

02

Enum para os tipos das transações

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-partel/archive/aula-3.zip) (<https://github.com/alura-cursos/financas-kotlin-partel/archive/aula-3.zip>).

Se observarmos a nossa representação visual para cada uma das transações temos o seguinte resultado atualmente:



Agora se dermos uma olhada na forma como está na App base:



Repara que na App é bem mais fácil de identificar se uma transação é uma receita ou despesa, ou seja, repara que existe um reforço visual por causa das cores, por exemplo, quando temos uma transação na cor azul, significa que ela é uma transação de receita, porém, quando é vermelha, sabemos que é uma despesa.

Até mesmo o ícone nos indica isso, veja que a seta pra cima é referente a uma receita e a seta pra baixo indica uma despesa...

Considerando todas essas características, vamos começar a implementar esse reforço visual, para isso, precisamos primeiro conseguir identificar uma transação como receita ou despesa.

Atualmente não temos uma informação, ou melhor, uma property que indique essa característica de uma maneira fácil e objetiva, portanto vamos utilizar um enum para facilitar a nossa vida considerando esse contexto.

Criando um enum

Para criar um enum, vá até o pacote model e peça para criar um arquivo Kotlin, então, no campo **kind** escolha a opção **Enum class** e dê o nome do enum de `Tipo` para indicar o tipo das transações. Então adicione os seguintes valores para o enum:

- RECEITA ;
- DESPESA

Em seguida, declare uma property no construtor primário na classe `Transacao` do tipo `Tipo` chamada `tipo` (parece confuso... Traduzindo para código fica `tipo: Tipo`).

Por fim, modifique as instâncias da classe `Transacao` em todo o projeto para que agora recebam o tipo, então, rode a App e veja se tudo funciona como o esperado.

Sugestão: quando atribuir o tipo das transações, tente diversificar, ou seja, alterne em colocar uma `RECEITA`, outra `DESPESA`, uma `RECEITA`, outra `DESPESA` ... Mantenha essa estratégia de atribuição.