

01

Introdução

Transcrição

Olá pessoal! Meu nome é Alex Felipe. Sou instrutor da Alura, e estarei apresentando o curso de **Android com Kotlin Parte 2**. Caso você ainda não tenha feito a primeira parte desse curso, recomendamos *fortemente* que você a faça antes de começar a parte 2.

Na primeira parte do treinamento, fizemos uma lista de transações. Mas naquela *app* básica, também era possível calcular quanto as transações rendiam de **receita** e o quanto as mesmas rendiam de **despesa**. E então, tínhamos o cálculo total para ter o líquido do nosso financeiro.

Além dessa funcionalidade, temos uma lista que deixamos como *fixa* no primeiro curso. Entretanto, o ideal é fazer com que ela seja criada *dinamicamente*, ou seja, fazer com que o usuário tenha as opções tanto de **receita** quanto de **despesa**.

Na parte 2 do curso de Kotlin, daremos essa possibilidade ao usuário, de acrescentar receitas de um valor qualquer, settar a data dessa receita, e também definir uma categoria para ela. O mesmo acontecerá com as despesas, onde o usuário também terá a possibilidade de acrescentá-las na lista, de qualquer valor, data e categoria.

Mas pode surgir aquela dúvida: O que vamos aprender de parte *técnica* do Kotlin em relação a essa lista dinâmica? Basicamente, teremos a capacidade de usar recursos um pouco mais específicos da linguagem, como por exemplo as **expressões Lambdas**. As Expressões Lambdas estão presentes também no *Java 8*. Já no Android, a princípio não temos essa possibilidade de usá-las, pois utilizamos o *Java 6* ou *Java 7*.

Com Kotlin, nós temos essa possibilidade. Por mais que estejamos usando o *Java 6* ou *Java 7*.

Além disso, veremos como podemos usar um único objeto, e chamar diversas funções e *properties* em uma única estrutura.

Ao longo do treinamento da segunda parte do Kotlin, faremos diversos refatoramentos no código. Todas as vezes que implementarmos uma tela ou uma nova funcionalidade, por exemplo, passaremos por diversos processos de refatoração visando um código mais **objetivo**, e essa é uma das propostas desse curso.

Tentaremos melhorar cada vez mais o código utilizando as *features* do Kotlin.

Vamos começar?