

Extraíndo funções

Caso você não tenha o projeto com o conteúdo da aula passada, você pode baixá-lo [neste link \(https://github.com/alura-cursos/financas-kotlin/archive/aula-2.zip\)](https://github.com/alura-cursos/financas-kotlin/archive/aula-2.zip).

Atualmente o nosso adapter, além de setar os valores nas views, também está com a responsabilidade de formatar para data no padrão brasileiro.

Em outras palavras, vamos refatorar o nosso Adapter para que ele tenha apenas a responsabilidade de computar as informações das views.

Para isso, vamos fazer com que todo o código que formata a data seja o comportamento de uma classe específica pra isso. Uma das possibilidades é extrair todo o código para uma classe específica, como por exemplo `DataUtils` ou `FormatadorDeData`.

Entretanto, já que estamos lidando com a API `Calendar`, faz todo o sentido que essa classe tenha essa responsabilidade, e podemos fazer isso no Kotlin utilizando a feature **extension function**.

Sendo assim, vamos começar com o processo de refatoração para delegar a responsabilidade para a classe `Calendar`.

Extraíndo função para formatar a data

O primeiro passo para essa refatoração é fazer com que todo o código que formata a data seja extraído para uma função que indique todo esse comportamento.

Sendo assim, crie a função `formataParaBrasileiro()` que retorna uma `String`, dentro dela, coloque todo o código que faz a formatação da data da transação.

Nessa função, receba o objeto `transacao` como parâmetro para que seja possível formatar a data de uma transação, então, no final da função, retorne a variável `dataFormatada` ou a própria função `format()`.

Por fim, após realizar o processo de extração de função, execute a App e veja se está tudo funcionando.