

Filtrando com NSPredicate

Transcrição

[00:00] E aí galera, beleza? De volta ao projeto, a gente já consegue alternar as telas da nossa aplicação. Antes, a gente tinha que colocar o storyboard entry point, essa setinha, no View Controller, depois em outro e, assim, ir testando um de cada vez.

[00:14] Então, na aula passada, a gente colocou um Tab Bar Controller, que gerencia toda essa navegação pra gente. Então, olha só, agora no simulador a gente já consegue ir de uma tela para outra.

[00:24] Só que a gente estava conversando que a tela Pacotes, a interface dela já está ok. A gente já formatou, já colocou as cores e tudo mais, só que tem um text field aqui ainda que a gente não usa para nada, a gente colocou ele para simular um search, só que ele ainda não funciona.

[00:39] Então, olha só, eu digito por exemplo, algum caractere, por exemplo Fortaleza, ou o início dele e ele não está buscando, então, a gente ainda não implementou isso.

[00:48] Então, agora, a gente vai começar a implementar o filtro nessa tela, então a gente vai filtrar. Quando a gente utiliza um aplicativo de vendas de qualquer produto ou serviço, é muito comum ter um search nesse aplicativo porque é muito custoso para o usuário ficar pesquisando um item por vez, até ele encontrar o que ele precisa.

[01:06] E no nosso caso é a mesma coisa, o usuário, se ele já sabe para onde ele quer viajar, ele não vai ficar passando viagem por viagem até ele achar a que ele quer.

[01:15] Então, o legal é a gente implementar esse search e o usuário digitar o comezinho da palavra e ele conseguir pesquisar e encontrar rapidamente a viagem. Então vamos voltar aqui no storyboard e começar a mexer nisso daí.

[01:29] Quando a gente implementou a tela de pacotes, a gente tinha colocado um text field aqui para simular um search, porém no iOS, a gente já tem um componente nativo que faz isso para gente, um search mesmo, então vamos utilizar ele.

[01:44] Vamos aqui no Object Library procurar por search, então a gente vai utilizar esse primeiro carinha aqui, search bar. Então, a gente vai clicar no text field, apertar o delete para apagar. Repare que quando a gente apagou, o próprio Stack View gerenciou o espaço pra gente, ele não apontou nenhum erro, nem nada, diferente se a gente estivesse utilizando as constraints.

[02:08] Um item estaria preso ao outro, iria dar algum probleminha e a gente teria que resolver. No Stack View não, ele se auto gerencia, então, a gente vai clicar aqui no search e arrastar para o mesmo lugar onde estava o text field.

[02:21] Reparem que quando a gente colocou o search aqui no View Controller, ele não apareceu. Olha só, a altura dele está 0, então o Stack View prendeu o nosso search, então o que a gente tem que fazer? A gente precisa setar uma constraint de altura, vamos pôr aqui, por exemplo, 35. Olha só, ele está lá bonitão.

[02:42] Com o search bar já no nosso View Controller, vamos aproveitar e criar um outlet porque a gente vai precisar mexer nele, então a gente vai clicar aqui, segurar o Ctrl, e arrastar para dentro do nosso View Controller.

[02:56] Então, a gente vai colocar aqui, por exemplo, pesquisar viagens. Então esse vai ser o nome do nosso search: Pesquisar Viagens. Vamos dar um connect e beleza.

[03:08] Feito isso pessoal, a gente vai começar a mexer no nosso View Controller novamente, então vamos clicar em cima dele. Agora a gente já tem um outlet dele, a gente já pode implementar os seus métodos de delegate.

[03:20] Assim como a table e a collection, quando a gente utilizou, a gente implementou seus protocolos de delegate e data source pra gente conseguir utilizar os seus métodos.

[03:29] O search é a mesma coisa, a gente vai utilizar um método que toda vez que o usuário digitar alguma coisa aqui no search, a gente vai conseguir capturar esses caracteres pra gente conseguir fazer a filtragem.

[03:40] Então como é que a gente vai fazer isso? A gente vai colocar uma vírgula e a gente vai implementar esse protocolo de delegate, então, olha só, UI Search Bar Delegate. Sempre que a gente utiliza o método de delegate, a gente tem que setar ele, a gente tem que falar que o nosso View Controller vai implementá-lo.

[03:58] Então, como é que a gente faz isso? A gente vai pegar aqui o Outlet que a gente criou, ponto, delegate, igual a selfie. Feito isso, a gente já pode utilizar o método de delegate do search. Então, lá embaixo, vamos vir aqui embaixo e já implementar esse método.

[04:18] Olha só, eu vou começar a digitar aqui search e ele já vai me trazer vários métodos que a gente pode implementar. O método que a gente precisa é um que toda vez que o usuário digite alguma coisa, a gente consiga capturar.

[04:30] Então, vamos pegar esse primeiro método aqui, Text Did Change, esse é o método que a gente vai utilizar. Vamos implementar ele.

[04:39] Repare que ele dá aqui, como parâmetro, um search text que é exatamente o que o usuário digitou. Vamos fazer um teste, vamos apagar isso aqui e vamos colocar um print pra gente ver no console se está funcionando realmente.

[04:53] Então, olha só, vou pôr um print e vou passar esse cara aqui, vamos apagar e passar ele. Vamos rodar o app e fazer um teste agora.

[05:05] Então vamos testar pessoal, vamos vir aqui na tela Pacotes e a gente vai digitar algum caracter aqui no nosso search bar. Então, olha só, vou digitar, por exemplo, o início da palavra Fortaleza. Repare que o que a gente digitou, olha só, está aparecendo aqui no console, ou seja, a gente está capturando já o que o usuário digita no search.

[05:26] Vamos apagar e digitar outro destino, por exemplo, Rio de Janeiro. Todos os caracteres que a gente vai digitando vão aparecendo aqui no nosso console. Então o nosso método e a implementação estão ok.

[05:40] Agora a gente precisa pensar uma forma da gente filtrar essas viagens. Lembra que a gente tem uma lista com todas essas viagens, uma rede de viagens? A gente vai ter que utilizar ele, a gente vai pegar o que o usuário vai digitar e ver se contém em algum objeto de dentro da nossa lista.

[05:59] Quando a gente trabalha com essa filtragem offline, tem uma classe do Swift que a gente consegue utilizar para filtrar os objetos em memória que é a NSPredicate, então a gente vai utilizar ela agora. Vamos apagar aqui esse print, que a gente só colocou para teste, e vamos começar a implementar essa classe.

[06:16] Então vamos lá, let, filtro lista viagem, a gente vai falar que ele é um NSPredicate que é a classe que a gente vai utilizar para fazer essa consulta. Então vamos inicializar ela com um formato e um argumento.

[06:33] Nesse formato, a gente tem que ver o nosso objeto, lembrando que a gente está utilizando o objeto viagem, que a gente criou na primeira parte do curso. Esse objeto, ele tem vários atributos como, por exemplo, o título, a quantidade de dias, o preço, o caminho da imagem.

[06:49] Então, olha só, na nossa coleção aqui, a gente tem todos esses atributos, o título, a quantidade de dias, o preço, a imagem.

[06:59] Quando o usuário digitar alguma coisa no search, qual seria o melhor atributo pra gente utilizar como base para fazer o filtro? Eu digitei Rio de Janeiro, então, uma boa seria a gente utilizar o título da viagem e não o preço e nem a quantidade de dias.

[07:16] O mais comum é pelo título mesmo, então a gente vai utilizar esse atributo. Então, a gente vai vir aqui no formato, como ele está esperando uma string, vamos abrir uma string, vamos digitar título. Aí, a gente vai ver se o título contém em algum objeto do nosso Array.

[07:34] Então, a gente vai digitar a palavra chave aqui, contains, e a gente vai concatenar com o que o usuário digitar. Então, eu vou ver se o que o usuário digitar contém no título de algum objeto de dentro da nossa lista. Aqui, a gente pode passar o que o usuário digitou, então o search text que é esse atributo aqui, que é passado por parâmetro.

[07:58] Feito isso, a gente já tem uma estrutura pra gente fazer a filtragem. Agora, pensa só, quando o usuário digitar aqui o nome de um destino, a gente vai ter que verificar no nosso Array, com todas as viagens, se contém alguma viagem com esse título.

[08:16] Só que a gente tem que guardar isso em algum lugar, então, pode ser que o usuário digite alguma palavra que tenha mais de uma viagem, então, o ideal seria a gente guardar isso dentro de outra lista, dentro de outro Array, então vamos fazer isso.

[08:31] Vamos criar aqui outro Array, let, lista filtrada, ele vai ser um array de viagens. Agora, a gente vai pegar o nosso array com todas as viagens e aplicar esse filtro.

[08:49] Então, eu vou abrir aqui um parênteses e pegar o nosso array lista viagens, eu vou transformar ele no NSArray que é uma classe do Objective-C pra gente utilizar o predicate.

[09:02] Então, NSArray, ponto, aí ele vai ter essa propriedade filtered, que ele espera um predicate que é o que a gente criou ali em cima, então, vamos utilizar ele.

[09:15] Olha só, eu vou passar aqui o filtro lista viagem e, como eu tipei o nosso array para array mesmo e não NSArray, eu vou ter que fazer o casting novamente. Então, olha só, as Array, a gente já tem uma lista que a gente está filtrando o que o usuário digitou.

[09:41] O que a gente precisa fazer agora é substituir a nossa lista atual com a lista com resultados da filtragem. Então, olha só, vou pegar aqui a lista viagens e vou falar que ela é igual a lista filtrada. Feito isso pessoal, vamos fazer um reload na Collection pra gente conseguir testar, então, a gente vai pegar a nossa coleção, coleção de pacotes, ponto, reload data.

[10:05] Esse reload, a gente utiliza para ela recarregar todos os dados, então vamos fazer isso daqui. Como a nossa lista viagens, a gente declarou como let para não alterar o seu valor, a gente precisa trocar aqui para var, ao invés de let, vamos pôr var porque a gente está substituindo o valor dela. Vamos testar o app novamente.

[10:27] Então, a gente vai vir aqui na tela de Pacotes e agora a gente vai digitar o nome de um destino que a gente queira fazer o filtro. Então, vamos digitar, por exemplo, Fortaleza, vou clicar aqui no search e digitar Fortaleza.

[10:41] Olha que legal pessoal, o que a gente digitou, a gente já está conseguindo filtrar através da expressão que a gente utilizou aqui no nosso predicate, isso já tá ok. Só que, repara que, se a gente apagar a string que a gente digitou aqui no search, a nossa collection, não está sendo carregada novamente.

[11:00] Então, olha só, vamos apagar aqui, apaguei a palavra Fortaleza e continua uma coleção vazia. O legal seria se a gente apagasse e voltasse com todas as viagens.

[11:14] Só que a gente tem um problema, olha só, aqui a gente está pegando a lista com todas as viagens e está igualando ela à lista filtrada, ou seja, a gente está perdendo as outras viagens.

[11:27] Então, o legal seria a gente ter um Array fixo com todas as viagens, que a gente não consiga alterar esse valor, e um outro array que a gente consiga fazer a filtragem, que a gente consiga tirar e colocar as viagens de acordo com que o usuário digitar.

[11:43] Assim, a gente não perde as viagens porque aqui a gente está perdendo, a gente está falando que a lista viagens, que a lista com todas as viagens é igual a lista filtrada, então isso não é muito legal.

[11:54] Vamos fazer o seguinte, vamos vir aqui onde a gente declarou esse array e vamos mudar aqui para let, e vamos digitar aqui "lista com todas viagens". Esse array a gente não vai mexer no valor dele, ele sempre vai ser o array que vai conter todas as viagens.

[12:13] E aqui a gente vai declarar uma lista mutável, que a gente vai tirar e colocar os objetos de acordo com o que o usuário filtrar, então, lista viagens, ela vai ser um array de viagem e a gente vai inicializar ela vazio.

[12:30] Aqui no View Did Load, a gente pode fazer a primeira cópia que quando a lista for carregada a primeira vez, ela vai ter que ter todas as viagens.

[12:39] Então, olha só, a gente vai fazer o seguinte, lista viagens é igual a lista com todas as viagens. Vamos fazer um teste? Vamos rodar o app para ver se está funcionando.

[12:48] Então pessoal, quando a gente inicializa o nosso aplicativo e a gente vem para a tela de pacotes, ele está pegando as viagens certinho, só que a gente ainda está com o mesmo problema.

[12:58] Olha só, vou digitar Fortaleza e apagar e a Collection não está recarregando com todas as viagens. Então, a gente vai ter que fazer uma verificação aqui, olha só, a gente vai ter que verificar se contém algum texto dentro do search.

[13:12] Se tiver, aí beleza, a gente faz o filtro. Se não, a gente pega todas as viagens, coloca no array e reloada a collection de novo.

[13:21] Vamos fazer um if. If, o search text for diferente de vazio, ele vai entrar nessa lógica que a gente vai colocar aqui, que é o filtro.

[13:37] Se ele não for, a gente vai pegar o nosso array, que a gente usa para renderizar na collection, lista viagens, e vai falar que ele é igual a lista com todas as viagens. Com isso, a gente consegue resolver o nosso problema. Vamos rodar de novo o app para conferir?

[13:54] Então pessoal, vamos abrir aqui a tela de pacotes pra gente ver se a nossa implementação está funcionando. Vamos vir aqui no search e digitar o título de alguma viagem, por exemplo, Fortaleza.

[14:04] Então, olha só, digitei, ele filtrou, só que quando eu apago o problema que a gente tinha era que a collection não atualizava com todas as viagens novamente.

[14:12] Então, olha só, agora a gente apagou e já está funcionando, então, a gente já está conseguindo filtrar através do search e quando não tiver nenhum valor, ele mostra todos os pacotes encontrados.

[14:24] A gente ainda tem um problema, pessoal, que é essa label, ela está com um valor fixo, 6 Pacotes. Isso a gente deixou porque quando a gente estava montando o layout, a gente só queria copiar a mesma interface, porém, agora a gente já pode deixar o valor dessa label dinâmica. Então, vamos fazer isso.

[14:41] De volta ao storyboard, a gente precisa criar um outlet da label pra gente conseguir alterar o seu valor. Então, a gente vai clicar em cima da label e vamos abrir o nosso View Controller.

[14:51] Então vamos lá, vamos segurar a tecla Ctrl e arrastar pra gente criar o outlet, então, vou segurar aqui, Ctrl e vou puxar. A gente vai chamar de "Label contador de pacotes", beleza, a gente já tem um outlet.

[15:10] Agora, a gente pode voltar no View Controller e implementar um método que a gente consiga saber quantos itens têm dentro do array pra gente colocar como o valor para essa label.

[15:19] Então, olha só, vou voltar aqui no View Controller e a gente pode criar um método, uma função, "Atualiza contador da label". Eu não vou passar nada como parâmetro e vou retornar uma string, que é o texto que a gente vai utilizar para setar na nossa label, então, eu vou dar um return.

[15:40] A gente está utilizando a lista viagens, então, vamos fazer aqui uma verificação. Olha só, se a lista viagens tiver apenas um item dentro, a gente vai retornar um pacote encontrado, senão a gente vai retornar a quantidade de itens que tem no Array e a string Pacotes Encontrados.

[16:06] Então, vamos concatenar aqui, a gente vai chamar a nossa lista viagens, ponto, count, pacotes encontrados, Então, a gente já tratou, se tiver apenas um, a label mostra pacote, se tiver mais que um, pacotes.

[16:28] A gente precisa chamar essa label, essa função aqui, pra gente setar o valor na label, que é onde a gente usa para filtrar. Então, vamos lá, self, ponto Label Contador Pacotes.text, e a gente chama a nossa função self.atualiza Contador Label. Vamos testar pessoal? Vamos rodar o app novamente.

[16:54] Então, vamos abrir aqui a tela Pacotes e, olha só, repare que ainda continua 6 Pactoes. Isso aconteceu porque a gente não chamou, a gente não setou o texto da label quando inicializa o View Controller.

[17:08] Então, a gente tem que colocar essa mesma linha aqui no View Did Load. Vamos rodar de novo o app para testar? Então vamos lá.

[17:18] Vamos abrir aqui a tela Pacotes. Então, olha só, agora sim, beleza, já está mostrando 5 pacotes encontrados. Já está mostrando a quantidade de itens que tem no nosso array, só que vamos fazer um filtro aqui para ver se ele está mudando mesmo? Então, olha só, Rio, aqui já está mostrando 1 pacote encontrado.

[17:37] Então, a nossa lógica que a gente colocou no nosso método já está funcionando. Quando a gente apaga, ele atualiza novamente para a quantidade de itens que tem no nosso array. Então, com isso a gente fechou, de verdade, a implementação da tela de pacotes.