

Removendo transações

Transcrição

[00:00] Atualmente, o nosso usuário quando ele entra aqui na nossa app ele consegue adicionar transações, seja de receita ou despesa. Então ele consegue adicionar uma transação e também ele consegue alterar. A gente viu que a gente está permitindo com que ele faça isso.

[00:12] Só que se de repente ele quiser testar a nossa App, e adicionar aqui uma transação qualquer só para ver como é que ela funciona, mas ele só colocou apenas pra teste e agora ele que remover, será que ele consegue fazer isso? Vamos testar.

[00:22] Se a gente pressionar aqui não acontece nada, se a gente ficar arrastando aqui, também não acontece nada. Em outras palavras o que acontece é o seguinte, o nosso usuário está bem restrito a apenas adicionar e alterações transações. Portanto, pensando justamente em permitir esse tipo de comportamento que é muito comum no nosso dia a dia em diversas apps, tanto você adicionar uma informação, alterar e também remover.

[00:43] A gente vai agora começar com um processo para também permitir que o nosso usuário faça isso também. E para isso a gente vai utilizar a nossa app base pra ver como que foi implementado e também fazer o mesmo tipo de implementação, o mesmo tipo de aspecto visual.

[00:55] Pegando aqui a nossa app base, eu já tinha colocado aqui uma transação pra teste, eu vou pressionar aqui e olha só, aparece esse menu aqui que é um Context Menu, vou até subir um pouquinho mais pra gente ver com mais facilidade. Veja que aqui aparece esse "Remover" que é justamente um Context Menu pro item da nossa List View. Eu vou pressionar aqui, vamos ver, agora olha só, ele foi lá e removeu a transação.

[01:20] Então é justamente esse comportamento que a gente vai colocar agora para nossa app. Vamos lá, como que a gente pode estar fazendo para criar aquele menu na nossa List View? Basicamente a gente pode vir aqui na nossa Activity, "ListaTransacoesActivity" e repara que aqui quando a gente faz aqui o nosso create, a gente tem o nosso "configuraLista". A gente pode entrar dentro dele.

[01:40] E aqui dentro é onde a gente tá fazendo a configuração da nossa lista. Portanto, é justamente aqui dentro também que a gente pode criar aquele menu pro nosso usuário, pra poder pegar cada item da nossa lista. Como que a gente pode estar fazendo isso? Como a gente pode estar criando um menu pra nossa ListView.

[01:53] Como a gente pode ver, aqui dentro do "with" a gente está criando a nossa ListView. Portanto a gente pode vir aqui, pular a linha e chamar mais um membro dele. E qual é o membro que cria esse nosso menu de contexto? Basicamente o membro é chamado de, no caso uma função né, "onCreateContextMenu", reparam que tem esse que é o "onCreateContextMenu". O qual a gente espera nessa caso vai ser a criação, e também já manter um Listener pra quando a gente tocar no nosso item já aparecer pra gente.

[02:22] Ao invés de ser um "onCreateContextMenu" a gente vai pegar o "onCreateContextMenuListener". Reparem que tem três maneiras de implementar esse cara aqui. A primeira é aquela que recebe a interface, portanto se a gente foi fazer ela, a ideia é fazer com que a gente faça com Object Expression. E a gente vê que no final aqui a gente tem aquele "setOnCreateContextMenuListener", que é justamente a implementação por expressão lambda. É justamente essa que vamos pegar.

[02:42] Eu vou dar um enter aqui, e ele começa até querer renomear os parâmetros para ver o que que dá mais sentido pra gente, a princípio a gente coloca o "ContextMenu", mas a gente pode deixar como "menu" que justamente representa

o menu que a gente vai utilizar. Em seguida ele tem a view, que só sugere a view mesmo, e depois ele tem esse "ContextMenuInfo", que é o menu info.

[03:00] A gente pode colocar aqui essas informações e agora a gente implementou aqui a nossa expressão lambda. Agora que a gente tem essa expressão lambda, a gente pode criar o nosso menu. Como que a gente cria ele? A partir desse parâmetro mesmo, que ele representa o menu que vai ser criado.

[03:14] Aqui a gente pode vir aqui em "menu" e pedir pra ele adicionar, então add. Reparem que ele tem essa sobrecarga de construtor pra poder criar. Qual que é a sobrecarga que a gente vai utilizar? A gente vai utilizar essa sobrecarga aqui, por que vamos utilizar ela? Porque a partir dela a gente vai ter a capacidade de colocar tanto aquele título de remover como também dar uma identificação pro nosso menu.

[03:32] Porque dessa forma a gente vai conseguir captar quando esse menu for tocado, por isso que vamos utilizar elas. A gente vai utilizar ela, e agora a gente precisa mandar esses parâmetros. O primeiro dos parâmetros, veja que não é tão intuitivo, só tá falando que é um "int" aqui, ele refere-se justamente a um grupo desse menu. A princípio a gente não vai colocar um grupo pra esse menu. Portanto a gente pode colocar um valor que é inativo, que não vale nada. Qual que é esse valor que a gente pode colocar? [03:58] A própria classe menu do android aqui já tem uma constante pra isso, uma constante pra identificar que o valor não está sendo válido. E como que é essa constante? É "Menu.NONE", então quando a gente coloca essa "Menu.NONE", significa que a gente não quer passar nenhum parâmetro pra o construtor dele, é por meio dela.

[04:14] Em seguida, ele espera o ID, neste caso a gente vai colocar uma informação pra conseguir identificar quando ele foi tocado. Qual a informação que a gente vai colocar? O valor 1. Então a partir desse valor 1 a gente vai conseguir identificar o nosso menu. Em seguida ele pede pra nós colocarmos uma ordem pra esse menu. Porque se a gente tivesse mais que um menu poderíamos indicar qual seria a ordem.

[04:32] Nesse caso a gente não vai colocar nenhuma ordem, portanto, novamente, "Menu.NONE". Então agora que a gente colocou o grupo, o ID e a ordem, a gente pode colocar o título que a gente espera. Neste caso, o título que a gente espera, a gente viu é justamente aquele remover então a gente pode colocar aqui uma string "Remover".

[04:48] A gente conseguiu criar aqui o nosso menu. Vamos só testar pra ver se está funcionando? Vamos lá, Alt+Shift+F10, veja que o Android Studio conseguiu executar. Vamos ver aqui, vamos criar uma transação, criamos. Vou adicionar uma receita de R\$ 100, agora vou pressionar. Olha só, o remover aparece, se a gente clica ele não vai acontecer nada. Justamente porque a gente não está tomando uma ação quando ele é clicado.

[05:12] Então o que a gente precisa fazer agora galera? A gente precisa agora implementar uma função que vai ficar responsável em pegar os menus de contexto que são tocados ali na Activity, e verificar se esse menu ele representa esse menu que a gente criou.

[05:22] E como a gente faz isso? A gente tem uma função da própria Activity que a gente pode sobrescrever pra isso. Qual é essa função? É justamente a "onContextItemSelected". Então a partir dessa "onContextItemSelected" a gente verifica qual o menu de contexto dentro da Activity foi selecionado, e verificar também se ele se enquadra justamente a esse ID que pegou aqui, que é o 1.

[05:46] Então a gente verifica dessa maneira. E como que a gente pode fazer isso então? Reparem que aqui a gente tem esse "item", que ele é justamente o menu item. A partir desse menu a gente tem capacidade, por exemplo, de pegar um ID desse menu. E é justamente isso que a gente vai fazer.

[05:58] Então aqui dentro a gente vai pegar o "item" e aqui a gente vai pegar o "itemid". Reparem que a gente tem aqui "itemid". Só que aí novamente tem aquele detalhe. Ele parou de compilar, porque ele parou? É porque a gente está lidando com um tipo que pode receber valor nulo. A gente precisa garantir que esse cara aqui ele não é nulo.

[06:17] A gente pode isso com uma safe call por exemplo, então aqui a gente está garantindo e a gente chama o menu "itemid". Agora vamos atribuir pra uma variável, então ".val" aqui, e a variável que vai ser aqui é justamente o "idDoMenu" que fica bem descritivo pra gente o que significa.

[06:36] Agora que a gente tem aqui esse ID, o que a gente precisa verificar? A gente precisa verificar o seguinte. A gente precisa colocar por exemplo um if aqui, e verificar se o "idDoMenu" é igual a 1. Que é justamente o ID que a gente espera pro menu. E agora que a gente está verificando podemos abrir o escopo do if e tomar uma ação.

[06:52] Vamos só realmente testar pra ver se isso funciona? Como que a gente testa? Colocando um Toast, por exemplo, então Toast.makeText, eu vou passar o contexto, eu vou passar um texto pra poder indicar que o menu remover foi tocado. E por fim eu vou colocar um tempo aqui "LENGTH_LONG".

[07:12] Agora é só pedir pra ele mostrar e agora a gente pode testar. Vamos executar aqui, Alt+Shift+F10, agora vamos esperar o Visual Studio, veja que ele conseguiu executar. Vamos testar nossa app. Adicionar uma transação, colocar uma despesa nesse momento, R\$ 100 aqui, vamos ver, eu estou pressionando, aparece o remover, e agora veja que está aparecendo que o nosso toast. Então aqui a gente está conseguindo captar o nosso menu. O que a gente precisa fazer agora?

[07:40] A gente precisa tomar uma ação na qual a gente vai conseguir remover a transação. Vamos lá. Eu vou apagar agora o Toast, apaguei, e agora a gente precisa, de fato, pegar a posição da nossa transação que foi tocada. Porque é dessa forma que a gente vai conseguir remover, porque a partir da posição a gente pode pegar a nossa lista de transações e falar, remova esse elemento a partir dessa posição. A gente consegue fazer isso.

[08:02] E como que a gente pode por exemplo pegar a posição do item que foi tocado? Se a gente, por exemplo, usar esse item, a gente percebe que ele é só o menu item. Ele não tem uma informação do que aconteceu dentro do menu, ele não as informações desse menu. Ele só tem justamente as informações mais básicas, mais genéricas.

[08:20] Que é justamente o "groupId" e o "itemId". Então pra gente conseguir mais informações do nosso menu, a gente precisa chamar mais uma property que é especialmente a "menuInfo". Então a partir dessa "menuInfo" a gente tem a capacidade de pegar mais informações sobre esse menu. Por exemplo, pegar a posição do item que foi tocado entre outras coisas. Só que apenas com menuInfo, o que a gente teria como retorno?

[08:41] Vamos dar uma olhada aqui, vou apagar e vou retornar. A gente tem um "ContextMenu", "ContextMenuInfo". Que é no caso uma classe bem genérica pra "menuInfo". E no caso essa classe genérica ela não tem capacidade de pegar a posição que foi tocada desse item que a gente colocou aqui, esse nosso item do ListView.

[08:53] E pra gente conseguir isso, a gente precisa fazer um cache de uma classe específica. E qual é esse cache que a gente espera? O cache justamente de "AdapterView.AdapterContextMenuInfo", a partir desse "AdapterContextMenuInfo" a gente tem muito mais informações sobre o que aconteceu no nosso menu, no momento que a gente tocou ali no ContextMenu.

[09:18] Então aqui eu vou devolver ele, vou devolver como adapterMenuInfo. E agora com esse adapterMenuInfo a gente tem capacidade de pegar a posição do item que foi tocado que a gente fez a partir da nossa ListView. Então aqui eu vou colocar "adapterMenuInfo.position". A gente tem a capacidade de pegar a posição, então aqui eu vou devolver pra uma variável que vai ser "posicaoDaTransacao".

[09:46] É justamente esse valor aqui que se refere-se a posição da nossa transação que a gente está tocando. E agora que a gente tem essa posição, a gente pode chegar na nossa property "transações" e falar pra ela remover. Repara que ela tem essas duas chamadas aqui. Esse remove, que é justamente o elemento que a gente quer, e esse removeAt que é justamente o remove em, e ele pede uma posição que é o índice.

[10:06] É justamente esse que a gente vai usar. Então `removeAt(posicaoDaTransacao)`. Agora que a gente mandou a posição da transação a gente só precisa atualizar os nossos componentes que é justamente o resumo e a lista. Então a gente pode pegar e colocar como `atualizaTransacoes`.

[10:23] Vamos testar pra ver o que acontece, `Alt+Shift+F10`, repare que ele conseguiu executar, vamos ver aqui. Vou adicionar uma transação aqui, uma receita R\$ 100, agora vou pressionar, ele aparece o remove, agora se eu tocar no remove ele vai lá e some. Então a gente conseguiu implementar esse comportamento pro nosso usuário.

[10:42] Independente se ele adicionar 500 mil transações aqui, ou outras transações que ele só quer testar, mas depois ele quer remover, ele consegue fazer isso. Então se a gente vir aqui em remover, a gente vai lá e remove. A gente conseguiu implementar isso pro nosso usuário.

[10:56] A gente conseguiu fazer essa implementação, só que agora vamos dar uma melhoria aqui nesse código, pra gente deixar ele coerente com tudo que a gente fez. Sempre que a gente fazia uma implementação e fazer refatoração. Nesse primeiro momento, repara que aqui a gente não usa `view`, nem `menuItem`, então a gente pode deixar o `_` sem nenhum problema.

[11:10] Em seguida aqui dentro desse `OnContextItemSelected`, repare que esse comportamento aqui é muito similar ao que a gente fazia no altera, como também no adiciona. Então a gente pode também extrair pra uma função, `Ctrl+Alt+M` e extrair pra função `remove`. A partir dessa função `remove` a gente está mandando a posição do item, então é só substituir o nome aqui, não vou deixar `posicaoDaTransacao`, vou deixar apenas `posicao` mesmo.

[11:37] E aqui a gente consegue remover a nossa transação. Então ficou bem mais simples de entender. Que a gente está pegando a `posicaoDaTransacao` e tá removendo aqui a partir da `posicaoDaTransacao`. Então a gente fez essa refatoração, e a gente está permitindo com que o nosso usuário também remova as transações, e agora sim a gente conseguiu colocar as features que a gente tem na nossa app base também. Até mais.