

## Finalizando a tela pacote de viagens

### Transcrição

[00:00] De volta ao projeto pessoal, a gente está com nosso app rodando aqui no iPhone 5. Repare que, por exemplo, falta ainda a gente alterar as labels. Olha só, a label "Pacote" e essa label aqui "6 pacotes encontrados", ela está numa formatação diferente ainda do no nosso design. Então, vamos começar a trabalhar nisso.

[00:21] Vamos ver aqui a label Pacotes, a gente tem que deixar ela com formatação, com a fonte Avenir Medium, tamanho 24, então, vamos voltar aqui no nosso storyboard. Repare que as labels não estão aparecendo aqui no nosso View Controller.

[00:34] Vamos abrir aqui o Stack View, a label Pacotes, por exemplo, se a gente vir aqui, ela está com uma altura de 0, o Stack View está prendendo ela, não está deixando espaço, vamos arrumar isso.

[00:45] Vamos aplicar uma constraint de 30, assim a gente consegue visualizar a nossa label, olha só. Vamos fazer a mesma coisa com a label 6 pacotes, vamos selecionar a label, abrir o popover de constraint e colocar uma altura aqui de 30.

[01:01] Agora a gente já está conseguindo visualizar as labels. Vamos selecionar a label Pacote e aplicar a formatação que a gente precisa.

[01:08] Então, vamos vir aqui em Custom, a fonte vai ser Avenir, o estilo vai ser Medium, tamanho 24. A cor, como a gente já utilizou nas outras vezes, ela vai estar nessa paletinha aqui de recentes, então a gente não precisa digitar de novo, só escolher a cor.

[01:30] Vamos seguir agora com a formatação dessa próxima label. Então, aqui, a gente tem Avenir Medium, tamanho 17, então vamos voltar lá, vamos mudar aqui a cor e a fonte para Avenir Medium, tamanho 17.

[01:47] Então, aqui Avenir, o estilo Medium, tamanho 17, como já está 17, vamos deixar assim mesmo. Agora, vamos rodar o aplicativo pra gente testar um outro tamanho de iPhone, pode ser o iPhone 6, por exemplo. Então, vamos vir aqui no simulador e trocar para iPhone 6, aí vamos dar um run, vamos rodar o app.

[02:11] Beleza pessoal, a formatação que a gente colocou na label está ok. Agora, repare que a collection está um pouco estranha. A gente estava até agora testando no iPhone 5, estava legalzinho, agora quando a gente rodou no iPhone 6, ela ficou com um espaçamento muito grande entre uma célula e outra.

[02:29] Olha só, a gente tem um espaço aqui que a gente está desperdiçando, o ideal seria a gente aproveitar melhor esse espaçamento para deixar as células um pouquinho maior para o usuário conseguir ver um pouco mais da imagem e algumas informações a mais.

[02:44] Inclusive no nosso design, olha só, a gente tem um espaço menor, a gente tem um espaço mais agradável aqui. Então, a gente vai começar a mexer nisso agora.

[02:54] Quando a gente trabalha com a Collection, a gente consegue alterar o tamanho, no caso a largura ou altura da célula dinamicamente, porque o que a gente fez foi deixar uma largura fixa, baseada no Iphone 5 que a gente estava utilizando.

[03:09] Quando a gente rodou o app no iPhone 6, continuou com a mesma largura, o ideal seria que esse valor se modificasse dinamicamente, de acordo com o tamanho do device. Tem como a gente fazer isso.

[03:20] Vamos voltar aqui no nosso View Controller, Pacotes de Viagens, e vamos continuar a implementação. Quando a gente trabalha com a Collection, a gente tem um protocolo que a gente pode implementar e que a gente pode trocar o valor da altura e da largura da Collection. Então, a gente vai fazer isso agora.

[03:26] Então, olha só, na frente aqui da Collection View, data source, a gente vai colocar uma vírgula e vai implementar um outro protocolo, que é esse protocolo aqui, `Ui Collection View delegate for layout`, esse cara aqui esse.

[03:54] Esse protocolo, ele habilita a gente a implementar o método que altera as propriedades de largura e altura da célula, só que para fazer isso, a gente tem que setar o delegate da nossa collection.

[04:06] A gente tem que falar, esse View Controller vai implementar esse método, pode ficar tranquilo, então vamos fazer isso. Vamos pegar nossa Collection View, Pacotes. Vamos colocar aqui ponto, delegate, e colocar `selfie`.

[04:20] Já setamos o delegate, já falamos que esse View Controller vai implementar o método. Agora aqui, a gente pode vir aqui embaixo, apagar esses comentários que a gente não vai utilizar e implementar o método.

[04:33] Vamos chamar aqui Collection View, a gente vai utilizar esse método aqui Collection View layout, então, a gente pode dar um enter que ele já completa pra gente. Repara que ele espera como retorno um CG size, um tamanho.

[04:48] Então vamos pensar, olha só, a gente tem aqui uma Collection e quantos itens a gente quer que ela tenha por linha? A gente quer que tenha dois itens por linha, que caibam dois pacotes de viagem, então, o que a gente pode fazer? A gente pode pegar a largura toda da collection e dividir por 2.

[05:08] Dividindo por 2, a gente pode usar esse tamanho para ser a largura da célula, assim a gente consegue aproveitar mais o espaço e não deixar esse espaço aqui do lado.

[05:18] Então, a gente pega a collection, pega a largura dela, divide por 2, pega esse valor e seta como largura da nossa célula, assim, a gente consegue aproveitar mais. Então, como é que a gente pode fazer isso? Como ele pede aqui, como retorno, um CG size, a gente já pode vir aqui, dar um return, CG size. A gente tem um método em que a gente passa a largura e a altura da célula.

[05:45] Então, olha só, a largura a gente quer pegar a collection, então vamos pôr aqui a nossa Collection View, que é esse parâmetro aqui. Como a gente quer acessar as propriedades de tamanho, largura e altura ou se a gente quisesse pegar a posição que ela está, tem uma variável que faz isso, que é essa ponto bounds.

[06:07] Aí a gente consegue acessar a largura e altura da nossa Collection. Então, olha só, eu vou pegar aqui ponto width, que é a largura, e a gente vai dividir isso por 2. A altura, a gente vai deixar como está, vamos passar aqui, por exemplo, 160 que é o que já estava. Com isso, teoricamente é para funcionar, vamos testar? Então vamos rodar o app novamente.

[06:33] Bom pessoal, acabamos de rodar o app no iPhone 6 e olha só, a nossa Collection está mais estranha ainda porque a gente implementou aqui uma lógica de pegar a largura da Collection e dividir por 2 para caber as outras células. Só que isso não aconteceu, ele está renderizando apenas uma célula por linha. Vamos entender porque que isso aconteceu.

[06:56] Quando a gente tem uma Collection View, por exemplo, a gente tem uma largura para ela. Aqui, a gente fez a simulação no iPhone 5, então, olha só, a Collection tem uma largura de 288, se a gente dividir ela por dois, a gente pega a Collection View, divide ela por 2, vai dar 144 que é o valor que a gente quer para a largura da nossa célula.

[07:23] Então a gente foi lá, pegou esse valor de 144 e colocou como largura das células. Então, teoricamente, aqui era para ter 144 e essa outra célula 144, aí somando as duas dá o valor da largura total da Collection.

[07:40] Porém, pessoal quando a gente trabalha com a Collection View, por default a Collection traz um espaçamento mínimo entre as células, que é esse espacinho aqui, olha só, essas duas setinhas, esse espaçamento.

[07:52] Então, olha só, se a gente vir aqui no storyboard, a gente consegue visualizar isso. Vamos abrir o storyboard e selecionar nossa collection, estou com a collection selecionada.

[08:02] Se a gente vir aqui nessa régua, ele mostra todos os espaçamentos que tem, que a gente pode setar na nossa collection, e olha, a gente nem mexeu e já veio aqui o espaçamento por células. Olha só, um espaçamento por células de 10. Então, a gente tem que considerar esse 10 na hora que a gente for pegar a largura das nossas células.

[08:26] Então o quê que aconteceu? A gente tem a largura da nossa collection, 288, cada célula tem 144, somando as duas dá 288. Porém, como a gente viu, tem mais os 10, que é o espaçamento entre as células. Então, 288, que é a soma das duas células, mais 10, ultrapassa o valor de 288.

[08:52] Como ultrapassa, ele não consegue renderizar as duas células na mesma linha porque não vai caber, aí o que ele faz? Ele renderiza apenas uma, uma embaixo da outra, então, a gente tem que modificar um pouquinho a nossa lógica. A gente tem que deixar um espacinho entre as células. Então, vamos voltar no nosso View Controller e mexer nisso.

[09:14] Agora, a gente pode criar aqui, vamos criar uma constante, let, largura da célula. Então, a gente vai pegar a collection, como a gente quer acessar a largura, a gente põe ponto bounds e a gente pega a largura e divide ela por 2.

[09:31] Isso a gente já fez, a gente colocou isso aqui embaixo. Então, como a gente já tem uma constante com esse valor, a gente pode pôr a largura da célula e a gente vai subtrair um valor, que a gente vai somar 144 + 144, vai dar 288, menos um valor que a gente vai deixar aqui pra célula se acomodar direitinho. Então, vamos pôr, por exemplo, menos 15. Vamos rodar o app agora pra gente ver como está.

[10:02] Beleza pessoal, agora a gente conseguiu deixar a nossa lógica legal. A gente está pegando a largura da célula, dividindo por 2, atribuindo esse valor na célula e tirando um espacinho para que ela fique confortável aqui na collection.

[10:17] Olha só, agora a largura dela ficou bem melhor. Só que, se a gente reparar, olha nossa imagem, ela está numa posição fixa, ela não está se esticando junto com a célula.

[10:31] O que está acontecendo? A gente não configurou, a gente não prendeu ela aos lados, a gente não colocou nenhuma constraint para isso. Então, vamos voltar aqui no storyboard e já arrumar isso.

[10:41] Vamos vir aqui, selecionar nossa imagem e aplicar as nossas constraints através do nosso popover aqui de constraints. Então, eu vou prender ela à esquerda, à direita, e acima e deixar uma largura. Aí a mesma coisa com as outras labels, vou selecionar a primeira label, prender ela aos lados e acima.

[11:03] A gente pode dar um atualizar e fazer exatamente isso com as demais. Vamos prender 0 à esquerda, 0 à direita e deixar um espacinho entre uma label e outra.

[11:19] Vamos atualizar e vamos fazer isso com a última label. Vamos prender ela 0 à esquerda, 0 à direita e um espacinho, vamos atualizar e ok. Agora a gente já está prendendo a imagem, então, conforme a célula vai se esticando, a imagem também vai.

[11:43] Para terminar pessoal, reparem que aqui a gente tem um espacinho entre a label e o text field, por exemplo. E no nosso app, a gente está com espaço mínimo, está bem grudado um com o outro. A gente pode arrumar esse espaçamento de uma forma bem simples.

[11:59] Olha só, como a gente colocou um Stack View e colocou todos os elementos dentro, facilmente a gente consegue modificar esse espaçamento. Eu vou abrir aqui nosso inspetor com todas as propriedades e vou vir no quarto ícone.

[12:14] Repare que aqui tem uma opção de espaçamento, então, conforme eu vou aumentando o espaçamento aqui, os nossos elementos vão se distanciando um do outro, então é isso que a gente vai fazer.

[12:25] Olha só, eu vou aplicar aqui um espaçamento de três, os elementos já se distanciaram um do outro. Então, teoricamente, já vai ficar mais bonitão nosso app. Vamos rodar de novo para testar?

[12:37] Legal pessoal, agora nosso aplicativo já está bem mais bonitão. A gente conseguiu aumentar o espaçamento, na verdade, diminuir o espaçamento entre as células para aproveitar mais o espaço. E a gente prendeu a imagem na célula, assim quando a célula aumentar ou diminuir de tamanho, a nossa imagem vai estar belezinha.

[12:57] A gente, também, alterou o espaçamento entre os elementos no Stack View, que estava bem colado um no outro, então a gente resolveu esse problema. Bom, todo momento, pessoal, a gente testou o nosso app apenas nessa tela de Pacotes de Viagem, porque a gente deixou o storyboard entry point nela.

[13:14] O legal agora é a gente começar a juntar essas telas para que a gente não precise ficar arrastando esse storyboard entry point de uma tela para outra pra gente poder testar. Então, isso a gente vai ver daqui a pouquinho.