

## Importando o projeto

### Transcrição

Começaremos abrindo o Android Studio da mesma maneira como em todos os cursos de Android, exceto pelo fato de estarmos utilizando a versão 3.0, versão necessária para a criação de uma aplicação em Kotlin de forma **nativa**, pelo suporte oferecido.

Repare que também trata-se da versão Beta 6. Atualmente, se acessarmos o [site do Android Studio \(https://developer.android.com/studio/index.html\)](https://developer.android.com/studio/index.html), veremos que a versão de *release* (a versão estável), é a 2.3.3, ou seja, não é a que dará suporte nativo na criação de aplicativos em Kotlin.

Portanto, iremos à opção "PREVIEW" no menu superior, com a versão em teste, que pode apresentar *bugs* inesperados, e então clicaremos no botão referente ao download da versão 3.0. Se no momento em que você estiver fazendo o curso surgir o *release* de uma versão 3.0, não é preciso usar a *preview*.

Neste primeiro momento, vamos abrir o projeto-base que utilizaremos para criar nossa app. Para isto, selecionaremos "Open an existing Android Studio project" e o arquivo `financas`, e o Gradle fará o processo de importação do projeto e das configurações necessárias.

Veremos um projeto Android usual e poderemos vasculhar o que existe ali abrindo o `build.gradle` por exemplo, arquivo de *build* do projeto, para verificarmos o SDK, o que utilizamos como dependência, e afins.

É possível fazermos o mesmo com o código fonte, em `br.com.alura.financas`, verificando seu conteúdo e as entidades que já conhecemos, como `activity`, e por aí vai.

Para entendermos o objetivo deste projeto, há um projeto pronto em Java, o qual conseguiremos consultar quando tivermos a aplicação em Kotlin atendendo às mesmas necessidades desta.

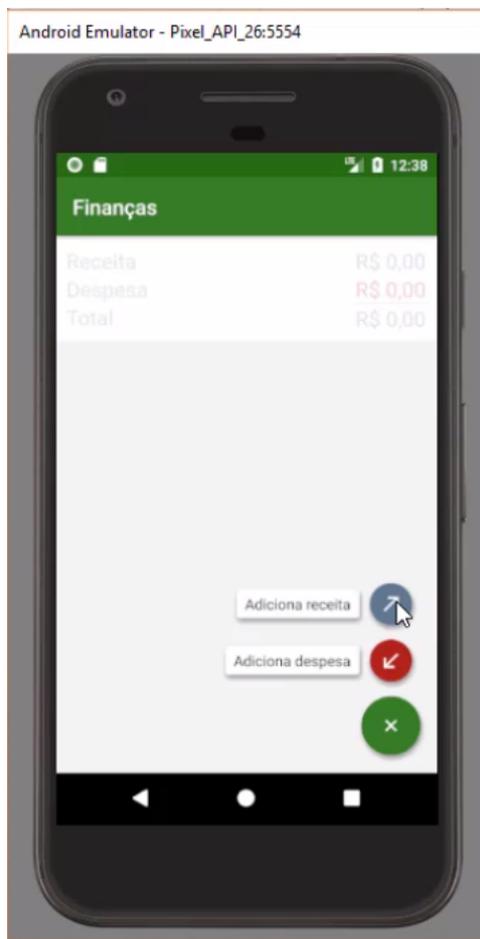
No entanto, neste momento não precisaremos ficar comparando os códigos de ambas as linguagens, pois chegaremos no mesmo resultado com Kotlin a partir do Java.

Só para entendermos o que se encontra "debaixo dos panos" e o que é executado como produto final, vamos rodar a app e ver o que acontece. Usaremos o atalho "Alt + Shift + F10", e escolheremos "app" na janela que se abre.

Outra janela está sendo aberta para o emulador, o "Pixel API 26", cujo API remete à versão 8 do Android, surgida recentemente. Apertaremos "OK" e aguardaremos. Também é possível clicar em "Play" e obter o mesmo resultado.

A plataforma do Android Studio é o **IntelliJ**, e há um curso voltado a ele [aqui \(https://cursos.alura.com.br/course/intellij-idea-truques-para-aumentar-sua-productividade-em-projetos-java\)](https://cursos.alura.com.br/course/intellij-idea-truques-para-aumentar-sua-productividade-em-projetos-java), em que aprendemos como aumentar a produtividade usando esta ferramenta.

Após o carregamento da nossa app - que não possui nada de muito relevante na tela inicial, certo? -, se clicarmos no botão verde com `+`, observe que aparecem opções para adicionarmos receitas e despesas:



O que acontece quando adicionamos uma receita? Acrescentaremos 100 como o valor, e manteremos a data como está. Colocaremos como categoria "Economia". Isso é adicionado à lista de transações, e na parte superior, com o resumo de tudo que temos, é mostrado este valor, bem como o total, que também é R\$100,00 .

Agora, vamos colocar uma despesa, de R\$20,50 , alterando a data para 22 de Setembro, para fins de teste. Na categoria, colocaremos "Comida", e na listagem obteremos como receita o valor de R\$100,00 , R\$20,50 de despesa e um total de R\$79,50 , cujo cálculo é realizado automaticamente.

Basicamente, é este o projeto a ser desenvolvido em Kotlin. Já temos essa aplicação desenvolvida em Java, que utilizaremos como referência para criar a nossa versão em Kotlin.