

Delegando responsabilidade do dialog

Se precisar do projeto com as alterações que foram feitas na aula anterior, você pode baixá-lo por meio [deste link](https://github.com/alura-cursos/financas-kotlin-parte2/archive/aula-5.zip) (<https://github.com/alura-cursos/financas-kotlin-parte2/archive/aula-5.zip>).

Atualmente a nossa App permite o usuário inserir receitas, entretanto, precisamos também possibilitar a adição de despesa.

Portanto, daremos início nessa implementação. Mas, antes mesmo de escrevermos o código para possibilitar essa feature, faremos uma refatoração no nosso código do dialog visando uma melhor leitura, qualidade e reutilização de código.

Extraindo função do código do dialog

Como primeiro passo dessa refatoração, extraia todo o código do dialog para a função `configuraDialog()`. Então, crie a classe `AdicionaTransacaoDialog` no pacote `ui.dialog` e envie a função `configuraDialog()`.

Note que nesse momento o código quebrou em diversos pontos. Não se preocupe, esse resultado já é esperado, ou seja, iremos resolver cada um deles.

Resolvendo as referências do código que cria a view

Como ponto de partida, extraia a função `criarLayout()` com todo o código responsável em criar o objeto `viewCriada`. Então, resolva todas as referências pedindo via construtor primário e transformando em properties as mesmas, por exemplo:

- `this` : trata-se da referência da Activity para representar um `Context`, logo, peça o `Context` via construtor.
- `view` : trata-se da view da Activity para obter acesso a um `ViewGroup`, portanto, peça o `ViewGroup` via construtor.

É importante notar que nesse momento vão existir códigos que não utilizaremos mais, como é o caso da property `window.decorView`, portanto, pode removê-la após resolver as referências.

Após essa refatoração, transforme a variável `viewCriada` em property e a inicialize com a função `criarLayout()`.

Extraindo as principais funções de configuração

Nesse ponto da refatoração, vamos aplicar a técnica de refatoração, ou seja, dentro da função `configuraDialog()` extraia as seguintes funções:

- `configuraCampoData()` : todo o código que configura o campo da data.
- `configuraCampoCategoria()` : todo o código que configura o campo categoria.
- `configuraFormulario()` : todo o código que configura o dialog que apresenta o formulário.

Repare que ambas as funções não precisam receber o objeto `viewCriada`, pois ele é acessível por todos os membros da classe.

Após extrair as funções, resolva as referências `this` substituindo cada uma delas pela property `context` .

Removendo parâmetro não utilizados nas expressões lambda

Depois de ajustar as referências de todas as funções, veja cada uma das expressões lambda, e confira quais parâmetros não estão sendo utilizados, então, substitua cada um deles com `_` , pois, dessa forma, evitamos a interpretação de algo que não faz diferença no resultado do código.