

02

## Auto Layout com UIStackview

### Transcrição

[00:00] E aí pessoal, tudo bem? Bem-vindos a segunda parte do curso de layouts responsivos para iOS. Na primeira parte do curso a gente começou a criação do aplicativo Alura Viagens, então, a gente já listou algumas viagens. E, o principal, a gente estudou o primeiro conceito de responsividade para iOS que foi as constraints.

[00:20] Nessa segunda parte do curso, a gente vai continuar o projeto, porém, implementando novas funcionalidades como, por exemplo, a próxima tela que a gente vai implementar que é a tela de pacotes de viagens.

[00:32] Então, a gente tem vários elementos já conhecidos como labels, text field, e a gente vai aprender outros como, por exemplo, coleções e outros elementos. Então, vamos voltar para o storyboard e continuar a implementação.

[00:46] Então pessoal, de volta aqui no storyboard, a gente precisa continuar a implementação do nosso app. Então, a gente viu que a próxima implementação vai ser a tela de pacotes.

[00:56] Então, pra gente colocar uma nova tela no nosso View Controller, como é que a gente faz? Através do Object Library, a gente tem opção View Controller, então a gente vai clicar aqui e arrastar para dentro do storyboard, repare que aqui a gente tem outra tela.

[01:12] Assim como no primeiro curso, a gente vai continuar seguindo um design. Então, a gente tem já um layout pronto e que a gente vai seguir que, no caso, é essa próxima tela. Vamos começar arrastando alguns objetos, primeiro a gente vai colocar uma Label com o título da tela que é Pacotes, então, vamos procurar aqui por label, vamos clicar e arrastar para cá.

[01:35] A gente já pode trocar o nome alterando a propriedade de texto da Label, então como é que a gente faz isso? Aqui na propriedade text, a gente apaga e coloca Pacotes, vamos arrastar ela um pouquinho mais para o lado.

[01:50] O próximo elemento que a gente vai utilizar é um text field. Então, vamos lá a gente vai vir aqui no Object Library, procurar por text Field.

[01:58] Mesmo esquema, a gente vai clicar e arrastar para o View Controller. Aqui embaixo, a gente tem outra Label, seis pacotes, né? Então como é que a gente faz? Vamos duplicar agora, por exemplo, essa primeira Label, lembra que a gente já aprendeu a fazer isso?

[02:14] A gente pode segurar aqui a tecla option, clicar e arrastar para baixo, ele copia as mesmas formatações das Labels que a gente utiliza. A gente vai trocar o texto aqui para 6 pacotes, 6 pacotes encontrados.

[02:30] Vamos fazer um teste. Assim como a gente fez na primeira parte do curso, em que a gente implementou todo nosso layout na visualização do iPhone SE porque a gente viu que era mais fácil a gente começar a implementação com tamanho de tela menor.

[02:46] Porque assim a gente já define como vai ser a interface do nosso aplicativo na tela menor e, quando a gente for aumentar a tela para outros tamanhos de iPhone, por exemplo, é mais fácil de adaptar porque tem mais espaço. A gente vai continuar utilizando o tamanho do iPhone SE.

[03:02] Se a gente vem aqui, por exemplo, no iPad, repare que o nosso text field ele não está indo até o final do nosso View Controller. Olha só, ele ficou com uma posição fixa na tela, a mesma posição que a gente usou para implementar

no iPhone SE, ou seja, ele não está se adaptando, ele não está se esticando.

[03:24] Aí foi que a gente começou a entender o primeiro conceito de responsividade que são as constraints. A gente usa as constraints para prender os elementos na tela e fazer ele se esticar ou diminuir dinamicamente. Como é que a gente fez isso? Vamos aplicar, então, as constraints nesses elementos pra gente relembrar.

[03:44] Então, a gente vai selecionar aqui primeiro essa label e vai abrir o popover de constraints que é esse popover aqui, olha só. Aí, a gente pode prender a margem desses elementos com relação a tela, então olha só! Eu vou prender ele a esquerda, eu vou prender à direita e prender a cima, e vou dar um "adicionar constraints".

[04:06] Quando a gente adiciona constraints pessoal, lembra que a gente tem essas linhas aqui indicando que a gente colocou as constraints? Então estão todas as linhas das constraints que a gente utilizou.

[04:17] Vamos fazer a mesma coisa com o text field aqui de baixo, a gente vai clicar aqui e vai aplicar as constraints para margem esquerda, direita e superior.

[04:26] Então o quê que a gente está fazendo? A gente está pegando o text field e prendendo a margem dele à esquerda, à direita e acima. Acima em relação a label Pacotes. Então, olha só, vou dar um "adicionar" e beleza. Vamos colocar também constraints na próxima label, mesmo esquema.

[04:44] A gente vai abrir o popover, colocar na margem esquerda, direita e acima. Olha só, a gente já colocou constraints em todos os elementos, ou seja, a gente prendeu eles em relação a tela para quando a gente ver esses elementos em outros tamanhos. Olha só, vamos vir aqui no iPad de novo.

[05:03] Repare que agora, o text field, ele está ocupando todo o espaço porque a gente configurou as constraints para ficar com uma margem à esquerda e à direita, ou seja, quando a tela aumenta o text field também aumenta. Isso foi o que a gente aprendeu no 1º curso, rapidamente passando, aplicar constraints nos elementos.

[05:24] Mas, olha só pessoal, e se a gente por acaso, nós como desenvolvedores, a gente precisa ter uma face manutenção das nossas telas.

[05:34] Imagina que, por exemplo, a gente está trabalhando numa Agência de Viagem e, de repente, a nossa equipe é informada de que a gente precisa mudar a interface do nosso aplicativo, por exemplo, por causa de uma promoção ou um Black Friday, por exemplo, e a gente prendeu todos os nossos elementos utilizando constraints.

[05:50] A gente prendeu as labels, o text field em relação a label, a gente prendeu a label 6 pacotes em relação ao text field. Então está um elemento preso em relação ao outro.

[06:01] Então vamos supor que, por exemplo, a nossa equipe fale que a gente precisa colocar uma label de pacotes promocionais, como é que a gente faria isso? Então, olha só, embaixo dessa label Pacotes, a gente precisa colocar uma nova label, só que essa label está presa aqui.

[06:19] A gente aplicou constraints e ela está presa em relação ao text field, então o que a gente faz aqui? Olha só, a gente pode clicar nessa constraint de top, do text field, e apagar.

[06:34] Aí a gente vai colocar uma nova label embaixo da label Pacotes. Então a gente vai selecionar o text field e a label "6 pacotes encontrados" e vamos dar um espaço aqui.

[06:44] Agora, a gente vai colocar uma nova label, então, vamos procurar aqui no Object Library por label e vamos arrastar aqui. Então, vamos supor que a nossa tela vai se chamar aqui "Pacotes Promocionais". Vou arrastar essa label até o final, igual a de cima, só que repare que storyboard, ele está apontando um erro aqui.

[07:09] Lembra que quando a gente não satisfaz todas as condições das constraints, ele aponta um erro pra gente? Isso acontece porque a gente apagou a constraint top do nosso text field, então a gente vai ter que recolocar essa constraint, ou seja, a gente vai ter que prender o topo do nosso text field em relação a nova label que a gente colocou, que é a label "Promocionais".

[07:31] Mas, antes, a gente tem que prender essa label que a gente colocou. A gente vai fazer o mesmo esquema, vamos prender ela à esquerda, à direita e acima.

[07:37] Então, olha só, vou prender à esquerda, à direita e acima, e vou dar um "adicionar". Essa label já está com as constraints necessárias. Agora a gente vai selecionar o text field e vai colocar a constraint de top, de novo, em relação à label.

[07:55] Então, beleza pessoal! A gente colocou todas as constraints que faltava, a gente já prendeu e colocou a constraint de top do text field em relação a nova label que a gente colocou. É muito comum, pessoal, quando a gente trabalha com desenvolvimento de aplicativo, surgir a necessidade de mudar a interface do aplicativo.

[08:12] Por exemplo, a gente está desenvolvendo um aplicativo de viagens, vamos supor que a diretoria da empresa amanhã peça para que a gente mude o aplicativo porque vai lançar uma nova promoção ou um Black Friday e a gente vai ter que mudar toda a interface do aplicativo e deixar ela chamativa e trocar os elementos de lugar.

[08:30] Isso é comum, a gente tem que seguir essas especificações, porém, quando a gente trabalha com constraints, é difícil de ficar dando essa manutenção de apagar elemento, colocar as constraints de novo e ver quais constraints estão faltando. A gente colocou apenas uma label aqui e já viu que deu um trabalhinho.

[08:49] A gente teve que apagar, colocar os elementos um pouco mais para baixo, recolocar as constraints do text field, então é um pouco trabalhoso isso aí. Por isso, a partir de agora, a gente vai começar a estudar um outro tipo de auto layout no iOS.

[09:04] A gente vai começar a empilhar os elementos, a gente vai empilhar os elementos para que eles ocupem o tamanho que a gente queira na tela. Para isso a gente vai utilizar um cara aqui no Object Library que é Stack View, que é o cara que empilha os elementos pra gente.

[09:21] Então, quando a gente procura por Stack View, ou seja, a gente vai empilhar os nossos elementos, têm dois tipos de empilhamento que a gente pode usar: ou o horizontal, que é um ao lado do outro, ou o vertical, olha o desenho, um embaixo do outro.

[09:36] Como é que a gente sabe quais desses dois a gente vai utilizar? A gente sabe analisando o design. Então, olha só, a gente tem aqui a tela de Pacotes.

[09:46] Então vamos dar uma olhada, repare que aqui a gente tem uma label, um text field, uma outra label e uma coleção aqui de outros elementos. Repare que tem um elemento embaixo do outro, se tem um elemento embaixo do outro, é uma boa a gente utilizar, por exemplo, o vertical Stack View.

[10:04] Então vamos fazer o seguinte, vamos apagar esses elementos que a gente colocou aqui na tela e vamos começar a trabalhar a partir de agora com o Stack View que é uma outra forma de trabalhar com auto layout no iOS. [10:16] Então vamos lá! Eu vou pegar aqui o Stack View e arrastar para dentro do View Controller que a gente está trabalhando.

[10:21] Vamos posicionar ele de uma forma que ele ocupe toda a tela, então eu vou puxar ele até a margem direita, a margem esquerda, e arrastar ele para baixo, vou puxar ele até aqui embaixo, agora a gente já colocou ele na nossa tela.

[10:42] Vamos fazer o seguinte, vamos aplicar constraint no Stack View porque a gente vai ter que utilizar alguns elementos dentro do Stack View com constraint, então, a gente precisa que o elemento pare, ele esteja com constraint

também.

[10:56] Vamos voltar aqui no nosso design e vamos dar uma olhada nos elementos que a gente precisa arrastar. Então olha só, a gente vai começar com a label Pacotes. Então, vamos de novo aqui no Object Library e procurar por label, vamos arrastar aqui para dentro da tela.

[11:10] O próximo elemento é um text field, então vamos vir aqui "text field" e vamos pôr novamente nossa label "6 pacotes encontrados". Então, olha só, eu vou procurar aqui por label, vou arrastar para a tela.

[11:27] Vamos começar a nomear esses elementos. Então, primeiro, a gente vai chamar de "Pacotes" e, a label aqui de baixo, a gente vai chamar de "6 pacotes encontrados". Agora pessoal, repare que o text field ele ficou bem grandão aqui na tela, ele está preenchendo basicamente todo o espaço do nosso View Controller.

[11:55] Por que isso acontece? Quando a gente coloca como Stack View, ele tem uma distribuição que a gente pode setar e já vem uma distribuição setada para ele que é o preenchimento.

[12:06] Ou seja, todos os elementos que a gente coloca nele, ele ocupa todo espaço, então, o nosso text field, ele ficou grandão porque ele está puxando as labels para baixo e para cima, ocupando, assim, todo o espaço do Stack View.

[12:21] Então, a distribuição dele é essa daqui "Fill", então ele vem já com essa distribuição e vai preencher todo o espaço necessário. Como é que a gente resolve isso? A gente pode colocar, por exemplo, uma constraint de altura para o nosso text field.

[12:38] Por padrão, o text field tem uma altura de 30, então o que a gente pode fazer? A gente pode vir aqui e colocar uma altura de 30. Repara que agora ele está apontando um erro aqui porque ele precisa preencher todo o espaço do text field.

[12:58] Então, o quê que falta a gente colocar aqui na tela? Falta a gente colocar o próximo elemento que é esse aqui. Então pessoal, o próximo elemento que a gente vai utilizar é uma Collection View.

[13:09] Então, a gente vai vir aqui no Object Library e procurar por Collection, é uma coleção de pacotes que a gente vai implementar. Então a gente vai escolher esse carinha aqui, Collection View, aí a gente vai arrastar ele para dentro do Stack View.

[13:25] Então, a gente vai clicar e arrastar para cá. Repara que agora ele está preenchendo todo o espaço do Stack View, aí ele parou de apitar o erro que estava dando aqui no nosso View Controller.

[13:39] Vamos testar nossa implementação para ver como é que está ficando. Como a gente ainda não vai trabalhar com várias telas no nosso aplicativo, a gente vai testar de uma forma simples, a gente vai clicar aqui no Storyboard Entry Point e arrastar para o View Controller que a gente acabou de criar.

[13:56] Então, vamos começar aqui testando, por exemplo, para o iPhone 6. Vamos dar aqui um run. Então, olha só pessoal, todos os elementos que a gente colocou está belezinha, eles estão empilhados a partir de agora. Repare que, olha só, tem um embaixo do outro sem nenhuma margem, eles estão meio que colados assim.

[14:16] Vamos fazer um outro teste. A gente viu que quando a gente precisa aumentar ou esticar os nossos elementos, a gente utilizava as constraints prendendo as bordas, a gente prendia a margem esquerda e a margem direita.

[14:28] Nesse caso, a gente só colocou os elementos dentro do Stack View. Vamos fazer outro teste rodando o nosso app, por exemplo no iPad Air 2. Vamos dar aqui um run e vamos ver o que vai acontecer.

[14:39] Então, olha só pessoal, reparem que o tex field ele está preenchendo todo o espaço do nosso View Controller sem a gente precisar prender as bordas dele, as margens dele, porque, como a gente viu o Stack View, ele usa um preenchimento, ou seja, quando aumenta o tamanho da tela, o text field automaticamente preenche todo esse espaço.

[15:01] Então, a gente não precisa ficar colocando as constraints de margem porque quando ele aumenta, os elementos de dentro também aumentam.

[15:08] Então, olha só, aqui a gente tem o text field preenchendo todo o espaço do View Controller. Isso é legal porque a gente evita de trabalhar com constraints, então se a gente precisar apagar ou colocar elementos, a gente não precisa se preocupar em quais constraints a gente precisa recolocar.

[15:26] A gente colocou agora uma Collection View e a gente ainda não implementou ela, então, o próximo passo é a gente começar a implementação da nossa Collection. Isso a gente vai ver daqui a pouquinho.