



Processo de Build

Professor Emerson Alencar

emerson@imd.ufrn.br

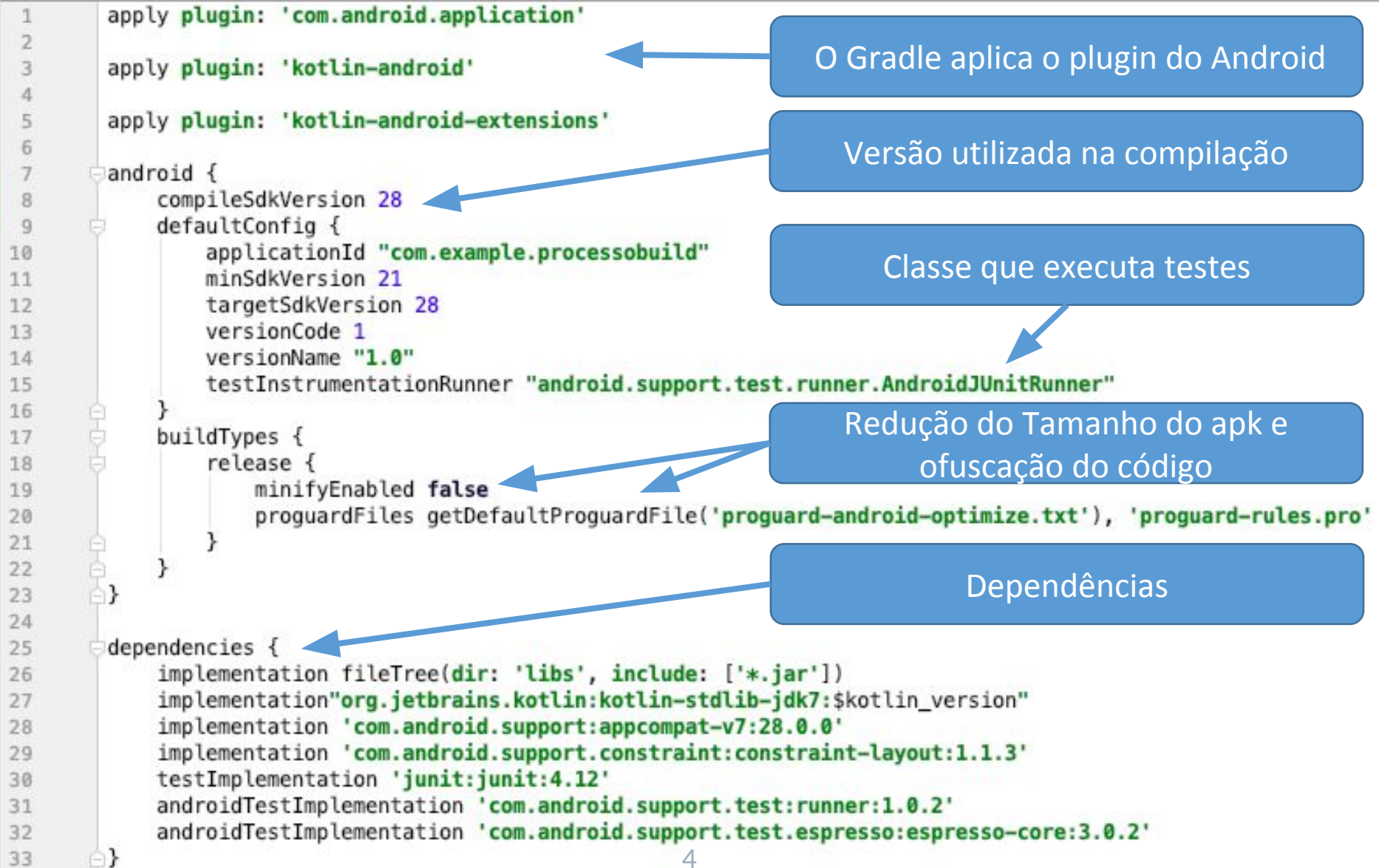
O que é Build?

- ▶ Construção do APK
- ▶ Debug e Release
- ▶ Gerenciador de Build do Android: Gradle

O Gradle

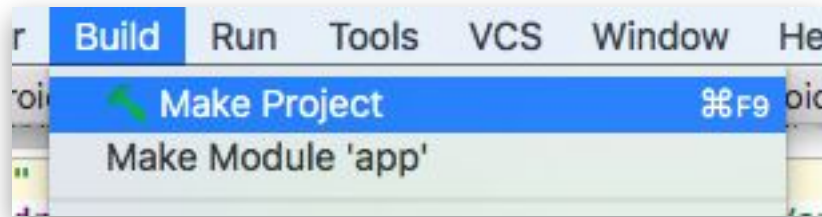
- ▶ Não é exclusivo do Android Studio
- ▶ É um componente de Software: Gerenciador de Build e dependências

build.gradle (Module: App)

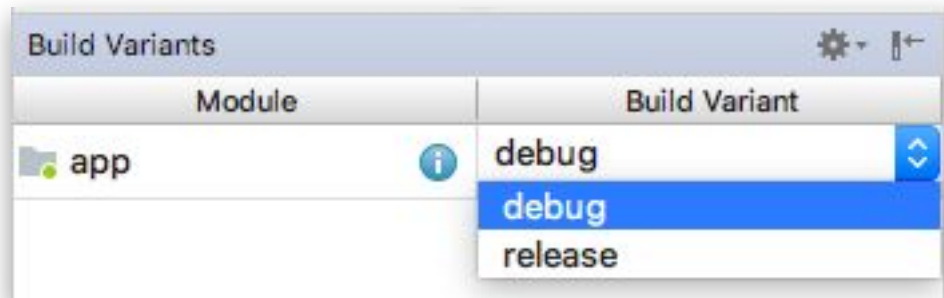


Build Padrão

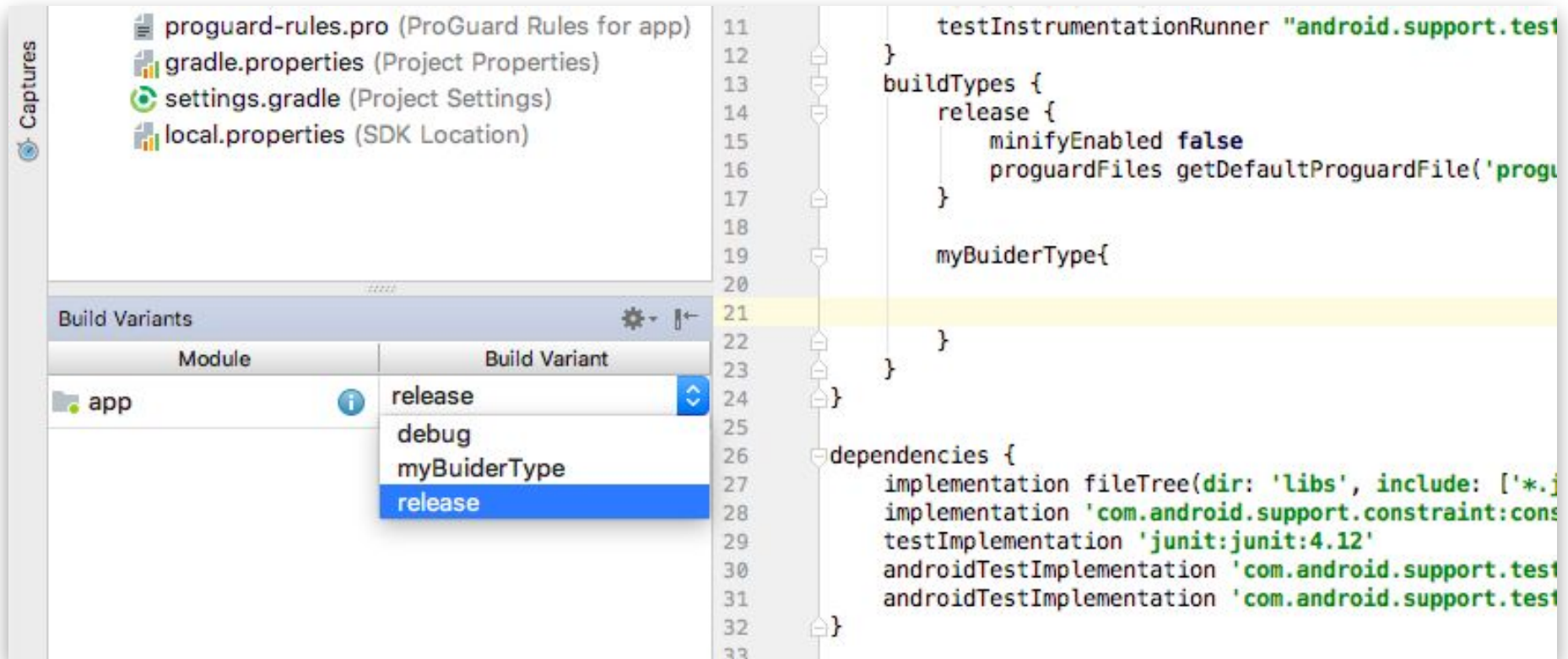
▷ DEBUG



▷ Como mudar para o build de release



CRIANDO UM NOVO BUILD



The screenshot shows an IDE interface with the Build Variants panel on the left and the build.gradle file on the right.

Build Variants Panel:

- Top section: proguard-rules.pro (ProGuard Rules for app), gradle.properties (Project Properties), settings.gradle (Project Settings), local.properties (SDK Location).
- Build Variants table:

Module	Build Variant
app	release
	debug
	myBuiderType
	release

build.gradle File:

```
11     testInstrumentationRunner "android.support.test
12 }
13 buildTypes {
14     release {
15         minifyEnabled false
16         proguardFiles getDefaultProguardFile('progu
17     }
18
19     myBuiderType{
20
21 }
22 }
23 }
24 }
25 }
26 dependencies {
27     implementation fileTree(dir: 'libs', include: ['*.j
28     implementation 'com.android.support.constraint:cons
29     testImplementation 'junit:junit:4.12'
30     androidTestImplementation 'com.android.support.test
31     androidTestImplementation 'com.android.support.test
32 }
33 }
```

Código de acordo com o Build

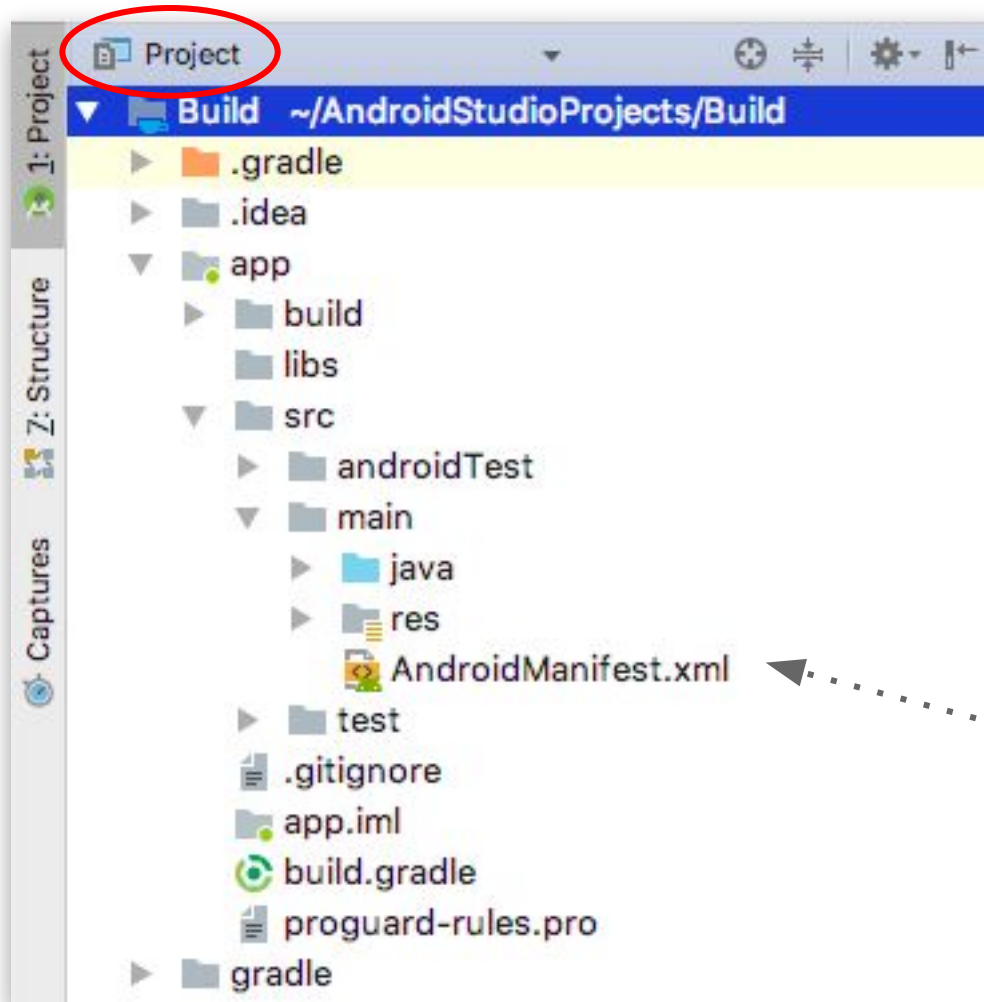
- ▶ Exemplo: Códigos que estejam presentes apenas no apk de debug
- ▶ Vamos a um exemplo:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3          package="br.imd.ufrn.build">
4
5      <uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION"/>
6
7      <application
8          android:allowBackup="true"
9          android:icon="@mipmap/ic_launcher"
```

Se o manifest.xml é único como posso usar essa permissão?

Deverias estar presente apenas no apk de debug

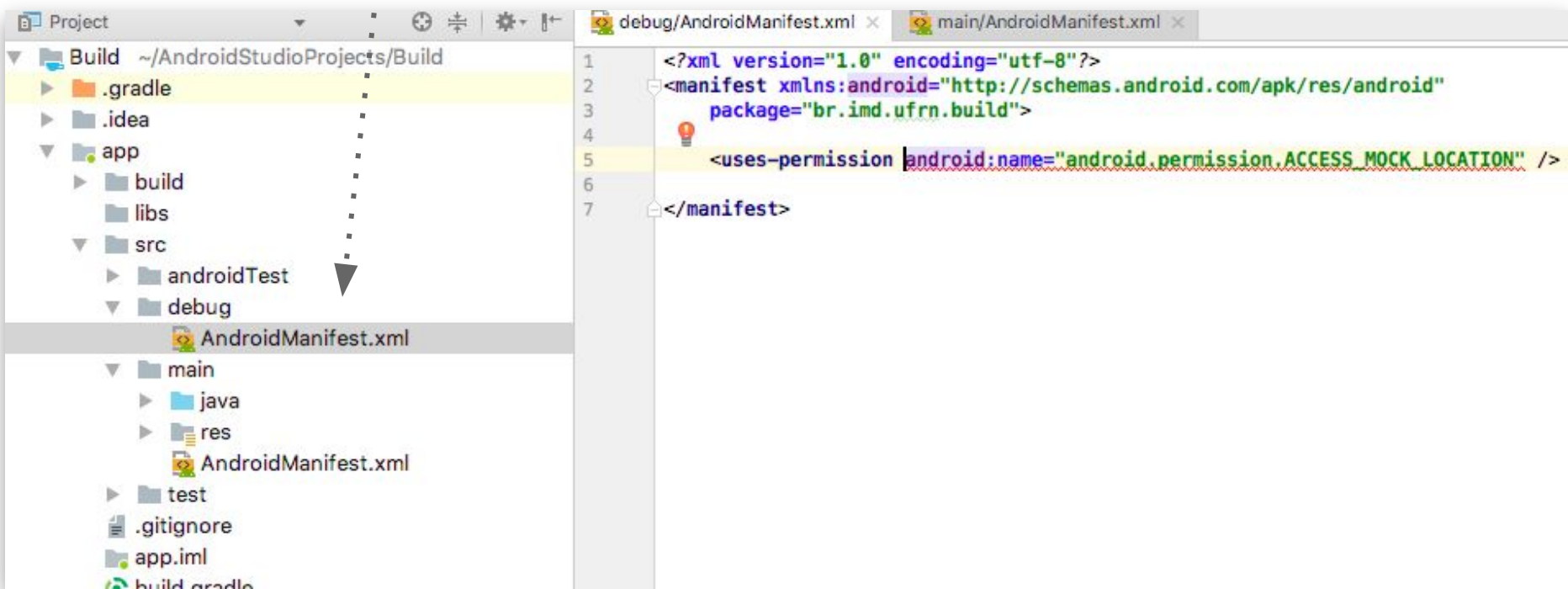
CÓDIGOS DE ACORDO COM BUILD



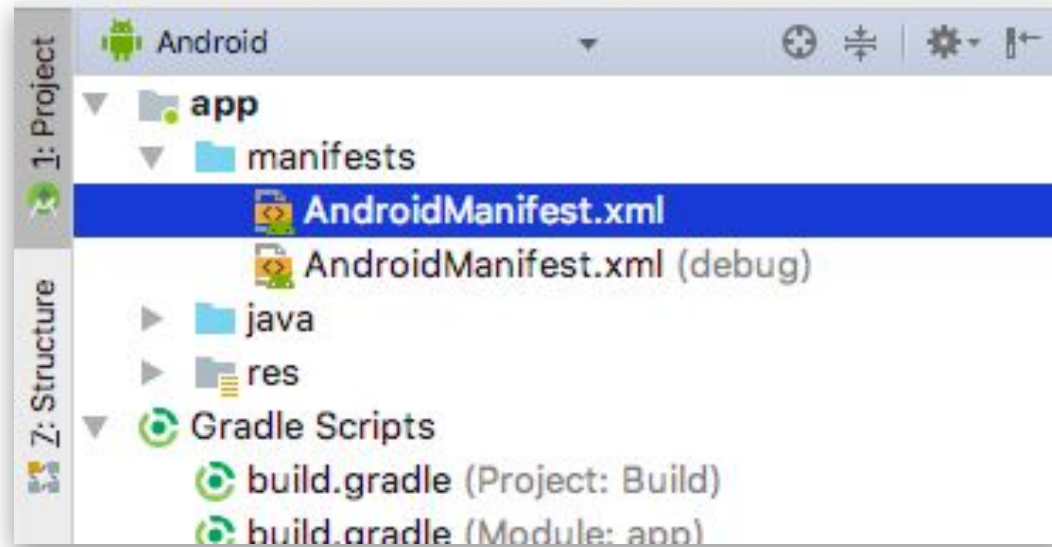
Um arquivo
AndroidManifest.xml

CÓDIGOS DE ACORDO COM BUILD

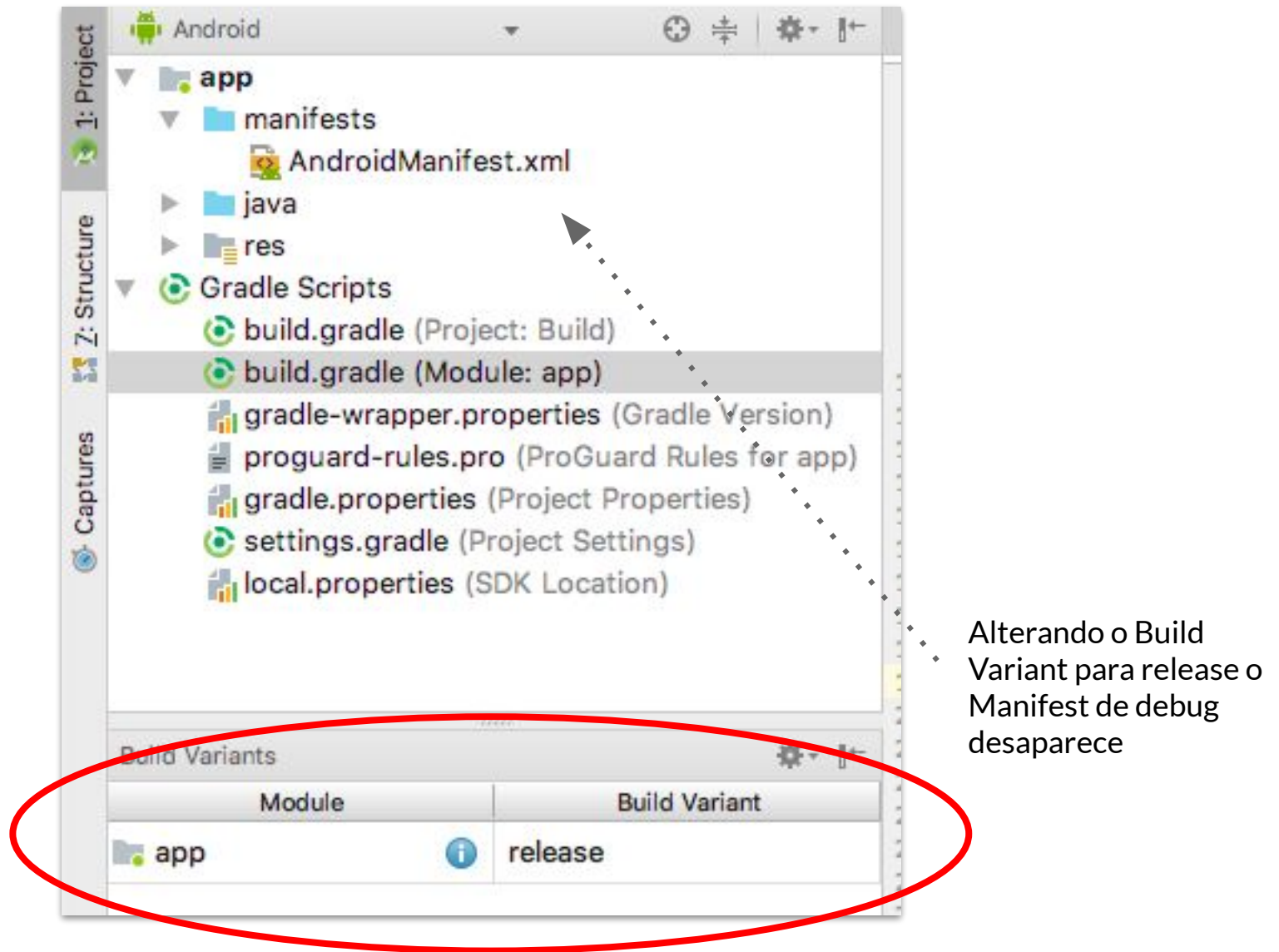
Cria uma pasta debug em src, copia o arquivo
AndroidManifest.xml com apenas a permissão para
debug



CÓDIGOS DE ACORDO COM BUILD



CÓDIGOS DE ACORDO COM BUILD



Configurar variações de produtos

The screenshot displays the Android Studio interface. On the left, the 'Build Variants' tab is active, showing a table with columns 'Module' and 'Build Variant'. The 'app' module is selected, and a dropdown menu lists the available build variants: 'phoneDebug' (highlighted), 'phoneRelease', 'tableDebug', and 'tableRelease'. A dashed arrow points from the 'phoneDebug' variant to the 'main' dimension in the code. On the right, the 'productFlavors' configuration in the 'build.gradle' file is shown. The 'flavorDimensions' is set to 'main', and the 'productFlavors' block defines two flavors: 'phone' with dimension 'main' and versionName '1.0-phone', and 'table' with dimension 'main' and versionName '1.0-tablet'. A dashed arrow points from the text 'Especifica nome do ProductFlavors dimensions do projeto' to the 'main' value in the 'flavorDimensions' property.

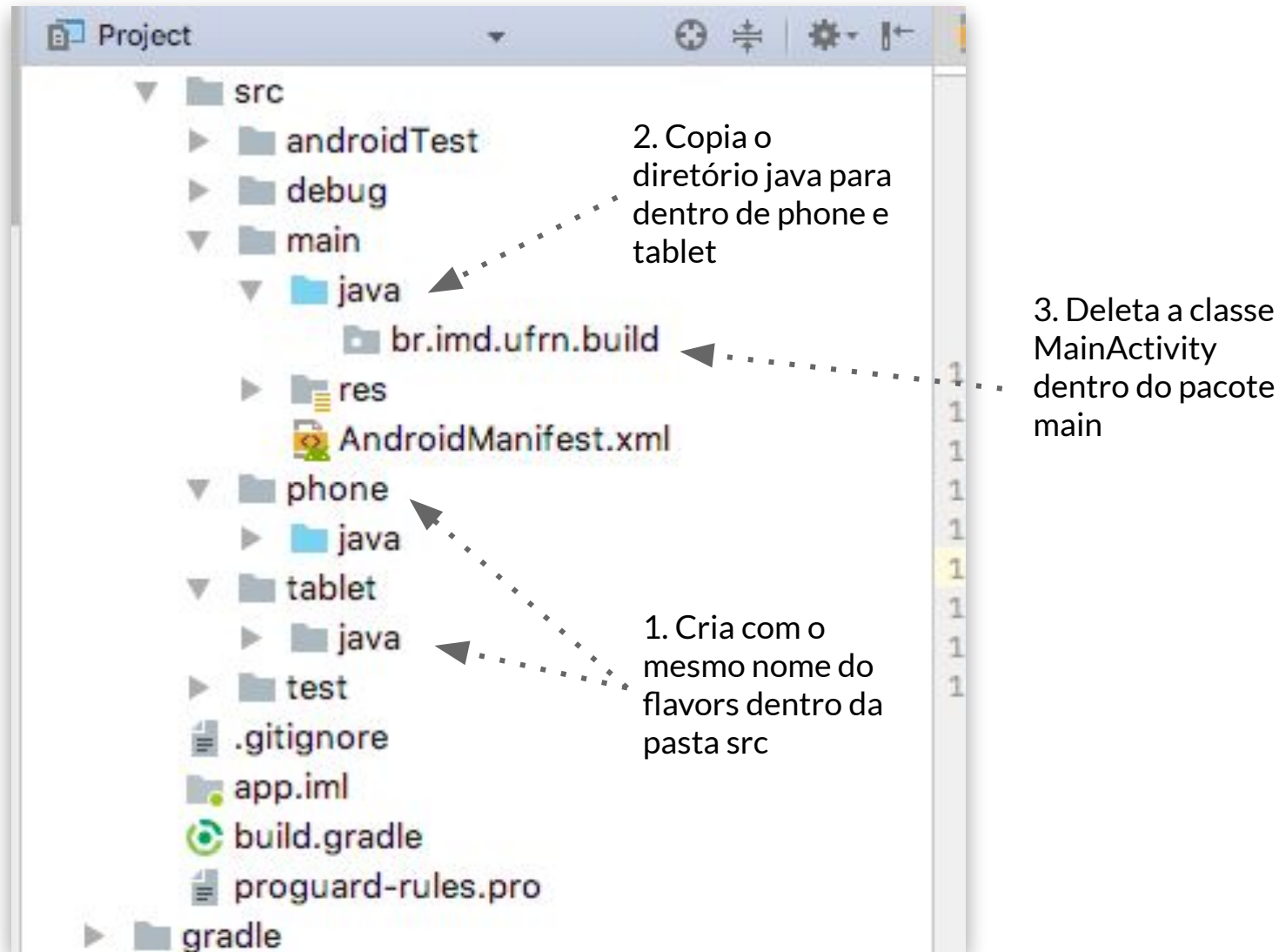
Build Variants

Module	Build Variant
app	phoneDebug
	phoneDebug
	phoneRelease
	tableDebug
	tableRelease

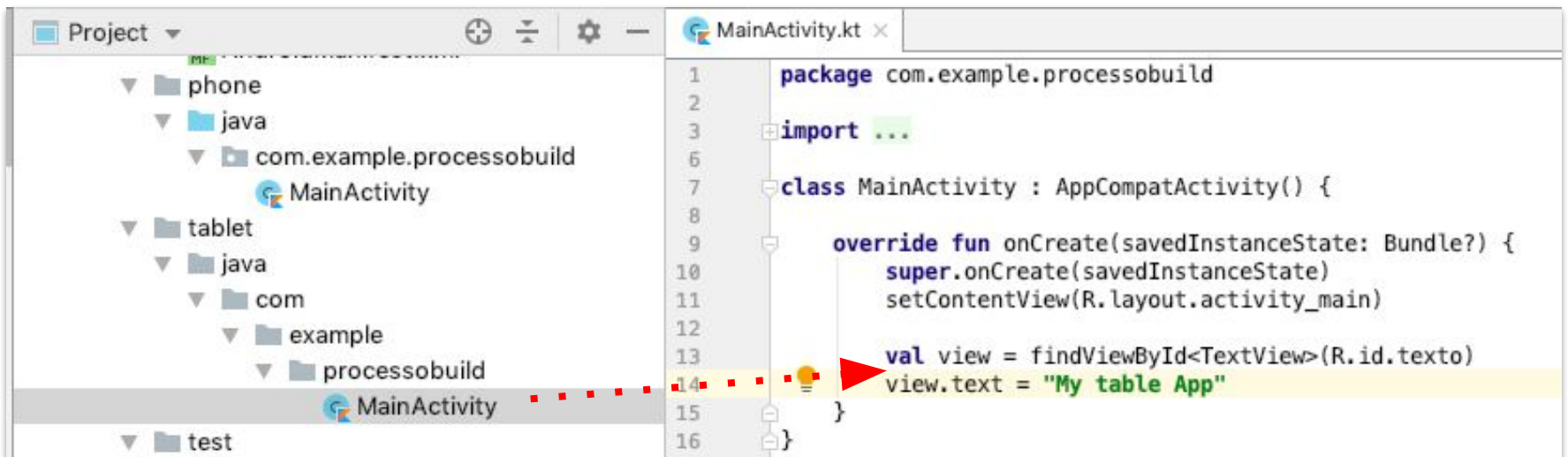
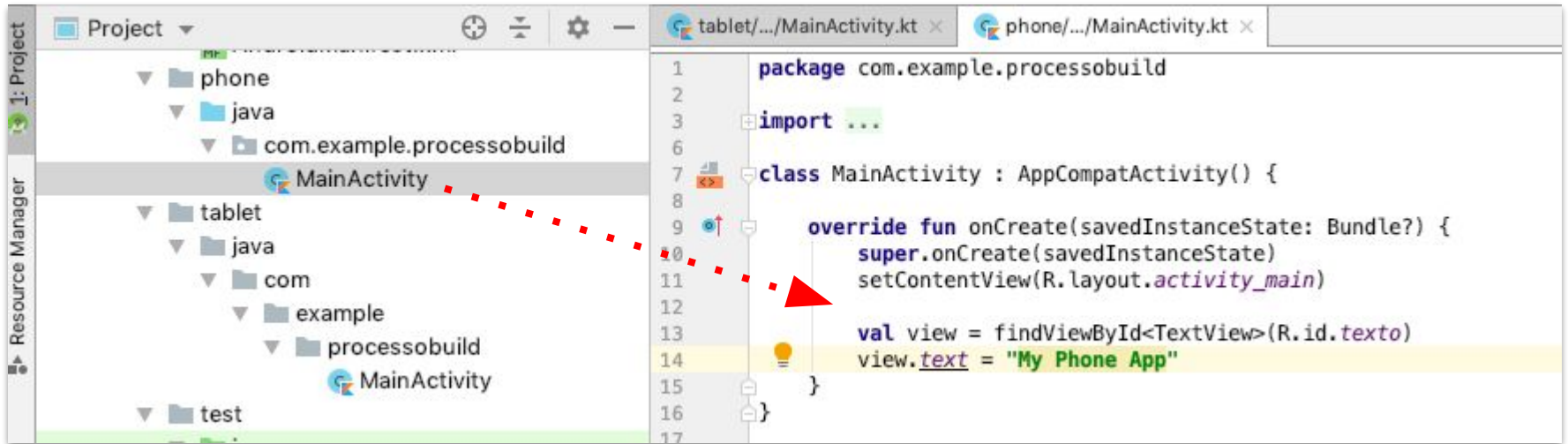
```
18 }
19
20
21 flavorDimensions "main"
22
23 productFlavors{
24     phone{
25         dimension "main"
26         versionName "1.0-phone"
27     }
28     table{
29         dimension "main"
30         versionName "1.0-tablet"
31     }
32 }
33
34
35 }
```

Especifica nome do ProductFlavors dimensions do projeto

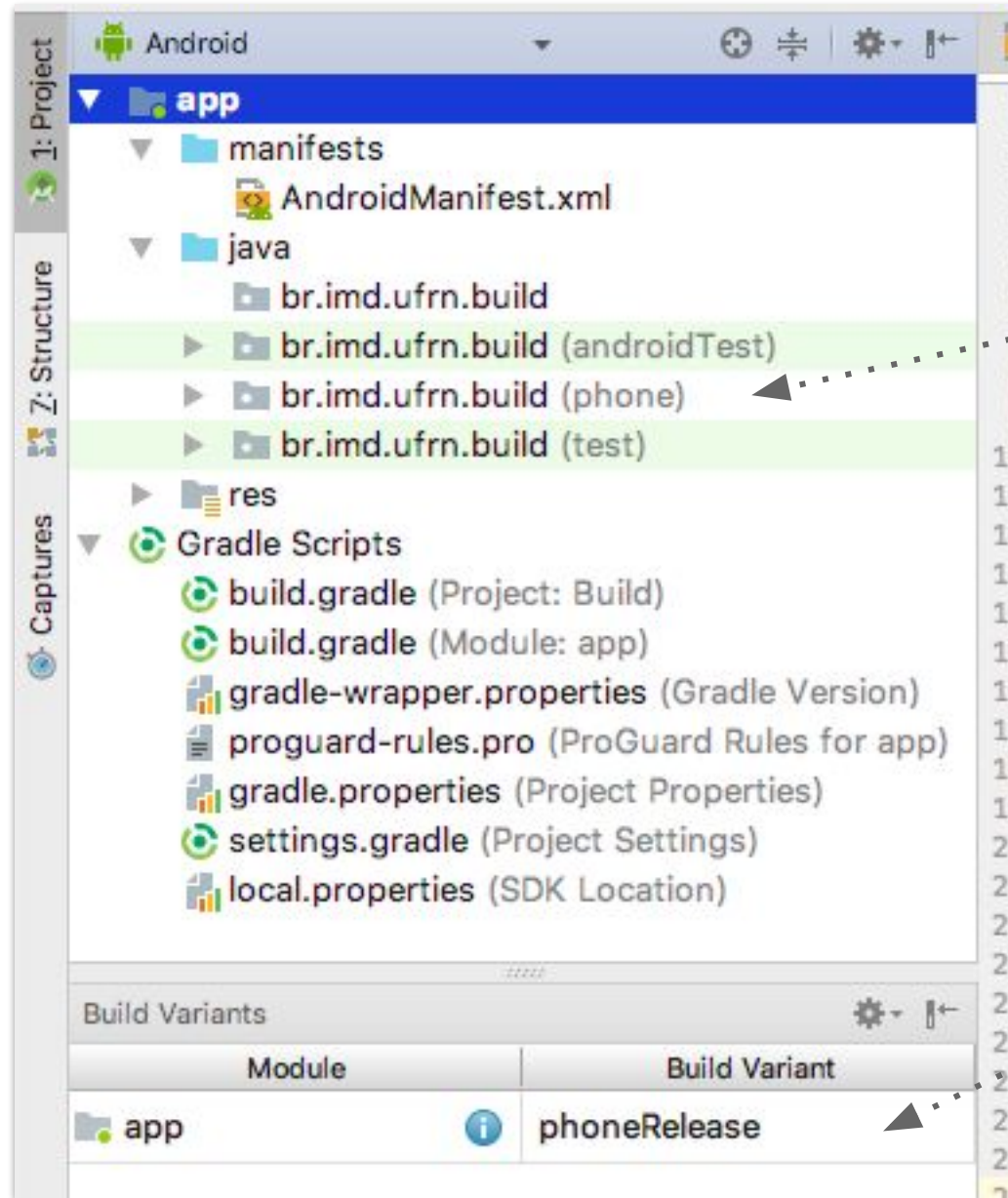
Alterando Código



Alterando Código



Alterando Código



Kotlin Android Extensions

- ▶ Acessa diretamente os componentes definidos no seu arquivo de layout
- ▶ É necessário aplicar o plugin no *build.gradle* do módulo e importar o arquivo de layout na activity

Import `kotlinx.android.synthetic.main.activity_main*`

```
1 package com.example.teste2
2
3 import android.support.v7.app.AppCompatActivity
4 import android.os.Bundle
5 import kotlinx.android.synthetic.main.activity_main.*
6
7 class MainActivity : AppCompatActivity() {
8
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12
13         txtTexto.text = "outro texto"
14     }
15 }
16 }
```

Importação do
arquivo de layout

Acesso feito após
o setContentView

Acesso direto
pelo android:id

Obrigado!
Dúvidas?

Professor Emerson Alencar
emerson@imd.ufrn.br