

05

## Criando o dialog para alteração

### Transcrição

[00:00] Só que agora tem um detalhe, por mais que tinha copiado, o código está exatamente o mesmo, o que a gente espera? A gente espera que esse cara, diferentemente do outro, que adiciona uma transação, ele altere, e como que a gente pode fazer isso?

[00:13] Nosso primeiro comportamento é, justamente, fazer com que ele receba uma transação, ao invés de simplesmente criar uma transação nova. Então, nesse momento que a gente faz essa chamada, desse função que chama, que é a função que chama o nosso dialog, a gente vai fazer o seguinte, a gente vai mandar a transação para ele também, a gente pode chegar aqui e falar que, além daqueles parâmetros que a gente espera, a gente vai mandar também uma transação.

[00:35] Até o Android Studio nos ajuda mostrando, a gente está mandando uma transação, "virgula" e, agora sim, ele está recebendo uma transação, agora que ele recebe uma transação, o que a gente fazer? A gente pode fazer, justamente, o processo de inicializar os campos, que é justamente o que a gente viu anteriormente, ele já inicializa os campos com os valores da transação que a gente tem aqui, então, é isso que a gente vai fazer.

[00:57] Vamos lá, o primeiro campo que a gente tem para poder inicializar é, justamente, o campo de valor, então a gente pode pegar o "campoValor", aqui, e agora a gente pode "setar" o valor dele, como que a gente pode fazer isso? Por meio da função "setText", a partir da função "setText", a gente vai pegar a nossa transação e, agora, a gente vai pegar o valor dela, e no momento que a gente pega esse valor dela, a gente precisa mandar o valor que é compatível com o "setText", que é, justamente, uma String.

[01:23] A gente pode mandar, por exemplo, o nosso "toString", é isso que a gente pode fazer nesse momento, como a gente viu anteriormente, a gente tava utilizando aquele formato de moeda, só que a gente precisa para esse momento do nosso dialog, vamos dar uma olhada, vamos ver o que acontece na nossa App base, se a gente clica aqui, repara que ele não coloca moeda, porque, esse campo, ele representa, de fato, o valor direto, então ele não representa a nossa moeda formatada, por isso, a gente vai deixar o "toString" direto.

[01:49] Caso você tenha estranhado, é justamente por esse motivo. Continuando com a inicialização dos campos, agora a gente tem, por exemplo, o campo data, no campo data a gente vai lá, coloca um "setText" e, nesse momento a gente vai lá e coloca o nosso para brasileiro, porque ele coloca aqui o formato para gente.

[02:08] Então, ao invés de só colocar o "toString", que vai vir aquela "toString" do Calendar padrão, que não tem muita relevância para a gente visualmente, a gente vai vir e colocar que a transação data vai pegar o "formataParaBrasileiro" para poder colocar a informação da data para gente.

[02:23] Agora a gente colocou, o campo valor e o campo data, o que nos resta agora é justamente colocar quem? É colocar aqui a informação da categoria, como que a gente pode colocar o campo da categoria? Basicamente, a gente precisa, de fato, de uma posição, por que a gente precisa de uma posição?

[02:39] Porque o campo da categoria, ele trata-se do que? Vamos lembrar novamente, vamos pedir para colocar o tipo, Alt+Enter, e olha, Specify type explicity, ele se trata de um Spinner, então, justamente, por ser um Spinner, ele não tem como colocar um valor, ele tem que colocar a posição do item que a gente quer que apareça, por isso que a gente precisa dessa posição primeiro e, como que a gente pode pegar essa posição?

[03:02] A gente pode fazer da seguinte maneira, o primeiro passo que tem que fazer, é pegar, justamente, as categorias disponíveis para gente e, como que a gente pega essas categorias? A gente pegou por meio daquele Array de String que a gente tem nos nossos Resources, para isso, a gente vai ter que pegar o nosso componente "contexto", que é por meio dele que a gente consegue pegar os Resources, e a gente vai pegar a property chamada de "resources".

[03:22] A partir dessa property "resource" a gente tem capacidade de chamar as Strings, Array de String no caso, então, "getStringArray", a partir desse momento que a gente está pegando o "getStringArray" a gente pode fazer o que, como que a gente pode falar pegar, por exemplo, uma categoria, ou de receita, ou de despesa?

[03:39] Lembra que a gente tem aquela função chamada de "categoriasPor" e ela recebe o tipo, então, nesse momento que a gente chama essa função, a gente pode mudar o tipo da nossa transação e ele vai trazer o Array de categorias, dependendo do tipo, se for o tipo de receita, ele vai trazer as categorias receita, se for tipo de despesa, ele vai trazer categorias de despesa, então a gente pode usar essa função.

[04:00] A gente vai lá, chama ela, "categoriasPor", e agora a gente manda o tipo, se a gente chega aqui e manda o tipo, reparem que a gente já tem tipo, aqui, que tá sendo enviado. só que agora tem um detalhe, toda transação, quando a gente criou, a gente fez com que o tipo fosse obrigado a ser enviado, em outras palavras, a gente não precisa mais desse parâmetro tipo, a gente não precisa mais utilizá-lo aqui, a gente pode apaga-lo, porque todas as transações já terão tipo, portanto, a gente pode utilizar o próprio tipo dela.

[04:26] Dessa variável "tipo", a gente vai criar logo aqui no começo, então, vai ser "transação.tipo.val" e, aqui, a gente vai criar o valor dela, aliás, a variável dela, desculpa, e agora a gente está utilizando o tipo baseando-se na transação. a gente não precisa pegar, por exemplo, o tipo de um parâmetro, porque a gente já tem uma transação que vai ter um tipo que, obrigatoriamente, ela vai ter, é um parâmetro a menos para a gente enviar.

[04:48] Agora que a gente está pegando todas as categorias, a gente pode devolver aqui, então, ".val", a gente vai lá e coloca "categoriasRetornadas", só para a gente diferenciar o nome dela, "categoriasRetornadas", baseando-se nessa chamada que a gente fez e, agora que a gente tem essas categorias retornadas, basicamente, o que a gente precisa fazer? A gente precisa pegar a posição do item que a gente tem dentro da nossa transação, como que a gente faz isso?

[05:13] A gente pega as categorias retornadas, que é, justamente, um Array e, aqui, dentro desse Array, a gente pode chamar a função "indexOf", que irá pegar justamente a posição, baseando-se no valor que a gente tem, de categoria, da nossa transação. a gente pode vir aqui na nossa transação e falar que quer pegar a partir da sua categoria e, aqui, a gente vai ter a posição da categoria, então, a gente vai ter "posicaoCategoria".

[05:37] Agora a gente consegue, por exemplo, indicar, para o nosso Spinner, qual é o item que a gente quer que ele mantenha aparecendo, no momento que a gente colocar para alterar, a gente tem aqui nossa posição, a gente pode vir aqui no "campoCategoria" e, aqui, a gente pode usar função "setSelection". Então, a partir dessa "setSelection", a gente pode colocar, por exemplo, para ele aparecer o item nessa posição, a gente tem até uma sobrecarga aqui, falando que essa é normal e, essa, com animação.

[06:05] A gente pode até utilizar essa com animação para a gente ver como funciona, a gente vai mandar, por exemplo, a "posicaoCategoria", e vamos mandar aqui uma animação como "true", da mesma maneira como a gente fez lá, no nosso menu verinho lá do FAB. Agora, perceba que a gente conseguiu colocar essas informações, aqui, para inicializar os campos, então, o que a gente precisa fazer agora que a gente está utilizando o nosso "AlteraTransacaoDialog" para aparecer o nosso dialog de alteração?

[06:29] A gente precisa utiliza-lo, voltando a nossa "ListaTransacoesActivity", a gente pode vir aqui no momento que a gente pega a nossa transação e começar o processo de instância para poder adicionar, agora, o "alteraTransacaoDialog".

[06:41] Vamos lá, "alteraTransacaoDialog", nele a gente vai criar uma instancia e, agora, a gente precisa mudar os parâmetros, o primeiro que ele pede, justamente, o ViewGroup, como a gente faz? A gente pega o "window", que é aquela property para poder pegar a janela e, aqui, a gente pega a View decorada, "decorView", a gente fala que essa View se trata do que? De um ViewGroup, agora a gente manda um "context", a gente pode mandar um "this", porque estamos utilizando a expressão lambda, que sobe o escopo, para quem a chama.

[07:07] Agora que a gente fez a instância, basta apenas a gente dar um "Enter" aqui e fazer chamada da "chama" e, aqui na "chama", a gente pode fazer o nosso Object Expression, "object" e, agora, a gente vai colocar quem? "Transação", aliás, desculpa, a gente vai ter que mudar todos os parâmetros, acabei errando um pouco aqui, então, qual é o primeiro parâmetro que a gente tem que mandar?

[07:26] É, justamente, a nossa transação, desculpa, tentei adiantar um pouquinho mais, o primeiro que a gente manda é a transação e, agora sim, a gente vai mandar o nosso Object Expression. Então, "object", a gente vai lá, a "transaçãoDelegate". A partir do momento que a gente tá fazendo essa chamada, deixa só eu colocar aqui o escopo dela, a gente precisa implementar essas funções que são necessárias, aqui a única função que é realmente necessária é justamente a "Delegate".

[07:50] Alt+Enter aqui, "Implement Members", ele aparece o "Delegate" e, aqui, ele veio com o nosso "TODO", podemos apagar porque a gente já vai fazer a implementação. Basicamente, a gente tem a transação que está sendo alterada nesse momento, a gente tem a nossa transação.

[08:01] O que a gente pode fazer nesse momento? Vamos tentar fazer o mesmo comportamento que a gente fez aqui, a gente atualiza as transações, a gente não precisa fechar o menu porque a gente não abre. Então, vamos só tentar atualizar as transações, "atualizaTransacoes", enviando o que? A transação, a gente tá conseguindo, agora, fazer um comportamento similar ao que a gente fazia no momento que a gente adicionava, só que agora a gente está com a ação de alterar uma transação.

[08:28] Vamos testar para ver o que acontece? Então, Alt+Shift+F10, vamos ver o que acontece, repara que o Android Studio conseguiu executar e vamos ver o que acontece aqui. Agora a gente tem a nossa App limpa novamente, vamos adicionar uma transação de receita, vou adicionar uma transação de receita bem simples, então, valor de R\$ 100, a data do dia 06/11 e uma categoria indefinida, adicionamos.

[08:50] Vamos clicar para ver o que acontece, clicando aqui ele aparece novamente o dialog, é claro, existe algumas peculiaridades aqui, a primeira delas, justamente, são esses valores que estão como "adiciona" ainda, sendo que, agora, a gente está alterando, mas, agora tem um detalhe, ele tá conseguindo colocar aqui as Informações para a gente, repara que, aqui, ele tá conseguindo colocar o valor de R\$ 100, como também, ele está conseguindo manter a data e a categoria aqui para gente.

[09:14] Ele está fazendo isso para a gente, só que agora o que acontece se a gente, por exemplo, tentar alterar aqui? Eu vou colocar aqui R\$ 150, eu vou colocar agora uma data do dia 8 de Novembro e eu vou colocar aqui uma categoria de "pagamento", por exemplo, então, eu adicionei, ao invés de alterar a transação, ele está adicionando uma nova transação.

[09:34] Pensando, justamente, em resolver esse detalhe, logo mais a gente vai ver o que a gente pode fazer para resolver essa peculiaridade que a gente viu, que ele está ainda adicionando, ele não está alterando a transação, que tem, aqui, no nosso ListView.

[09:47] inclusive, tem até um detalhe que é engraçado, se a gente clica novamente para ver se tem as informações, ele aparece R\$ 150, mas essas outras informações, a data e, também, essa outra informação da receita, ele ainda está mantendo a antiga, ele não tá colocando essa nova informação que a gente tem aqui.

[10:01] Pensando, justamente, em resolver esses detalhes, a gente vai ver no próximo vídeo como a gente pode resolver isso. Até já!