



## Declarando a Higher-Order Function

Caso precisar do projeto com os ajustes que foram feitos na aula anterior, você pode baixá-lo a partir [deste link](https://github.com/alura-cursos/financas-kotlin-parte3/archive/aula-4.zip) (<https://github.com/alura-cursos/financas-kotlin-parte3/archive/aula-4.zip>).

Por mais que tenhamos implementado o padrão delegate com a interface `TrancasaoDelegate`, o Kotlin não permite que a implementação da mesma seja feita via expressão lambda, pois ele só realiza a SAM Conversion para interfaces do Java.

Considerando esse detalhe, vamos utilizar a função de alta ordem (Higher-Order Function) para permitir esse mesmo comportamento dentro do código Kotlin.

### Substituindo a interface do delegate por Higher-Order Function

Para realizar a modificação, vá na superclass `FormularioTransacaoDialog` remova todas as referências da interface `TrancasaoDelegate` e utilize a HOF (Higher-Order Function) que se chama `delegate` que recebe uma transação e que o retorno é um `Unit`.

Então, da mesma maneira, faça a mesma modificação na classe `AlteraTransacaoDialog` que implementa a própria função `chama()`.

Lembre-se que para enviar uma Higher-Order Function como parâmetro de uma chamada de função é necessário receber a mesma quantidade e tipo de parâmetros, como também, o mesmo retorno.

Para este exercício, não precisa rodar a App, pois a ideia é declarar a HOF primeiro.