

03

## Chamando a tela de confirmação de pagamento

### Transcrição

[00:00] Agora, pessoal, a gente precisa chamar essa tela que a gente acabou de criar aqui no storyboard. Como é que a gente faz para chamar?

[00:06] A gente precisa criar um arquivo para gerenciar essa tela. Até então, a gente apenas colocou um objeto aqui do View Controller para o storyboard e os elementos dentro, mas não tem nenhum arquivo gerenciando essa tela. Então, é isso que a gente precisa criar agora.

[00:20] Então, vamos voltar aqui na hierarquia das nossas pastas e vamos criar um novo grupinho, uma nova pastinha para organizar esse arquivo. Então, vou vir aqui em "New Group" e digitar, por exemplo, "ConfirmacaoPagamento". Vou arrastar ela para baixo do Detalhes e vou criar uma nova pastinha Controller.

[00:41] "Controller" e dentro dela, sim, a gente pode criar o nosso View Controller. Então a gente vai clicar aqui com botão direito, "New File", dar um "Next", aqui em subclasse a gente escolhe "UIViewController".

[00:53] Então, como é que a gente vai chamar? "ConfirmacaoPagamentoViewController". Linguagem Swift, vamos dar um "Next" e um "Create".

[01:03] Legal, agora a gente já tem o arquivo que a gente pode usar naquele View Controller. Como é que a gente vai fazer agora? Vamos abrir o storyboard, clicando em cima dele. Aqui em Class, a gente coloca o arquivo que a gente acabou de criar. Então, vamos colocar aqui "ConfirmacaoPagamentoViewController".

[01:10] Dei um "Enter", beleza, o arquivo já está gerenciando esse novo View Controller.

[01:11] Então, de volta o projeto, no simulador quando a gente escolhe um pacote, o que a gente quer que aconteça quando a gente clica em "finalizar compra"? A gente quer chamar essa tela que a gente acabou de criar.

[01:37] Como é que a gente faz para chamar? A gente precisa criar uma Action nesse botão. Então, olha só, vamos clicar aqui, vamos dividir a tela e vamos criar uma Action. Como é que eu faço? Eu vou clicar aqui com botão, com o "Ctrl" segurado, clico e arrasto para o View Controller.

[01:58] Action, aqui vai ser "UIButton" e o nome dele vai ser "botaoFinalizarCompra" e vou daqui um "Connect". Legal, a gente já tem um botãozinho, então. Quando o usuário clicar nesse botão, vai disparar o que a gente colocar dentro desse método, dentro dessa ação.

[02:17] Então, vamos voltar lá e começar a implementar isso daí. Então eu vou abrir aqui o View Controller de Detalhe e vou vir no método que a gente acabou de criar.

[02:26] Primeiro passo, pessoal, é a gente pegar o nosso storyboard. Então a gente tem que criar uma instância dele para a gente poder acessar os nossos View Controllers que estão lá dentro. Então, vamos lá.

[02:37] "let storyboard = UIStoryboard" e a gente inicializa ele com o nome. Qual é o nome dele? O nome dele é "Main", olha aqui. Então a gente vai pôr "Main, bundle: nil", porque ele está nessa estrutura mesmo.

[02:56] Agora a gente vai criar o nosso Controller. Então, "let controller" é igual o storyboard que a gente criou, ponto, a gente vai iniciar o View Controller com o identificador. Qual identificador? A gente não colocou identificador ainda, a gente vai pôr agora.

[03:13] Então, eu vou clicar aqui no storyboard novamente, vou selecionar o View Controller que a gente acabou de criar e vamos colocar um identificador.

[03:23] Aqui em "Storyboard ID", a gente pode pôr qualquer identificador. Então eu vou pôr, por exemplo, "confirmacaoPagamento". Essa mesma string que eu coloquei aqui, é o que a gente vai utilizar lá, então vamos voltar lá e utilizar esse identificador.

[03:40] Então, está vendo que ele espera aqui uma string? Vamos passar esse identificador. "Command + V" e a gente pode falar que ele é do tipo "ConfirmacaoPagamentoViewController".

[03:54] Agora que a gente já acessou o storyboard e já acessou View Controller, a gente consegue apresentar essa tela na tela. Então, como é que a gente faz isso?

[04:05] "self.present" e a gente vai passar aqui o View Controller, que é o Controller que a gente criou. Animado a gente passa verdadeiro e na closure de completion a gente não vai querer fazer nada. Ou seja, quando a gente chamar a tela, a gente não vai querer fazer nada. Então a gente pode passar aqui "nil".

[04:23] Vamos testar e ver se isso daí está funcionando? Vamos rodar o app. Legal, galera, vamos testar. Vamos vir em "Pacote" e selecionar qualquer pacote aqui, por exemplo, "Salvador". Cliquei, apareceu a tela de detalhe e eu vou clicar agora em "finalizar".

[04:38] Cliquei em "finalizar", apareceu a tela que a gente criou. Vamos comprar a tela que a gente tem até o momento com o design. Olha só, vamos abrir aqui o design. O header, beleza, a gente já tem. A parte de baixo não está aparecendo a imagem.

[04:55] Repare que aqui, o nosso Porto de Galinhas, o nosso título e as outras Labels também, ela está bem grudada uma com a outra, o espaçamento dela está bem grudado e ele está grudado com a margem, também.

[05:06] Então, a margem a gente tem que deixar um espacinho de cada lado, para ele ficar melhor. Então, vamos começar a mexer nisso agora. Vamos voltar aqui no storyboard e começar a mexer nesses itens.

[05:18] Primeiro passo, pessoal, a gente vai ter que colocar o espaçamento entre os elementos que tem aqui dentro. Então eu vou selecionar o Stack e vou abrir a propriedade Spacing dele, que a gente já utilizou. Vamos pôr, por exemplo, 5, é um espaçamento legal.

[05:32] Só que, olha só, a gente precisa agora colocar o espaçamento na margem. Como é que a gente faz isso? A gente pode selecionar as Constraints aqui do Stack. Então eu venho aqui na reguinha, está vendo que ele mostra quais são as Constraints que tem nele? Então, vou ter que mexer nessa Constraint e nessa outra Constraint aqui.

[05:50] Então, primeiro vamos mexer nessa daqui. Cliquei, vou dar dois cliques na Constraint de leading. Dei dois cliques, aqui onde está a Constant, a gente vai alterar o valor dela. Vamos pôr, por exemplo, 10. Vamos fazer a mesma coisa com a de trailing. Selecionei, 10.

[06:08] Só que tem um probleminha, a gente já viu isso antes. A margem que a gente colocou, ela está colocando em todos os elementos dentro do Stack, inclusive no header.

[06:19] Só que o header a gente não quer que tenha essa bordinha, o header tem que ocupar todo espaço. Então a gente vai ter que tirar o header de dentro do View Controller, de dentro do Stack View.

[06:30] Primeiro passo é gente selecionar o Stack e apagar a Constraint de Top, porque o Stack View não vai estar mais preso em relação ao View Controller, ele vai estar preso em relação ao header. Então vamos clicar aqui e dar um "Delete". Ele está apontando erro, não tem problema.

[06:47] Agora a gente pega a View, arrasta para cima, fora do Stack View. Olha só, arrastei ela para cima e agora posicionei ela novamente lá em cima.

[06:58] Então, vou clicar e colocar ela aqui no topo. Vou arrastar as margens dela e agora eu prendo ela em todos os cantos aqui.

[07:10] Agora eu dou dois cliques em cima do Stack View e arrasto ele para baixo. Ele vai ficar um pouquinho mais para baixo aqui do header. Vou arrastar, agora eu recoloco apenas a Constraint de Top.

[07:24] Então vou recolocar essa Constraint. Vou dar um "Add" e vou dar um Update Frame aqui. Ótimo, a gente já colocou o espaçamento mais bacana aqui. Agora vamos centralizar essas Labels no meio aqui do header.

[07:40] Está vendo que ele aponta uma linhazinha quando a gente chega no centro? Vamos deixar aqui.

[07:45] Agora, pessoal, a gente precisa exibir de verdade o pacote que o cara comprou, porque até então está tudo fixo. Então, para exibir, a gente vai ter que criar os Outlets. Então, vamos começar criando aqui os Outlets.

[07:59] O primeiro vai ser a imagem. Então a gente vem aqui, vamos criar os Outlets da imagem. Segura a tecla "Ctrl" e puxa aqui para dentro. Então, olha só, "ImagenPacoteViagem", primeiro Outlet.

[08:14] Segundo aqui é o título, então vamos selecionar, "Ctrl", puxo para cá. Então, "labelTituloPacoteViagem". O próximo vai ser o hotel, então vou clicar e arrastar. Então, "labelHotelPacoteViagem".

[08:35] O próximo vai ser a quantidade de dias, ou seja, pode ser a data, quando que o usuário vai poder viajar. Então seleciono e arrasto pra cá. "labelDataPacoteViagem".

[08:51] Agora falta a gente criar mais dois Outlets. Vou criar esse, "labelQuantidadePessoas".

[09:03] E o último, label descrição. O que vai ter no seu pacote, você comprou e vai ter o quê? Café da manhã, hotel, voo, o que mais? Essa Label vai dizer. Então, "labelDescricaoPacoteViagem".

[09:23] Agora a gente já tem todas as Labels aqui no nosso View Controller. Porém, pessoal, agora a gente precisa passar o objeto, ou seja, a viagem que a gente selecionou, que a gente vai comprar, por parâmetro.

[09:37] Então a gente tem que vir aqui em ConfirmacaoPagamento e criar uma variável do tipo Pacote. Então, vamos fazer isso. Vamos vir aqui, "var pacoteComprado", ele vai ser do tipo PacoteViagem e a gente inicializa ele sem nada, porque a gente vai passar ele por parâmetro.

[09:57] De volta aqui no Detalhes, a gente precisa passar esse objeto. Como é que a gente faz? A gente pega aqui "controller.pacoteComprado =", qual é o nome dessa variável que a gente usa aqui? pacoteSelecionado. Então " = pacoteSelecionado".

[10:19] Vamos ver se a gente já está conseguindo passar exatamente o objeto? Então, no método viewDidLoad, eu vou colocar aqui um "if let pacote = pacoteComprado".

[10:29] Vamos desempacotar esse objeto de dentro dessa variável aqui. Então eu vou pôr um print "pacote.", por exemplo, "viagem.titulo".

[10:43] Então vamos rodar e ver se está aparecendo o nome do pacote que a gente comprou aqui. Então, vamos lá. Vamos testar? Então a gente vai vir aqui em "Pacote", vamos selecionar, por exemplo, "Salvador" e dar um "finalizar".

[10:57] Vamos limpar aqui o console, vamos dar um "finalizar". Legal, então a gente já está conseguindo pegar o objeto certinho que a gente está comprando. Agora só falta a gente setar ele nos nossos Outlets.