o espaço, ele preenche todo o espaço, mas isso a gente pode configurar daqui a pouco.

[04:51] Vamos continuar arrastando as outras labels, a gente tem aqui uma label de título. Vamos pôr Porto de Galinhas, por exemplo, então vamos procurar aqui por label, vamos clicar e arrastar para o nosso View Controller. Então, olha só, a gente já arrastou a label e ela não apareceu.

[05:07] Vamos diminuir aqui, por exemplo, a altura da imagem só pra gente conseguir visualizar a nossa label. Embora, na hora que a gente rode o app, ele vai empurrar todos os elementos para baixo, mas aqui, visualmente a gente consegue alterar as propriedades.

[05:21] Eu vou clicar em cima da label e mudar a propriedade text da label. Então vamos pôr aqui, por exemplo, Porto de Galinhas, vamos deixar isso fixo, daqui a pouquinho a gente deixa tudo dinâmico.

[05:34] A gente já tem um título, vamos colocar outra label que é pacote, então, vamos clicar e arrastar. A próxima label é "Pacote: Hotel + café da manhã", então, vamos mudar a propriedade de texto para isso Pacote: hotel + café da manhã.

[05:55] A próxima label, pessoal, é a label que indica a quantidade de dias. Então, aqui, 7 dias, vamos clicar e arrastar essa label para o nosso View Controller e a gente altera a propriedade de texto para 7 dias. [06:09] A gente colocou algumas labels, colocou uma imagem, vamos já formatar essas labels. Olha só, a primeira tem o título Porto de Galinhas, ela tem o tamanho 23 e a fonte é Avenir Black. Então, vamos clicar aqui e mudar a fonte para Custom, a gente vai escolher Avenir e o estilo dela é Black.

[06:30] Então, a gente escolhe a opção Black e o tamanho 23, vamos mudar aqui para 23. Como a gente já usou a cor que ele está pedindo aqui, que é o 85, a gente não precisa colocar o RGB de novo, na nossa paleta de cores ela está recente.

[06:48] Agora vamos alterar as próximas labels, olha só, a próxima é a Avenir Medium, tamanho 17, então vamos clicar nela e alterar novamente a fonte. Vamos pôr Custom, aqui em fonte a gente escolhe Avenir, o estilo é Medium, tamanho 17 já está. Vamos alterar a cor para cor recente.

[07:11] Essa é a mesma coisa, vamos dar uma olhada? Avenir Medium 17, então mesmo esquema, vamos alterar a fonte Custom, family, que é a família de fontes, a gente vai escolher Avenir, o estilo é Medium, tamanho 17 já está. A gente

altera a propriedade de cor para a cor recente que a gente já estava utilizando.

[07:36] Mas pensa só, pessoal, por onde a gente vai chamar essa tela, porque a gente tem aqui um tab bar que a gente colocou as duas telas da nossa aplicação. Então, olha só, a gente tem a primeira tela que é a tela onde lista as viagens, a segunda tela onde lista os pacotes. Porém, a gente colocou tudo isso para o Tab Bar Controller gerenciar.

[07:56] Então, eu tinha explicado que o Tab Bar agrupa as funcionalidades do nosso aplicativo. Só que essa tela não é uma funcionalidade que ela pode ficar solta no nosso app, ela depende de alguma coisa.

[08:08] Como assim? Então, olha só, a gente aqui com o simulador aberto, quando que a gente vai mostrar essa tela? Então, eu quero saber mais, por exemplo, sobre o Ceará, então como é que eu faço para chamar essa tela?

[08:20] O ideal seria que quando eu clicar em cima de Ceará, eu chame essa tela passando as informações, passando o objeto Ceará como parâmetro. Então, vamos aprender a chamar essa tela a partir do click de um item da collection.

[08:33] Para isso, pessoal, a gente vai ter que voltar no nosso View Controller de pacotes e vai ter que implementar um método da collection.

[08:44] Qual método é esse? É um método que quando eu clico em cima de um item, eu consigo disparar uma ação. Então, olha só, para chamar esse método novamente a gente precisa de um protocolo, a gente precisa implementar um protocolo que é UI Collection View Delegate.

[08:59] Vamos clicar aqui, colocar uma vírgula e digitar UI Collection View Delegate. A gente implementando esse protocolo, pessoal, a gente tem que setar o delegate no nosso View Controller novamente.

[09:14] Inclusive, a gente já colocou a nossa Collection View. Delegate igual a self, então, como a gente já setou, a gente já pode implementar o método. Vamos colocar esse método de delegate em baixo dos métodos da collection, então, o último que é esse Collection View Layout, vamos colocar ele aqui embaixo.

[09:32] O método que a gente precisa é o método de Select Item, então acabei de selecionar um item na collection, esse método vai ser disparado.

[09:40] Vamos fazer um teste, com o método implementado, vamos colocar aqui um print para mostrar aqui no console alguma mensagem quando a gente clicar em uma viagem, então vamos colocar aqui, por exemplo, uma string e colocar assim "cliqueu em uma viagem". Vamos rodar o app para ver se está funcionando? Então, a gente vai vir aqui em run e rodar o aplicativo.

[10:04] Com o aplicativo rodando, a gente vai vir na tela de pacotes e a gente vai clicar em cima de um item aqui da collection.

[10:11] Então, eu vou clicar em cima de Rio de Janeiro, por exemplo, cliquei. Olha só no console, ele já está disparando um método que a gente implementou, "cliqueu em uma viagem", que é esse print que a gente colocou aqui só para testar o método, então está funcionando.

[10:25] O que a gente precisa fazer agora? A gente precisa chamar a nossa tela. Como a gente faz isso? Primeiro, a gente tem que ir no storyboard e saber qual dessas telas a gente quer chamar. Então, olha só, aqui no storyboard, que é esse arquivo aqui, Main, a gente tem várias telas.

[10:42] Tem a primeira que é de viagens, tem a de pacotes e tem essa última de detalhes, que é a que a gente criou. Só que a gente ainda não criou nenhum arquivo para controlar esse View Controller.

[10:53] Lembra que quando a gente trabalha com uma nova tela, a gente tem que ter um arquivo que a gente vai usar para ele gerenciar todos os elementos que tem dentro, os métodos e tudo mais? Então falta a gente criar esse arquivo.

[11:05] Assim como a gente fez com as outras funcionalidades, a gente criou as pastinhas para organizar o projeto. Vamos criar uma nova pasta para a tela de viagens, então a gente vem aqui em New Group, a gente vai colocar, por exemplo, "Detalhes da viagem".

[11:21] Vamos arrastar essa pastinha para baixo da Pacote, aí dentro dela, por costume, a gente já cria duas pastas também. Uma que é a pasta View que vai cuidar dos arquivos de visualização e a outra pasta que a Controller que é a pasta que vai gerenciar os Views Controllers. Coloquei aqui a minha pasta View e a minha pasta Controller.

[11:45] Com a pasta Controller criada, pessoal, vamos criar o View Controller para detalhe da viagem. Então, a gente vem aqui em New File, dá um next e aqui a gente escolhe subclasse de UI View Controller mesmo, e a gente dá um nome "Detalhes Viagens View Controller", então, a gente vai criar o arquivo.

[12:09] Com o arquivo criado, pessoal, a gente tem que voltar aqui no storyboard e setar o nosso arquivo. Aqui em classe, a gente vai digitar o nome do arquivo que a gente criou, "Detalhes Viagens View Controller", então, com isso a gente já setou o View Controller para esse arquivo aqui.

[12:29] Agora o que a gente precisa fazer? No click do item da collection, a gente precisa chamar essa tela. Então, vamos voltar aqui na tela de Pacote, no View Controller, e no método do Did Select Item na Collection. Então, olha só, vamos apagar essa linha que a gente colocou só para teste.

[12:47] O que a gente precisa fazer? No storyboard, a gente precisa recuperar aquele View Controller, então, olha só, a gente tem vários controllers aqui. A gente tem a primeira tela, a gente tem a tela de pacotes e tem essa última tela que é a tela de detalhes que a gente criou.

[13:03] Então, a gente precisa criar uma variável com essa tela pra gente poder apresentar na tela. Como é que a gente faz isso? No método, a gente pode criar, por exemplo, uma constante no storyboard pra gente poder buscar o nosso View Controller, então, vamos colocar aqui storyboard, ele vai ser UI Storyboard.

[13:23] Aqui, a gente precisa dar o nome do nosso storyboard. Qual é o nome do nosso storyboard? O nome do nosso storyboard é Main, que já vem por padrão, então a gente vai abrir aqui uma string e colocar Main. O bundle, a gente passa New, porque está nessa estrutura mesmo, aí a gente já tem o storyboard.

[13:43] Agora a gente vai criar uma variável, pessoal, para recuperar o nosso View Controller. A gente vai criar uma constante, let Controller, a gente vai utilizar a nossa variável que a gente criou, nossa constante do storyboard. Então, olha só, UI Storyboard, ponto, Instantiate View Controller.

[13:57] Então, a gente vai precisar de um identificador para o nosso View Controller. Então, vamos vir aqui no storyboard e setar um identificador. Como é que a gente faz isso? A gente clica em cima do View Controller.

[14:19] Aqui no storyboard ID tem parte de identify, identificação, então, a gente precisa dar um identificador para o nosso View Controller, porque, senão como é que a gente vai saber, qual é o Controller que a gente quer? A gente tem o primeiro, a gente tem a tela de pacotes, a gente tem a tela que a gente acabou de criar, então como a gente vai saber qual View Controller a gente quer? Colocando um identificador.

[14:40] Então, vamos lá, a gente clicou em cima do View Controller, aí a gente vem em identify. A gente pode pôr, por exemplo, "Detalhe". Aqui, a gente pode pôr qualquer nome, então, vamos pôr detalhes, por exemplo. Vamos copiar e vamos colar esse cara aqui no método de Select Item, no Controller, a gente tem o parâmetro de identificador, então a gente vai passar ele aqui. Então, vamos abrir uma string e colocar detalhes.

[15:11] Agora, a gente precisa apresentar esse View Controller na nossa tela. Como é que a gente faz? A gente dá um self, ponto, present, que é um método do próprio UI View Controller, então, a gente passa um UI View Controller.

[15:27] Qual é o controle que a gente criou? Esse aqui, então, a gente vai passar o controller, se a gente quer animado ou não, vamos passar verdadeiro e se a gente quer fazer alguma coisa, depois que ele chamar esse View Controller. Como a gente não quer fazer nada, pode passar nil nesse parâmetro.

[15:43] Agora a gente tem que converter, fazer um casting desse Controller para o arquivo que a gente criou. O arquivo que a gente criou não é esse "Detalhes Viagens View Controller"? Então, vamos colocar aqui um as, Detalhes Viagens View Controller.

[15:59] Recapitulando, pessoal, no click do item tem da Collection, a gente vai buscar no storyboard. A gente deu o nome de Main porque é o arquivo padrão do storyboard, o Main como storyboard.

[16:11] Aí, dentro do storyboard, qual View Controller a gente quer? A gente quer o que tem o identificador detalhes. Qual é o que tem o identificador detalhes? O que a gente acabou de criar, que a gente colocou aqui o identificador de detalhes. Então, a gente quer esse View Controller.

[16:27] O que a gente faz depois disso? Depois disso, a gente dá um present, a gente apresenta na tela esse View Controller. Vamos testar para ver se está funcionando? Então vamos dar um run.

[16:38] Então, vamos tentar abrir a tela que a gente acabou de criar, como é que a gente faz? A gente vem aqui em Pacotes, que é o método que a gente implementou, o de Select Item, que é o método que quando o cara clicar em cima de um item, ele vai ser disparado, chamando nossa tela.

[16:54] Vamos escolher qualquer item aqui da nossa coleção, por exemplo, Rio de Janeiro, então, vou clicar aqui e, olha só, a gente conseguiu apresentar a tela que a gente acabou de criar.

[17:04] Tem alguns probleminhas ainda, como, por exemplo, se eu quiser voltar para as telas anteriores, como é que eu faço? Ainda não dá. E a gente tem umas labels com os valores fixos e uma imagem zona aqui, então isso a gente vai começar a tratar daqui a pouquinho.