01

Refatorando o código

Transcrição

[00:00] Agora que implementamos a funcionalidade para o nosso usuário conseguir alterar as transações, vamos dar uma olhada no que a gente fez aqui no código da "Activity" e, vamos ver, alguns pontos de melhoria que a gente pode fazer, ou seja, aquela mini refaturação que a gente costuma fazer.

[00:12] Considerado o que a gente mudou, a gente percebe que, nessa função "chamaDialogDeAdicao", a gente modificou o nosso Delegate, porque antes a gente só chamava essa função "atualizaTransacoes", sendo que, agora, a gente chama tanto o "add" do nosso componente, aliás, da nossa property "transacoes", que é a nossa lista de transações, a gente chama tanto esse "add", como também, o "atualizaTransacoes".

[00:32] Por que a gente está fazendo isso? Porque antes, o "atualizaTransações" tinha a responsabilidade, também, de adicionar a transação dentro da nossa lista, porque era a única maneira que que a gente fazia com o nosso usuário, só adicionar transações. Só que agora a gente fez com que, essa "atualizaTransacoes" ela também seja compartilhada entre outras funcionalidades na nossa App, como é o caso de alterar, então, por isso a gente tirou essa responsabilidade.

[00:55] Só que, agora, a gente tem que interpretar dois códigos, sendo que, a ação desses dois códigos é, justamente, adicionar uma transação, em outras palavras, a gente pode extrair uma função, para indicar o que ele está fazendo, ou seja, a gente pode chegar aqui e colocar um Control+M para indicar a extração que a gente quer. Então ele fala, se a gente quer extrair uma função para esse Object Expression, que é aquela implementação de função anônima, ou se a gente quer fazer essa extração para a nossa classe.

[01:20] Nesse caso, a gente quer para classe, porque vai ser ela que vai ficar responsável em adicionar e depois atualizar as transações, aliás, os componentes, que é o resumo e a ListView. Então a gente pode selecionar para classe e, agora, a gente pode colocar um nome para ela, a gente pode colocar o nome, a princípio, de "AdicionaTransacao", a gente pode fazer isso, só que reparem como que vai ficar, "AdicionaTransacao" e ele recebe uma transação, a gente pode fazer uma maneira resumida, porque a gente tá adicionando uma transação.

[01:45] Então, a gente pode deixar como "adiciona", a gente está indicando que essa adição, vai ser da nossa transação, eu vou dar um "Ok" aqui e, olha como ficou mais simples. A gente percebe que no momento que a gente chama esse "adiciona", logo depois que o Delegate é acionado, a gente vem aqui na implementação dele e ele adiciona e, também, atualiza as transações, então, essa implementação, essa refaturação, deixa bem mais simples a compreensão do que está acontecendo aqui.

[02:09] Continuando essa nossa análise, essa refaturação, vamos dar uma olhada no que a gente modificou também, reparem que aqui, quando a gente fez essa nossa implementação do nosso dialog, olha o tanto de códigos que a gente deixou, olha o tanto de coisas que a gente está tendo que implementar, para entender que a gente está chamando o dialog para alterar uma transação.

[02:25] Em outras palavras, a gente pode fazer a mesma coisa que a gente fez aqui, com esse dialog aqui, a gente pode criar uma função chamada "chamaDialog" de alteração, por exemplo. Então, eu vou fazer isso, a gente pode selecionar todo esse código aqui, por exemplo, e a gente vai dar um Control+M nele, a gente pode chama-lo de "chamaDialogDeAlteracao", a gente pode fazer isso, e agora a gente vai e dá um "Ok".

[02:46] Reparem, agora, como ficou essa chamada do "configuraLista", primeiro ele faz aqui a chamada do adapter, para configurar o adapter, e agora ele faz a implementação do "setonitemClickListener", que é para o Listener de cada item. Agora, ele chama primeiro a transação, baseando-se na posição que ele tem, e ele chama o nosso Dialog, para poder

alterar a transação, que é o dialog de alteração, enviando a transação e a posição, que são os parâmetros utilizados para conseguir alterar uma transação na nossa lista de transações.

[03:18] Então, a gente precisa desses dois parâmetros e, quem está chamando, precisa se responsabilizar e mandar essas informações para gente, por isso que a nossa função ficou bem clara, o suficiente para entender o que está acontecendo. Vamos dar uma olhada agora, por exemplo, como é que ficou essa extração ele a gente fez aqui, depois a gente volta aqui para dar uma melhorada.

[03:35] Nessa extração, a gente tem o mesmo comportamento que a gente tem na nossa outra extração de função, que é a "chamaDialogDeAdicao", reparem que a gente faz a instancia, enviando os parâmetros, faz a chamada do "chama" e, logo em seguida, a gente vai lá e faz aquele processo bem similar ao que a gente viu, também, no outro Delegate. Então, primeiro a gente está alterando aqui a transação e, depois, a gente está atualizando as transações, ou seja, a gente pode, também, fazer uma extração, aqui, para indicar que a gente só quer alterar.

[04:05] A gente pode fazer com que essas duas chamadas sejam extraídas, dou Ctrl+M para a própria classe mesmo, não para função anônima, e a gente pode falar que é uma chamada de função que vai alterar a transação, portanto, "altera". Reparem que ele até deixa aqui os parâmetros, primeiro a posição e depois a transação, já que a nossa intenção, por exemplo, é deixar a transação como primeiro parâmetro, para poder indicar que a gente quer alterar a transação, eu tô modificando a ordem.

[04:31] Antes ele estava assim, veja que dá para arrastar aqui, e agora modifiquei para que a transação seja o primeiro parâmetro, para a função ser chamada de "altera" e, depois, a gente interpreta o transação. Vou dar um "Ok" e olha como ficou, bem mais simples, agora que a gente fez essa refaturação para colocar essa chamada da função "altera", vamos voltar naquela função "configuraLista" e vamos ver o que a gente pode melhorar.

[04:53] Reparem que, a princípio, a gente tem essa chamada da "lista_transacoes_listview", duas vezes, e a gente viu esse cenário antes, quando a gente estava no nosso "resumoView". Em outras palavra, a gente pode utilizar aquela feature conhecida como "with", a gente pode chegar aqui e falar o seguinte, "with" com esse componente que é uma "ListView", eu quero que você abra um escopo para chamar todos os membros dele, se tem que referencia-lo novamente.

[05:17] Portanto, a gente pode chegar aqui e definir o nosso Adapter, eu vou copiar e colar, e, também, a gente pode chegar aqui e definir o Listener dele, a gente pode fazer dessa maneira, só que, agora, entra um detalhe, novamente, aqui é um problema de escopo, porque é um problema de escopo?

[05:32] Porque aqui, veja que esse "this", quando a gente tá mandando, antes de entrar no escopo do "with", a gente espera que ele seja o próprio context, só que o "this", dentro desse escopo do "with", é justamente esse objeto que a gente está mandando. Portanto, quando a gente entrar nesse tipo de cenário, a gente precisa colocar aquela label que a gente viu anteriormente, para poder indicar que esse "this" se refere a Activity, então, a gente precisa fazer isso, é uma das técnicas que a gente pode fazer.

[05:58] A label aqui, "ListaTransacoesActivity", essa é uma das técnicas que a gente pode fazer, também, a gente pode fazer uma outra técnica, que é, por exemplo, ao invés de criar a instância do adapter, aqui dentro, a gente pode criar o adapter, fora do escopo do "with", ou seja, a gente pode chegar aqui e fazer o seguinte, falar que Adapter que a gente está criando, a gente vai manda-lo para uma variável aqui, "val_adapterTransações" por exemplo, e aqui a gente vai lá e cria ele, a gente pode fazer isso.

[06:32] Vou até deixar um nome parecido com o que a gente fez aqui, que é para poder indicar que é o é "ListaTransacoesAdapter", aqui, Alt+Shift+F6, eu vou colocar, "ListaTransacoesAdapter", para poder indicar que é justamente o Adapter em relação a nossa lista de transações. E agora, ao invés de a gente ter que colocar toda essa instância, a gente pode referenciar, a partir desse objeto, que a gente criou ali em cima, essa é uma outra técnica que a gente pode fazer, pra evitar essa parte de colocar um label e deixar o nosso código cada vez maior.

[07:03] Essas são as duas abordagens, agora que a gente conseguiu colocar tudo dentro do "with", a gente não está mais com reutilizando, ou referenciando novamente o objeto, a gente conseguiu modificar essa chamada e, agora, aqui dentro do escopo do "with", considerando a "lista_transacoes_listview", a gente tem algumas coisas que a gente pode modificar, por exemplo, essa chamada da expressão lambda.

[07:25] Reparem que até o próprio Android Studio está falando que, esses parâmetros, que é o "parent", "view" e "ID", a gente não está utilizando, logo, a gente pode substituí-los por "underline", "undescore", da maneira que você preferir chamar, porque a gente não está utilizando, da mesma maneira como a gente viu anteriormente. Essa é uma das técnicas que a gente pode fazer para conseguir refaturar a nossa "configuraLista".

[07:48] Agora, só um último ponto que eu queria comentar com vocês, é justamente, a forma como a gente tá pegando esse ViewGroup, porque? Reparem que esse "chamaDialogDeAlteracao", como também, se a gente pegar o "chamaDialogDeAdicao", eles estão pegando duas vezes o "decorView", para poder pegar esse "ViewGroup", entre outras coisas.

[08:08] Além deles, também existe um outro componente, aqui, que faz a utilização dele, que é justamente o nosso resumo. Em outras palavras, faz todo sentido a gente transformar esse "window. decorView" em uma property da classe, para a gente reutilizar em todos os membros, faz todo sentido a gente fazer isso.

[08:24] Portanto, a gente pode chegar aqui em cima e falar o seguinte, eu quero pegar uma property que vai ser a "viewDaActivity", que é justamente o que a gente está fazendo ali, quando a gente pega um "decorView", a gente está pegando a view da Activity e, aqui, eu quero que ela seja inicializada pelo próprio "decorView", a gente pode fazer dessa maneira e, ao invés da gente ter que ficar, novamente, "windows.decorView", a gente pode chegar, em cada um deles e colocar, agora, a property que a gente espera.

[08:55] Ctrl+V aqui, que é a view da Activity, agora vamos ver aqui, um resumo, aqui no resumo também, "viewDaActivity" e agora, aqui, a gente vai lá e coloca como "viewDaActivity". Dessa maneira, a gente está fazendo com que, as nossas classes, aliás, os membros da Activity consigam pegar a mesma property, sem ter que ficar pedindo a "decorView", toda vez, a gente consegue fazer isso.

[09:18] Agora, vamos e ver se toda essa refaturação que a gente fez está funcionando? Vamos lá, Alt+Shit+F10, vamos executar aqui, veja que o Android Studio conseguiu executar para gente, vamos dar uma olhada como ficou, ele quebrou, porque será que isso aconteceu? Vamos entrar aqui no "Logcat" e ver qual foi o caso que entrou aqui, para poder quebrar.

[09:39] Reparem que, aqui, ele falou o seguinte, que, no momento em que a gente fez essa chamada, do "decorView", ele deu o Null Pointer, por que isso aconteceu? Esse é um assunto que a gente vai ver, logo mais no próximo vídeo, não precisa se preocupar, mas vou deixar esse mistério para vocês, para vocês entenderam porque, no momento que a gente fez com que o "decorView" entrasse com uma property, a nossa App quebrou. Logo mais a gente vai ver porquê isso aconteceu e como a gente pode evitar esse tipo de situação.