

## Criando o dialog para alteração

### Transcrição

[00:00] Atualmente, a nossa App permite que o nosso usuário adicione transações, então, só para a gente dar uma relembração de como a gente fazia, basicamente a gente clica aqui no botãozinho verde, a gente vem, por exemplo, em adicionar receita e, aqui dentro, ele abre aquele dialog que agente criou. Então a gente consegue colocar as informações.

[00:14] Como, por exemplo, o valor, eu vou colocar um valor de R\$ 100, a data eu consigo mudar também, por exemplo, 8 de novembro e, aqui, a gente tem a categoria, que a gente consegue selecionar algumas categorias pré-selecionadas para gente, então eu vou colocar como "Economias", aí eu vou lá em "Adicionar".

[00:30] Então reparem que, atualmente, a nossa App permite que o nosso usuário faça isso, só que agora o que acontece? Se a gente clicar aqui nessa transação, reparem que só tem aquele comportamento padrão do Android, que é justamente mostrar que foi clicado, eu clico várias vezes e só tá aparecendo que foi clicado, como, também, se a gente fizer aquele clique pressionado, que é o clique longo, ele também não acontece nada, o que eu quero dizer galera?

[00:51] É o seguinte, apenas esse comportamento de apenas adicionar uma transação, não é um comportamento que talvez o nosso usuário espera, ou seja, ele também quer, por exemplo, se ele adicionou uma transação errada, alguma informação errada, ele quer, também, alterar essa informação, ele também quer ter esse tipo de comportamento na nossa App.

[01:08] Em outras palavras, se a gente olhar aqui a nossa App base, vamos lá dar uma olhada nela, agente clica aqui no verdinho, a gente adiciona uma receita, R\$ 50, a data eu vou manter e a categoria também, eu vou adicionar e, além de adicionar, ela também, quando a gente clica novamente, ela abre um dialog novo, só que esse dialog novo é justamente para alterar uma receita.

[01:27] Pensando justamente em permitir esse tipo de comportamento para o nosso usuário, a gente vai começar a implementar esse código e, só para a gente ver, mais ou menos, como funciona esse "Alterar receita", basicamente ele está colocando as informações da transação, que a gente colocou lá na nossa lista e está permitindo que o cliente modifique.

[01:42] Por exemplo, a gente coloca agora R\$ 150 para aquela transação que era R\$ 50, a gente modifica a data do dia 6 para o dia 8 e, agora, aqui a gente modifica a Receita indefinida, por exemplo, para "Vendas", aqui a gente vai lá e coloca em "Alterar", reparem que ele consegue fazer isso para a gente e, é pensando justamente em dar esse comportamentos para o nosso usuário, que a gente vai começar essa implementação.

[02:04] Para isso, como a gente pode fazer, para colocar aquele código que a gente acabou de ver, ou seja, vamos começar por aquele Listener, que é clicar no item da lista, a gente pode vir aqui na nossa Activity. Então Control "N", "ListaTransacoesActivity", aqui dentro, deixa eu aumentar o código, Ctrl+Shift+F12, aqui dentro a gente tem que procurar justamente o componente que lida com a ListView, e onde que a gente pode achar esse componente?

[02:31] Se a gente ver aqui o nosso "onCreate", a gente percebe que a gente tem aquelas funções, que fazem as configurações dos componentes, a gente tem o "configuraResumo", a "configuraLista" e a "configuraFab", que é do "Floating action button", aqui dentro a gente pode vir na "configuraLista", então Control "B" nela e olha o que acontece, ela vai justamente, aqui, no código onde a gente coloca o "adapter" do nosso componente de ListView.

[02:51] O nosso componentes é justamente esse "lista\_transacoes\_listview" e, agora, a gente vai reutiliza-lo aqui embaixo, logo depois que a gente coloca o "adapter", para isso, a gente vem aqui embaixo, "lista\_transacoes\_listview" e, aqui, dentro desse componente, a gente vai falar o seguinte "eu quero implementar o Listener que você pegue o item da sua lista" como que a gente pode fazer isso?

[03:12] A gente faz uma função "setonitemClickListener", então ao invés de ser "setonClickListener", a gente coloca que o "setonitemClickListener", que é para pegar justamente o clique do item da lista e, aqui, novamente, ele tem duas implementações, a primeira, como a gente pode ver, é aquela que ele faz com que a gente implemente uma função anônima, Object Expression, como a gente viu anteriormente, que é justamente aquela classe anônima que a gente tem no Java também.

[03:39] A de baixo, é o nosso Expressão Lambda, como a gente viu anteriormente é bem mais resumido, portanto, eu vou pegar essa implementação de baixo. Eu vou dar um "Enter" aqui e ele apresentou aqui o nosso Expressão Lambda, só que, os nomes que ele colocou aqui, não está tão claro a que eles se referem para a gente, então vamos modificar esses nomes antes da gente começar com a implementação.

[03:58] O primeiro deles ele mostra como o "adapterView", se a gente for olhar de uma maneira mais resumida e objetiva para a gente, ele refere-se justamente ao "Viewgroup", ao "parente" do item que foi clicado, então a gente pode colocar como "parent", a gente pode fazer isso, dando o "Enter", ele está indicando que isso aqui é uma "View", realmente ele é a "view" que foi clicada, então a gente pode manter esse nome, em seguida ele vem com esse "i", o que significa "i" para a gente?

[04:19] Para a gente não significa nada, em outras palavras, esse "i", o que ele representa, nesse momento que a gente está implementando o "setonitemClickListener"? Ele representa a posição do item que foi clicado, a gente pode colocar como "posição" e, o último parâmetro, que é esse "i", que não tem nada a ver com a gente aqui, a gente não tem a mínima ideia do que seja, ele representa o ID do item que foi clicado, então a gente pode colocar como ID.

[04:38] Agora ficou bem mais claro o que cada parâmetro significa para a gente e, agora sim, a gente pode continuar com a implementação. Eu vou dar aqui um "Enter" aqui na expressão Lambda e vamos começar a implementação. Qual o primeiro passo que a gente tem que fazer no momento que a gente clica na nossa transação, o que a gente quer daquele comportamento?

[04:55] A gente quer, de fato, pegar a transação, a gente quer pegar a transação que foi clicada, para isso, o que a gente pode fazer? A gente pode pegar a nossa Property "transações" e, a partir dela, a gente pode pegar uma transação baseando-se na posição do item que foi clicado, essa "posição" que está recebendo aqui é a que vai utilizar para poder retornar a transação que foi clicada, agora que a gente está retornando ela, vamos atribui-la para um objeto, então o objeto "val", chamado de transição.

[05:21] Agora que a gente tem essa transação, a gente espera abrir aquela caixinha que a gente viu, que é o nosso dialog, e como é que a gente fez anteriormente? Se a gente vem um pouquinho aqui em cima, a gente tem essa função aqui chamada de "chamaDialogDeAdicao", o que ela faz?

[05:35] Ela faz uma instancia daquela classe que a gente criou, para criar um dialog de adição, a gente vai fazer um código meio que parecido com o que fez anteriormente, só que agora, a gente vai ter que fazer esse dialog com o objetivo de alterar uma transação, é justamente isso que a gente vai fazer.

[05:52] Só que agora, o que você deve estar pensando, a gente vai ter que implementar todas aquelas funções, aqueles propertyts, tudo que a gente fez no nosso "AdicionarTransaçãoDialog"?

[06:00] Vamos até ver aqui de novo, para a gente ver, mais ou menos, o que a gente fez, será que a gente vai ter que implementar tudo isso na mão? A princípio, se a gente fosse pensar o que a gente poderia fazer, seria realmente tem

que fazer novamente esses códigos, só que no Android Studio a gente tem uma featuring bem legal, para poder nos ajudar, nesse comportamento, para a gente também reutilizar alguns códigos que a gente viu por aqui.

[06:17] A gente tem a capacidade de fazer o seguinte, já que a gente vai utilizar boa parte desse código, como o caso de configurar o campo data, pegar as categorias, configurar o campo da categoria, a gente vai copiar todo esse código aqui, só que para uma nova classe, que vai ter responsabilidade alterar as transações, então, como que a gente pode fazer essa cópia aqui no Android Studio?

[06:35] A gente pode utilizar uma feature chamada de "Copy Declarations", como que a gente utiliza ela? Temos o atalho "F5", a partir desse atalho "F5", a gente habilita ou, no caso, a gente começa a ação da "Copy Declarations", o que ela faz? Ela pega o código que a gente usou aqui, o atalho, e ela faz uma cópia dele, só que agora ela fala o seguinte "Estou fazendo uma cópia dessa classe, agora, eu quero que você atribua o nome para a classe que vai ser copiada", que vai copiar toda essa classe.

[07:02] Qual é o nome que a gente espera para, alterar uma transação, baseando-se no dialog? A gente espera o nome de "AlterarTransacaoDialog", é justamente isso que a gente vai fazer, a gente vai fazer uma cópia do nosso "AdicionaTransacaoDialog" para essa nova classe chamada de "AlterarTransacaoDialog". Novamente, no mesmo pacote, que é o pacote de Dialogs, e aqui ele está falando se a gente quer deixar no source "main, ou no source "teste", no caso, é no source "main" mesmo, agora é só clicar em "Ok".

[07:28] Olha o que acontece, no momento que a gente usa essa featuring, ele conseguiu fazer a cópia, ele copiou tudo para gente, é claro, vai ter algumas referências que ele não conseguiu pegar, basta apenas a gente fazer o Import.

[07:38] Repare que, aqui, nessa parte que ele faz o synthetic, ele não conseguiu pegar, então a gente faz um "Alt" e "Enter" aqui, bem tranquilo, no primeiro "Alt" e "Enter" ele conseguiu pegar os outros componentes dessa view, aqui embaixo ele tem outros carinhas que ele não pegou, que é no caso as nossas extension functions. Então a gente vai vir e dar um import nelas também, vamos ver se tem mais algum carinha, aqui embaixo, novamente uma extension que ele não pegou, e aqui, ele conseguiu importar tudo.

[08:05] Agora a gente tem uma cópia exata do que a gente viu do "AdicionaTransação", só que agora, o nome dela é "AlterarTransacaoDialog".