

# {CodeLab}

## Introdução ao Android Studio



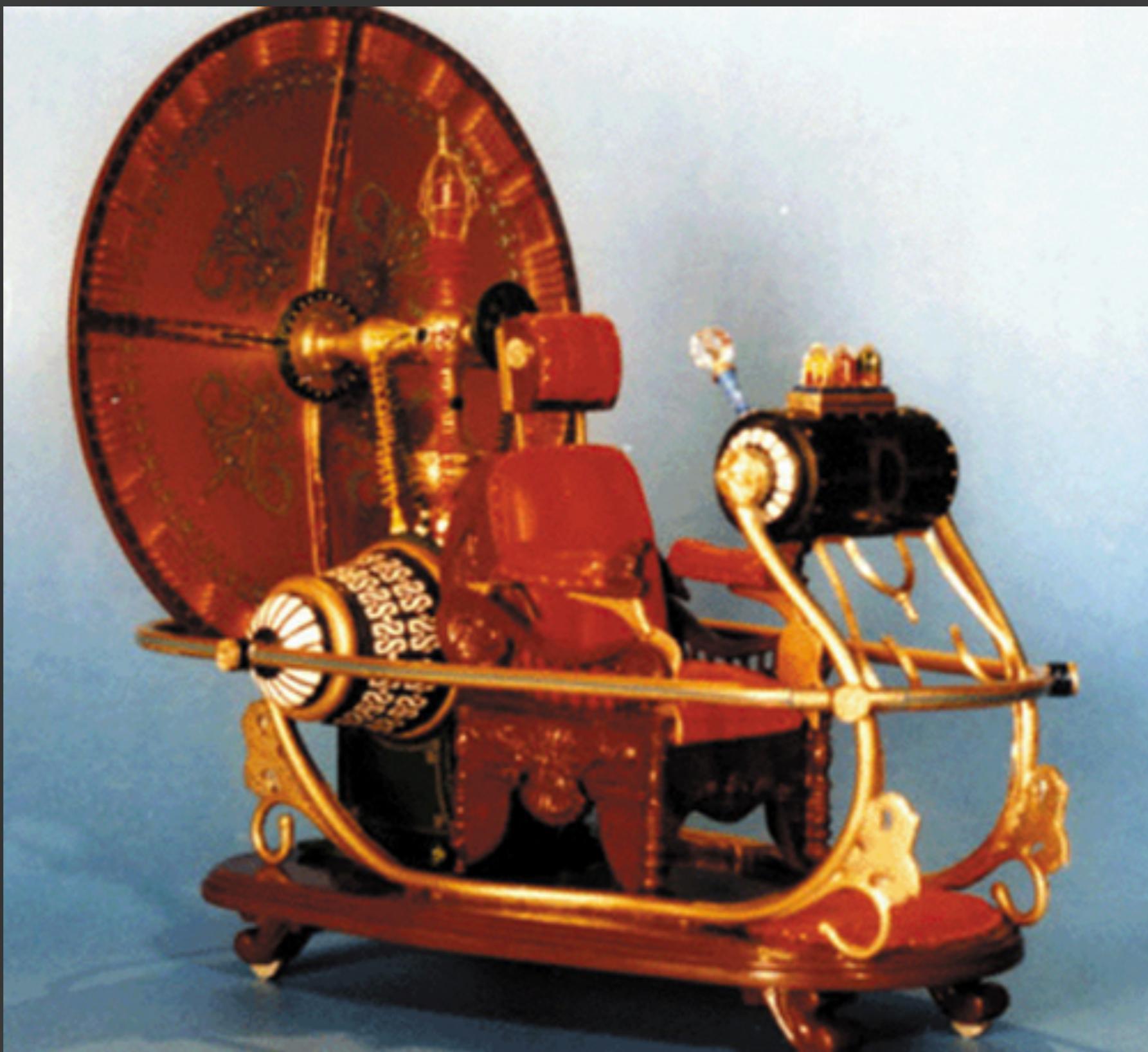
Ramon Rabello  
Tá Safo

# 7 anos \o/



Mas antes...

---



2007

## Android Developer Tools

---



O ADT foi a **primeira IDE** para **desenvolvimento** em **Android**

Todo o **processo** de **build** é baseado na **ferramenta Apache Ant**

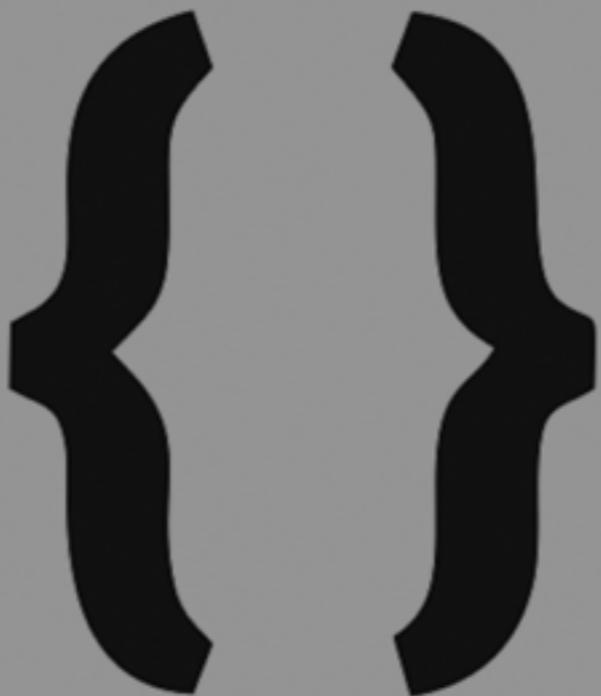


Construído na **infraestrutura do Eclipse**  
(Rich Client Platform)

2014+

O ADT será descontinuado!

---



2013+

# Android Studio



The screenshot shows the Android Studio interface. On the left is the XML code editor with the file "countdown.xml" open, displaying the layout for a countdown fragment. On the right is the "Design" tab of the Layout Editor, showing a smartphone preview of the stopwatch application with a red "START" button at the bottom.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#color/countdown_background"
    android:fill="@layout/countdown_layout"
    android:orientation="vertical">

    <TextView android:id="@+id/time_counter" android:textSize="40sp"
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:text="00:00:00.000" android:fontFamily="sans-serif-light"
        android:layout_marginTop="50dp" />

    <com.geekyup.android.usstopwatch.fragments.StopwatchCustomView
        android:id="@+id/cavview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_marginBottom="40dp"
        android:layout_below="@+id/time_counter"
        customWatchType="type_countdown"/>

    <LinearLayout android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:divider="@drawable/divider_vertical_blue"
        android:showDividers="middle"
        android:background="#color/blue_button_bg"
        android:orientation="horizontal">

        <Button android:id="@+id/startbutton"
            android:layout_width="match_parent"
            android:layout_height="45dp"
            android:background="@drawable/blue_button_bg"
            android:text="@string/start"
            android:textSize="@dimen/button_bar_portrait"
            android:textColor="@drawable/blue_button_text"
            android:fontFamily="sans-serif"
            android:layout_weight="1"
            android:contentDescription="Start/Stop Countdowm" />

        <Button android:id="@+id/resetbutton"
            android:layout_width="match_parent"
            android:layout_height="45dp"
            android:background="@drawable/blue_button_bg"
            android:text="@string/reset"
            android:textSize="@dimen/button_bar_portrait"
            android:textColor="@drawable/blue_button_text"
            android:fontFamily="sans-serif"
            android:layout_weight="1"
            android:contentDescription="Reset Countdowm" />

    </LinearLayout>

```

# O que é o Android Studio?

---



**Android Studio** é a nova suíte de **desenvolvimento** de **apps** em **Android**

Foi projetado utilizando a **infraestrutura** da IDE **IntelliJ IDEA**, da JetBrains



Utiliza o Gradle como novo sistema de build no processo de desenvolvimento



# Android Studio vs. ADT

## Funcionalidade

|   |  |  |
|---|---|---|
| Sistema de build  |  |  |
| Build de dependencias baseadas no Maven                           | Sim   | Não   |
| Variantes de build e geração de múltiplos APKs (ex: Android Wear) | Sim   | Não   |
| Acabamento de código Android avançando e refactoring              | Sim   | Não   |
| Editor gráfico de layout  | Sim   | Sim   |
| Assinatura de APK e gerenciamento de keystore                     | Sim   | Sim   |
| Supporte à NDK  | Em Breve  | Sim   |

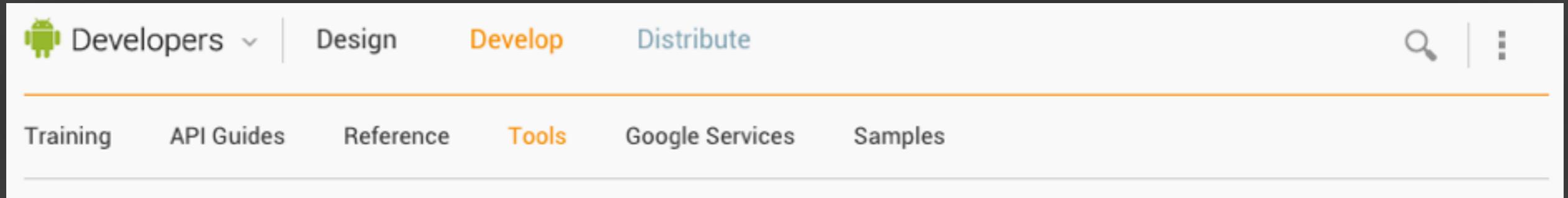
# Features do Android Studio

---

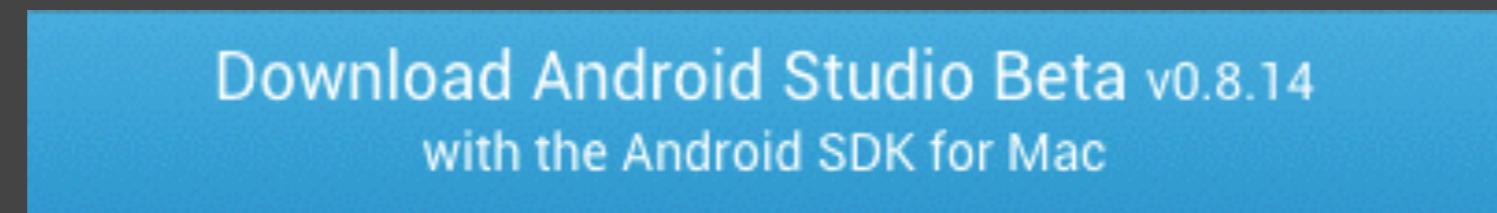
- Baseado no **flexível** sistema de build **Gradle**
- **Variantes** de **build** e **geração** de **múltiplos APKs**
- Suporte **expandido** de **templates** para **Google Play Services** e **vários** tipos de **dispositivos**
- **Editor** de **layout poderoso** com suporte para **edição** de **temas**
- Ferramenta **Lint** para **capturar problemas** de performance, usabilidade, compatibilidade de versões, dentre outros
- suporte para utilização do **ProGuard** e **assinatura** de **apps**
- **suporte** embutido para **Google Cloud Platform**, tornando mais fácil integrar funcionalidades do **Google Cloud Messaging** e **App Engine**

# Fazendo download do Android Studio

1 Acesse o link [d.android.com/develop/](https://d.android.com/develop/) > vá na seção *Tools*



2 Clique no link *Get Android Studio Beta* e depois no botão *Download Android Studio Beta v<versao>*

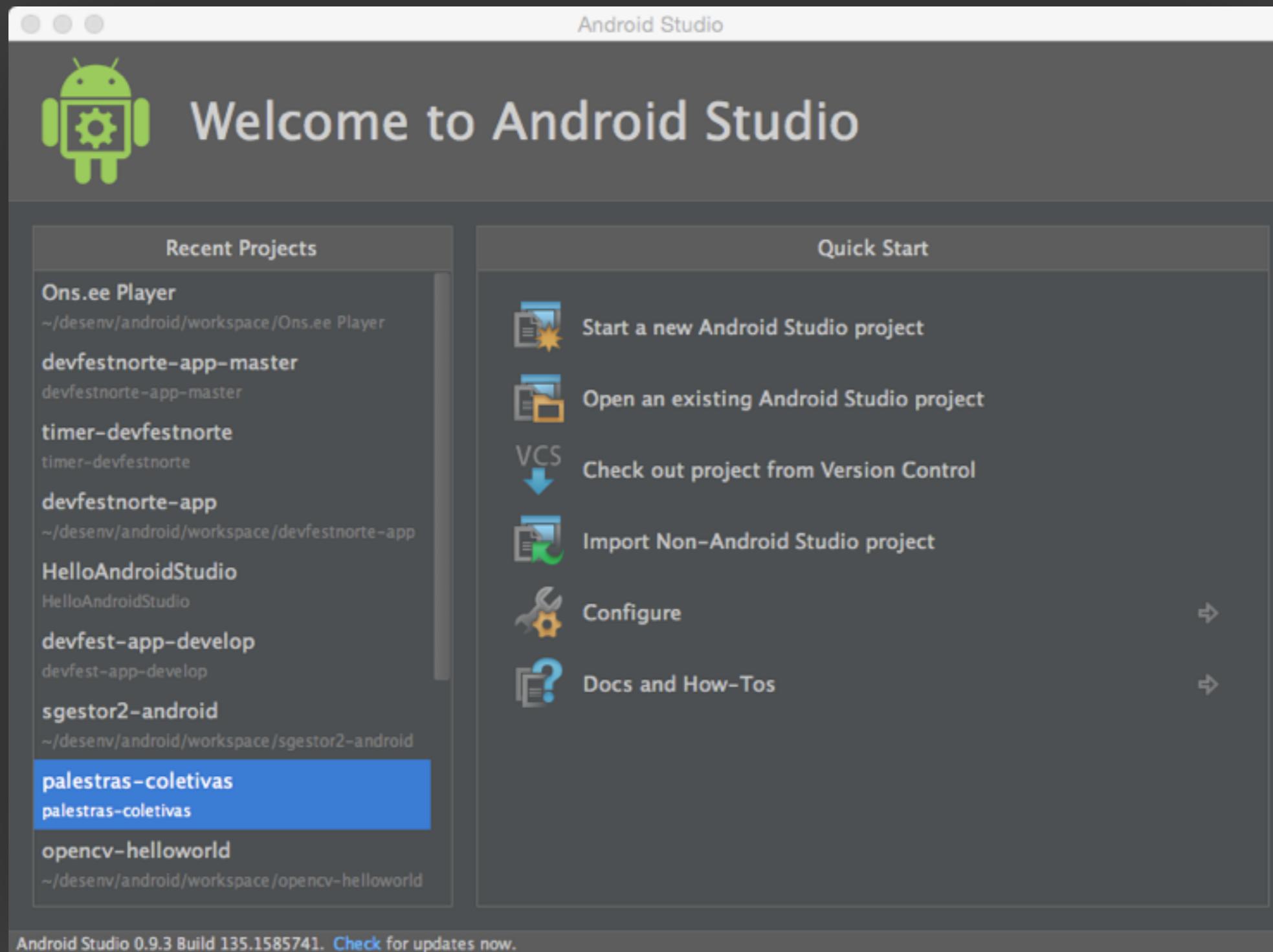


3 Aceite os termos e clique no botão de download. Pronto, agora é só esperar o término do download e instalar a ferramenta

# Criando projeto no Android Studio

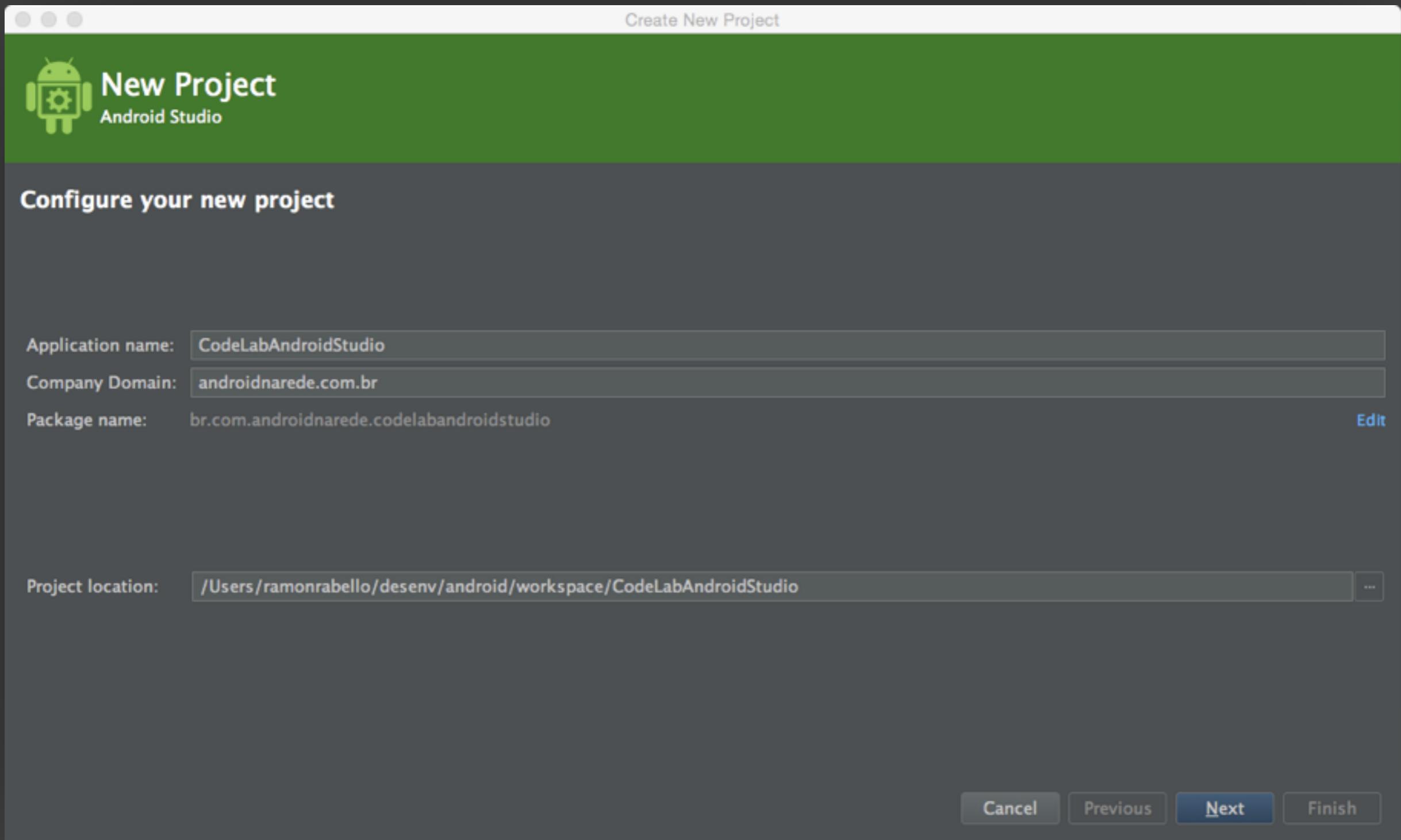
1

Ao abrir o Android Studio, clique no botão *Start a new Android Studio project*



# Criando projeto no Android Studio

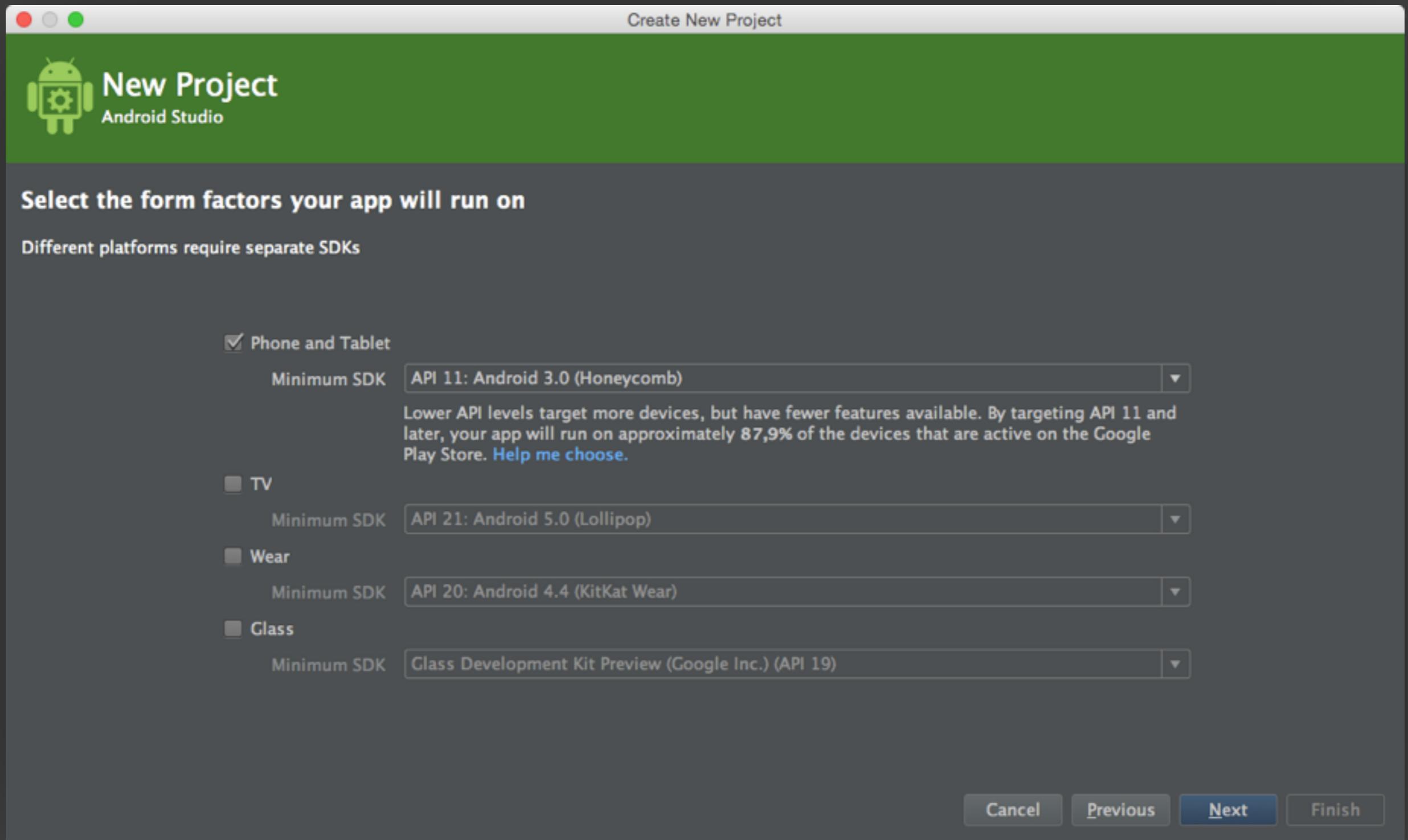
2 Na próxima tela, você irá configurar o seu projeto. Clique em *Next*



# Criando projeto no Android Studio

3

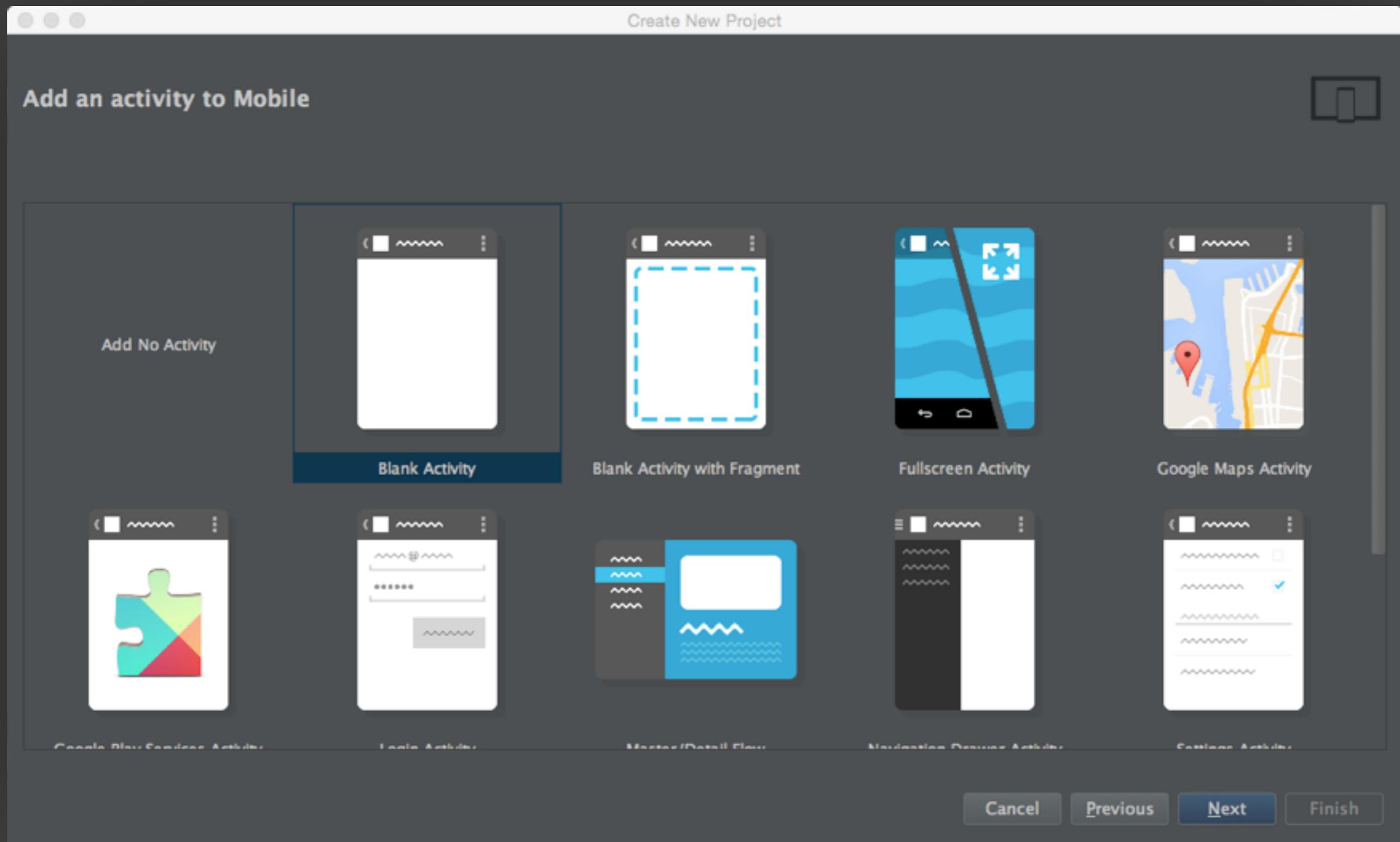
Agora você poderá escolher em quais os dispositivos sua app irá ser executada (TV, Wear, Glass). Clique em *Next*



# Criando projeto no Android Studio

4

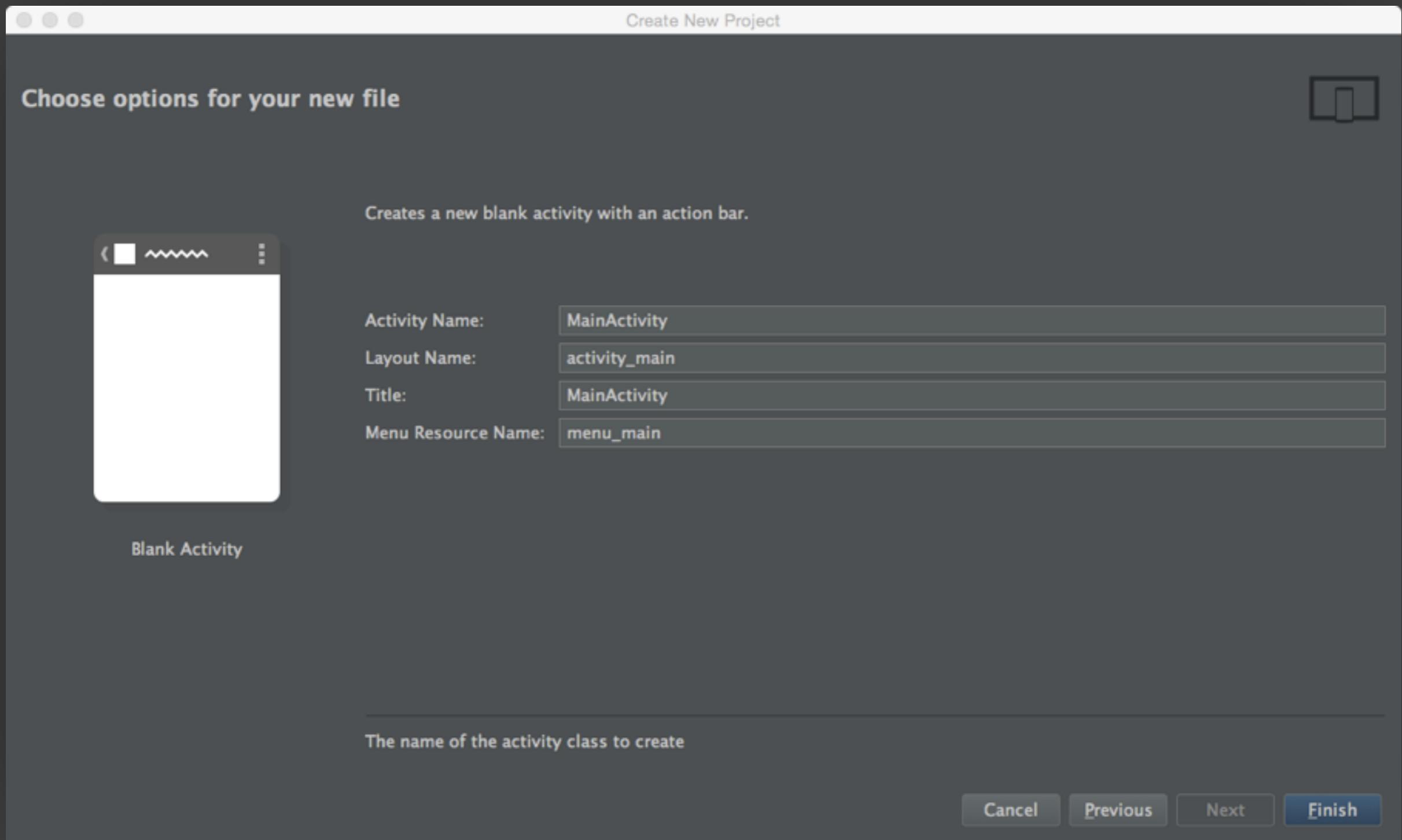
Nessa tela você poderá escolher o template da sua Activity.  
Depois de selecionar, clique em *Next* para prosseguir.



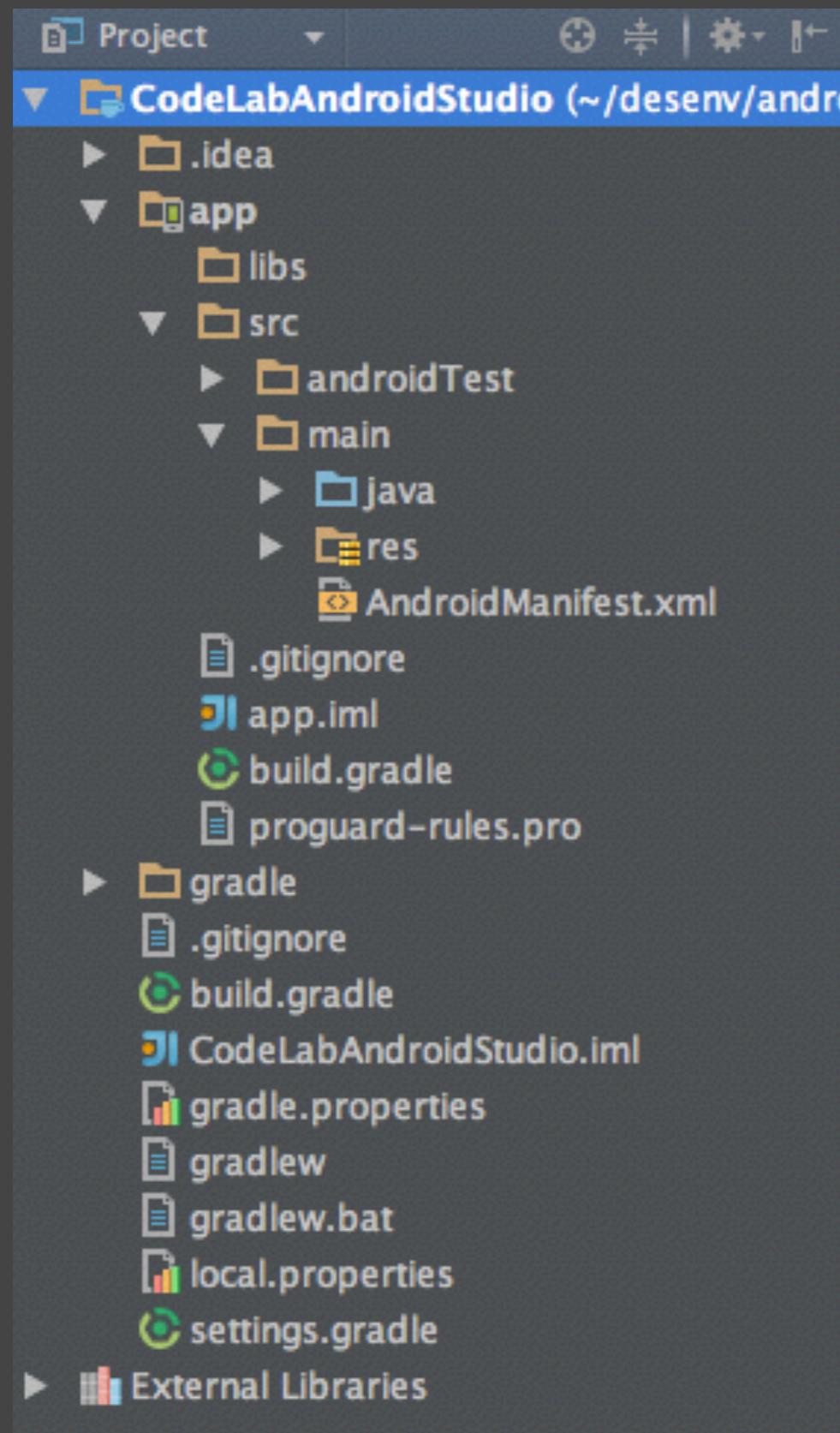
# Criando projeto no Android Studio

5

Agora você irá configurar as informações relacionadas à Activity, como nome, layout, título e menu. Para finalizar, clique em *Finish*



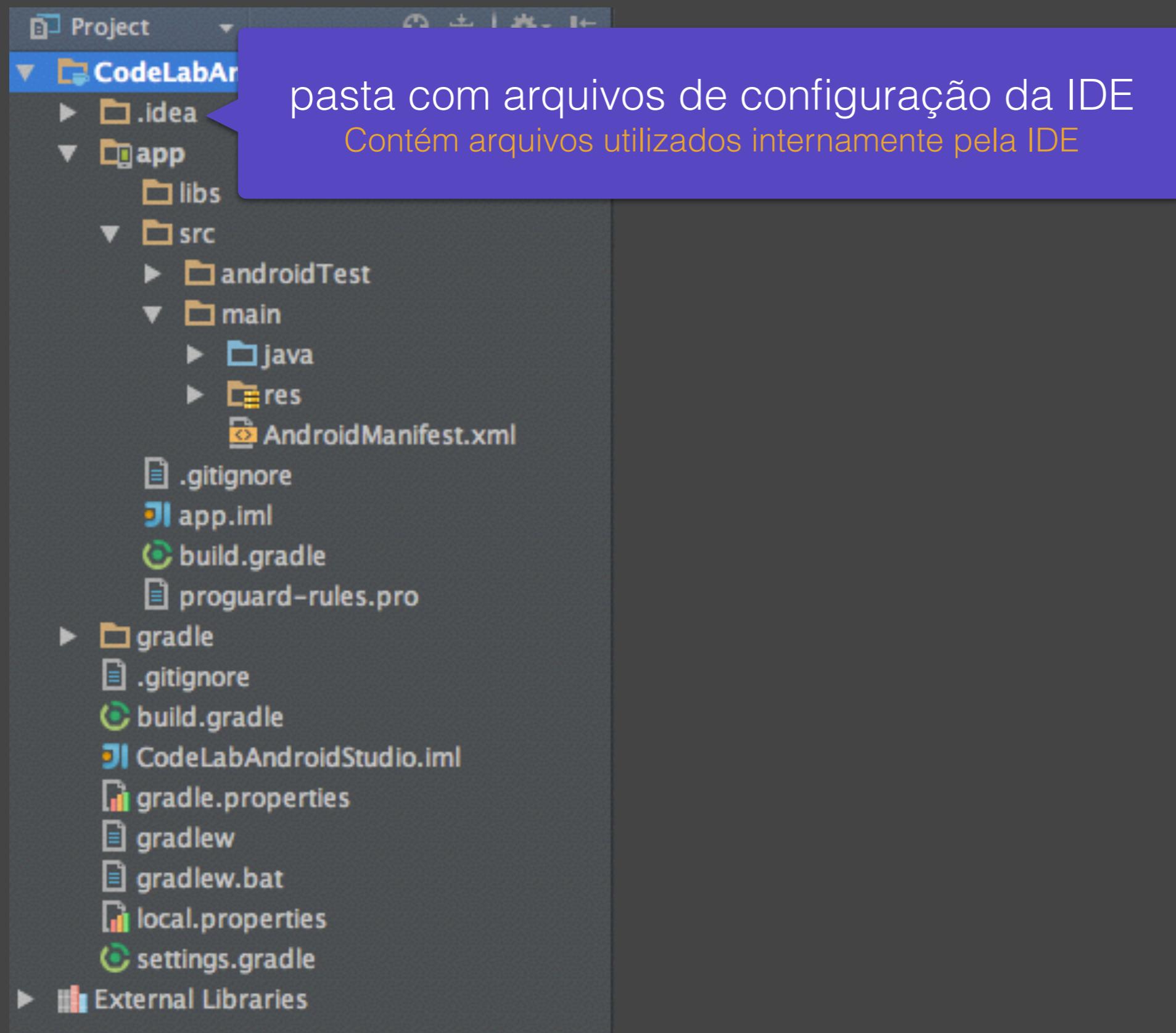
# Entendendo a estrutura de um projeto



Nome do Projeto

O nome do projeto (mesmo nome do diretório)

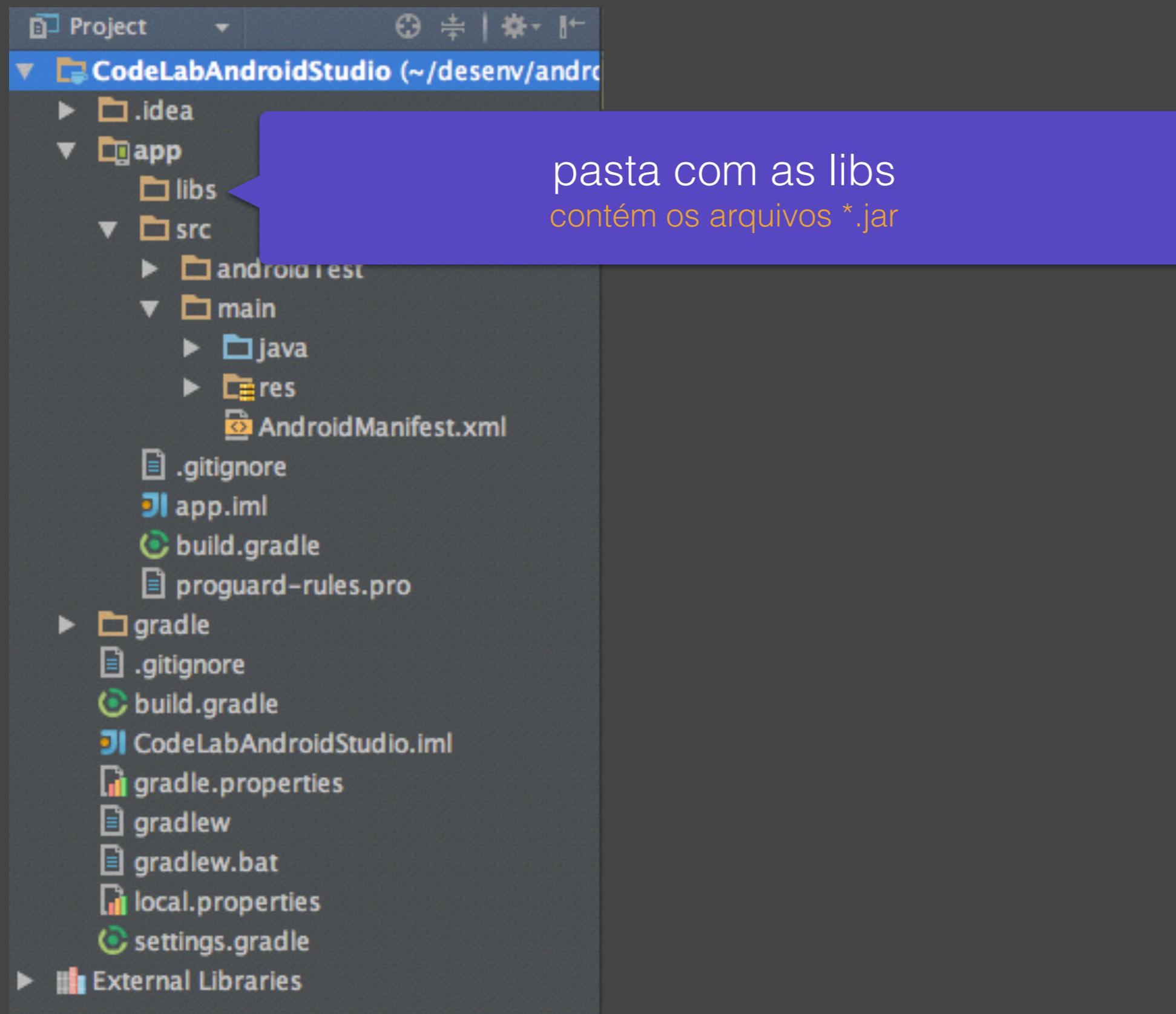
# Entendendo a estrutura de um projeto



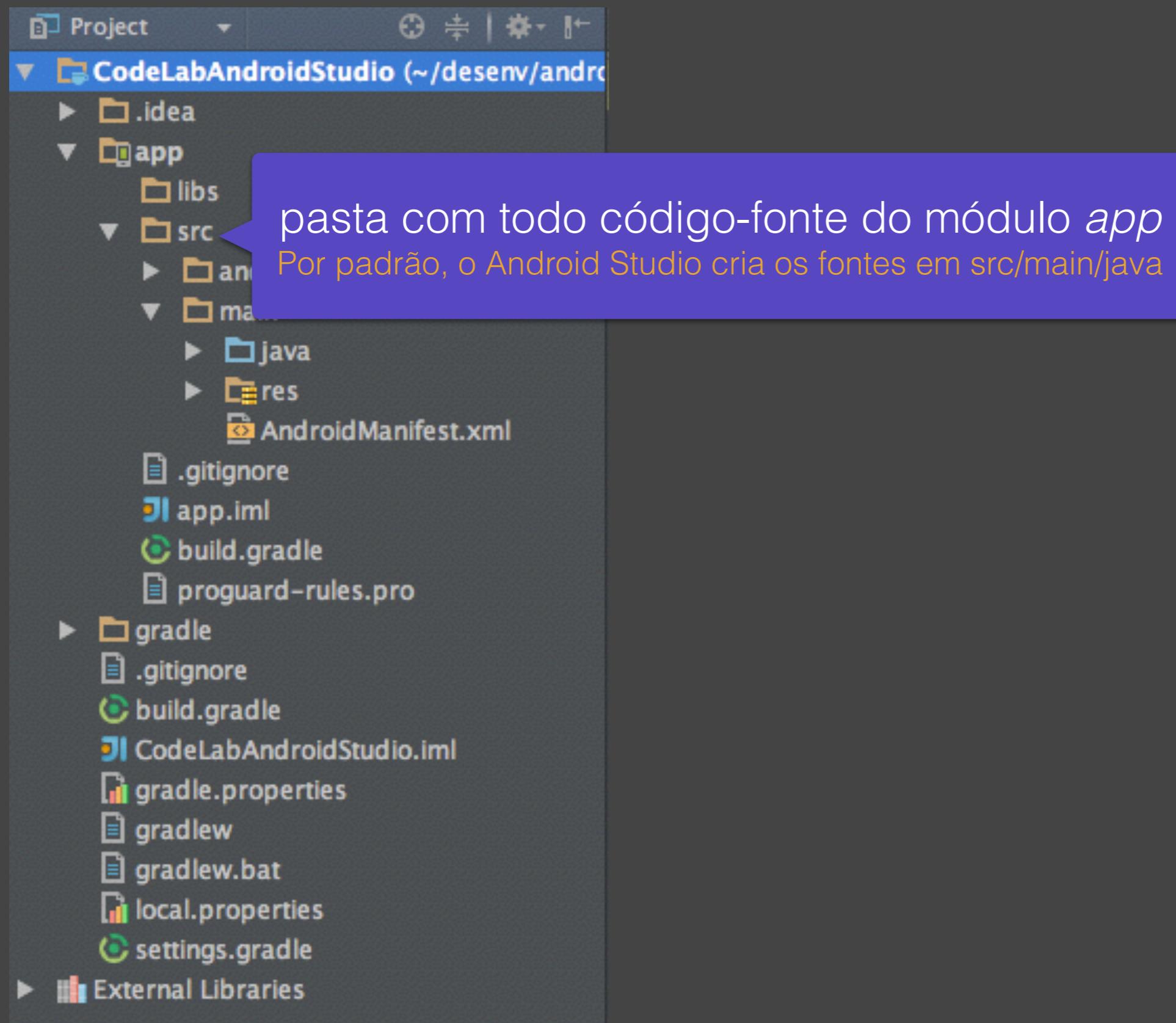
# Entendendo a estrutura de um projeto



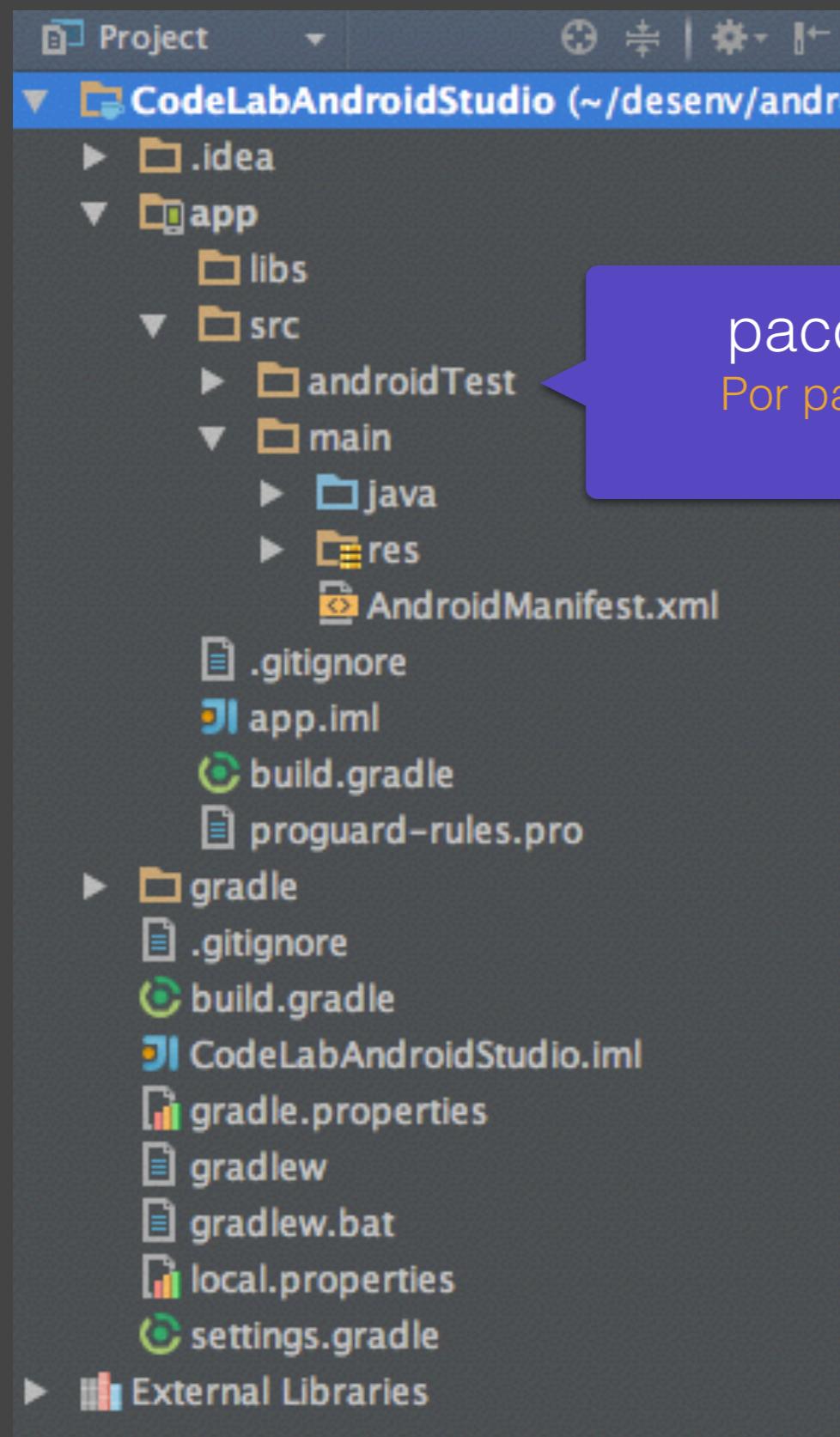
# Entendendo a estrutura de um projeto



# Entendendo a estrutura de um projeto

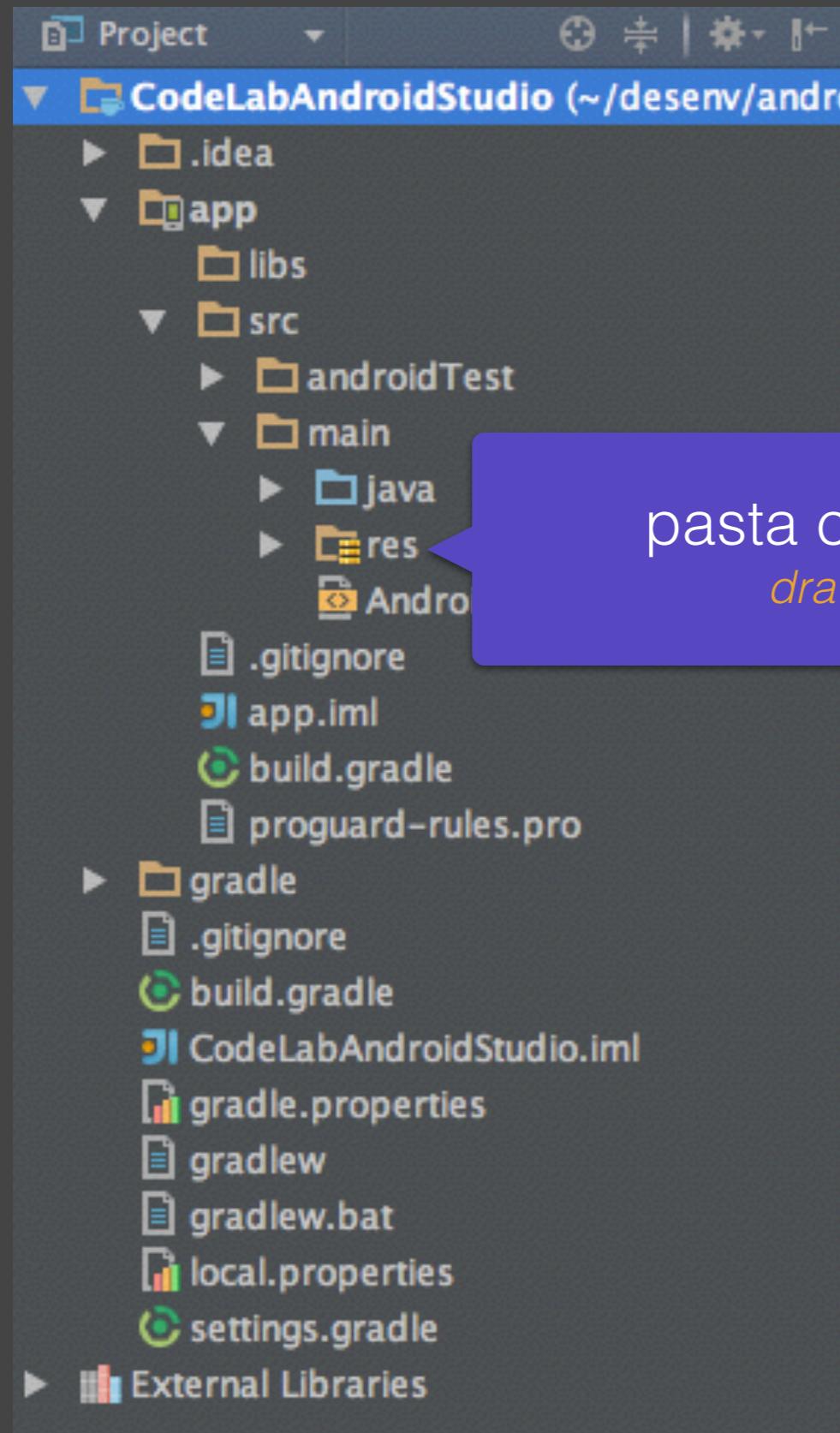


# Entendendo a estrutura de um projeto



pacote principal das classes de teste  
Por padrão, o Android Studio cria os fontes em src/  
androidTest/java

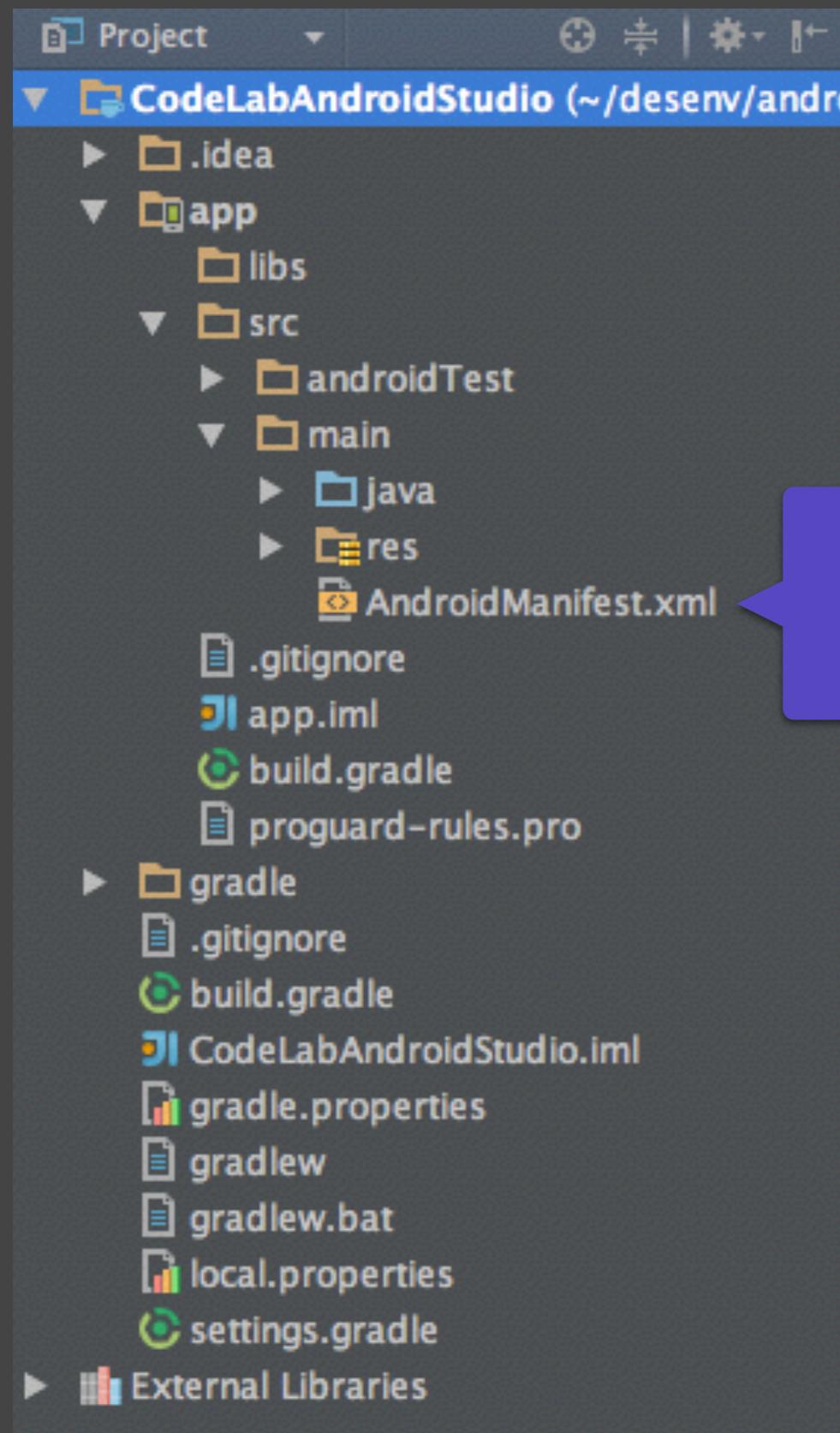
# Entendendo a estrutura de um projeto



pasta com os arquivos de recursos

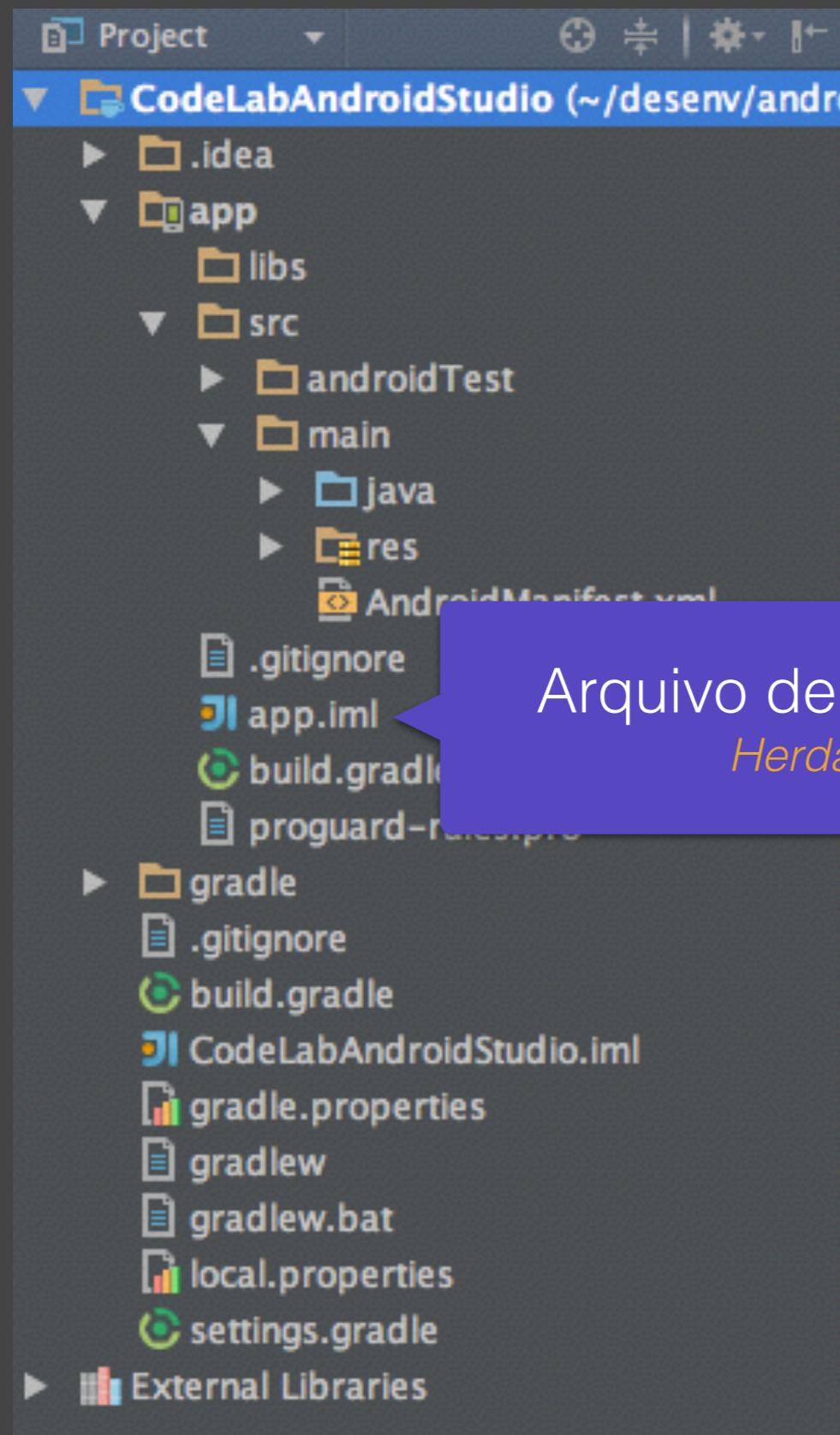
*drawables, xml, layouts, values, etc*

# Entendendo a estrutura de um projeto



Arquivo de Manifesto do módulo *app*  
*No Android Studio, existe um manifesto para cada módulo*

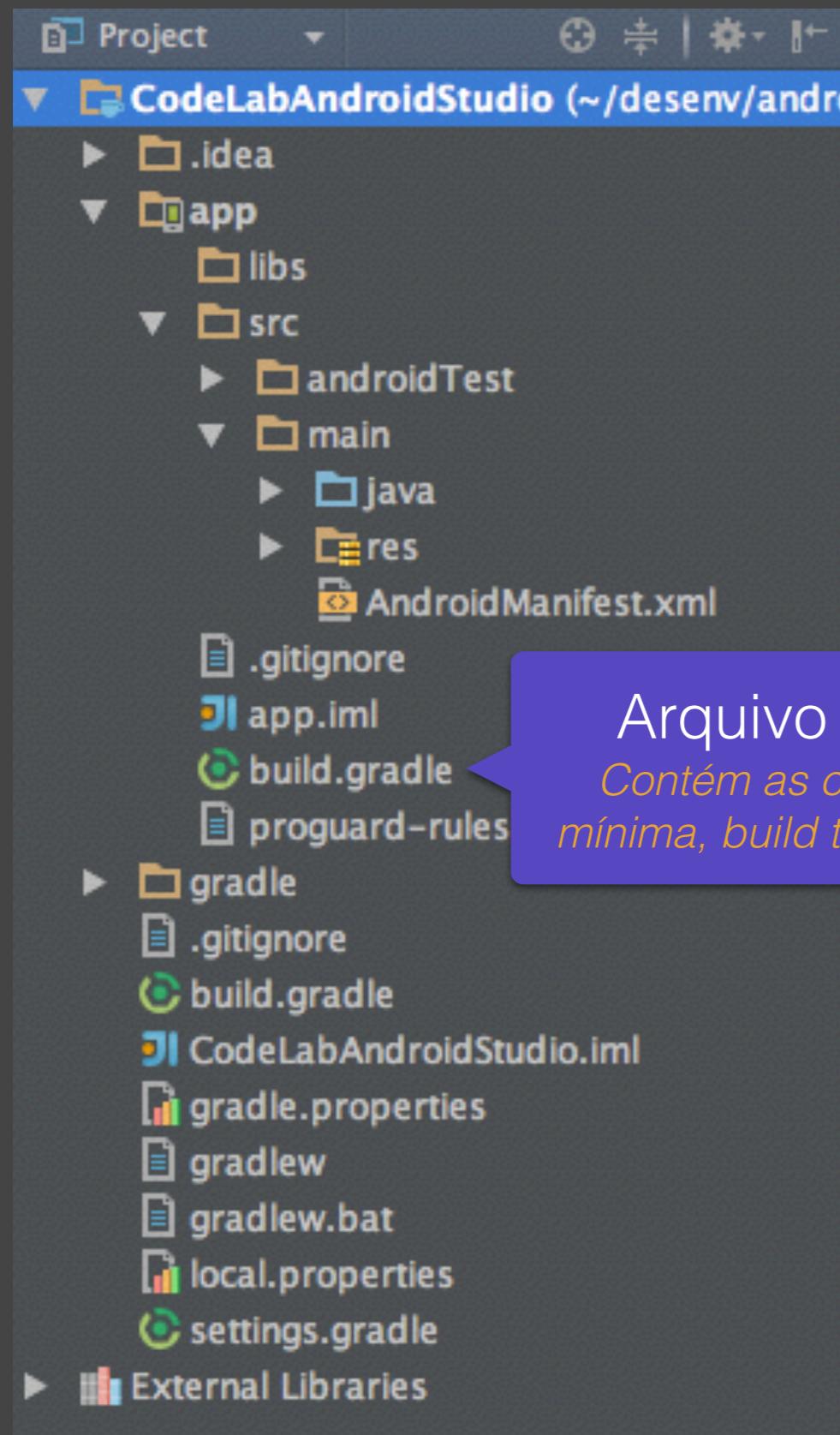
# Entendendo a estrutura de um projeto



Arquivo de configuração do módulo *app*

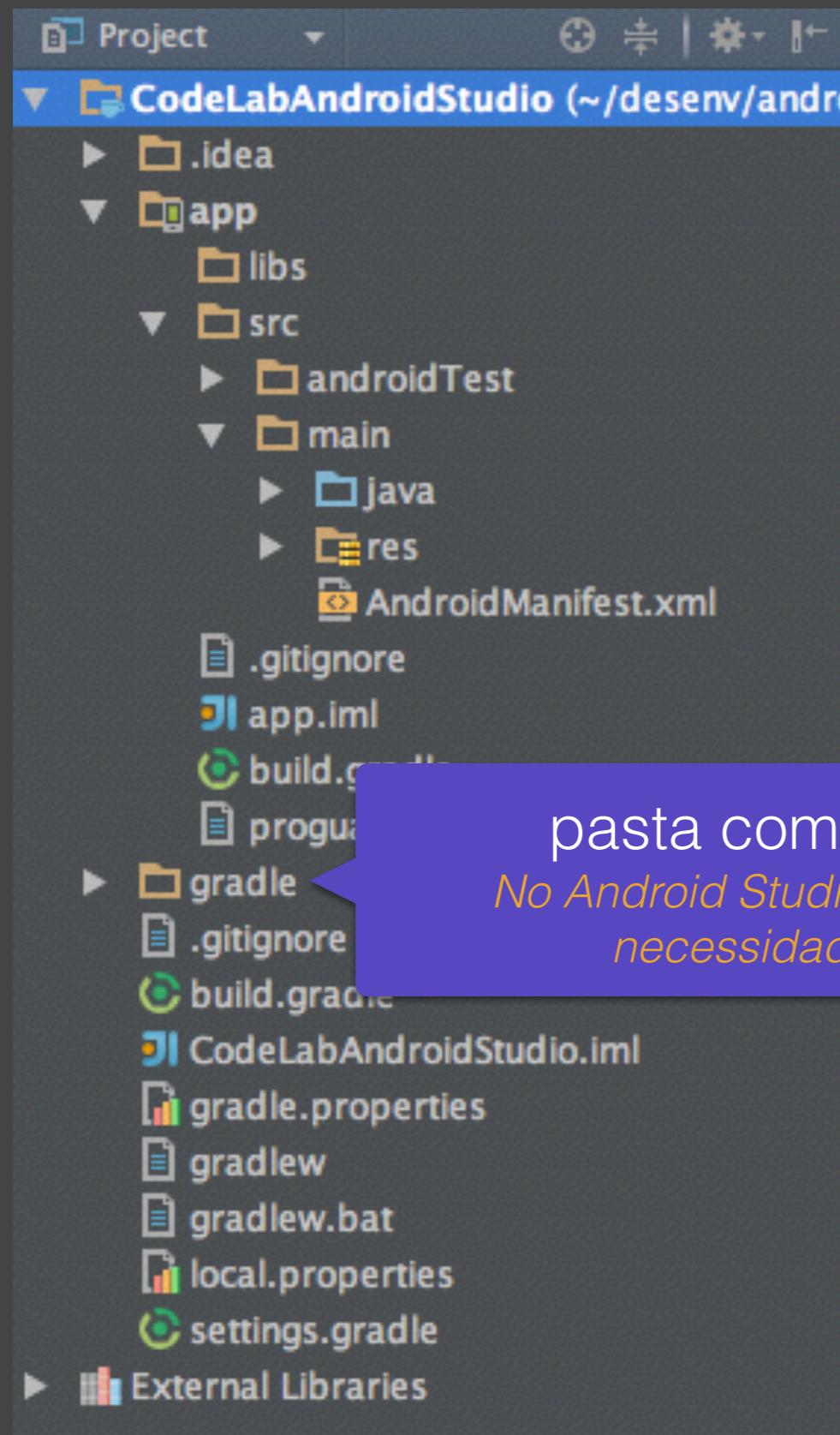
*Herdado da infraestrutura do IntelliJ*

# Entendendo a estrutura de um projeto



Arquivo `build.gradle` para o módulo `app`  
*Contém as configurações e tasks do módulo, como versão mínima, build types, empacotamento e outras DSLs em Groovy*

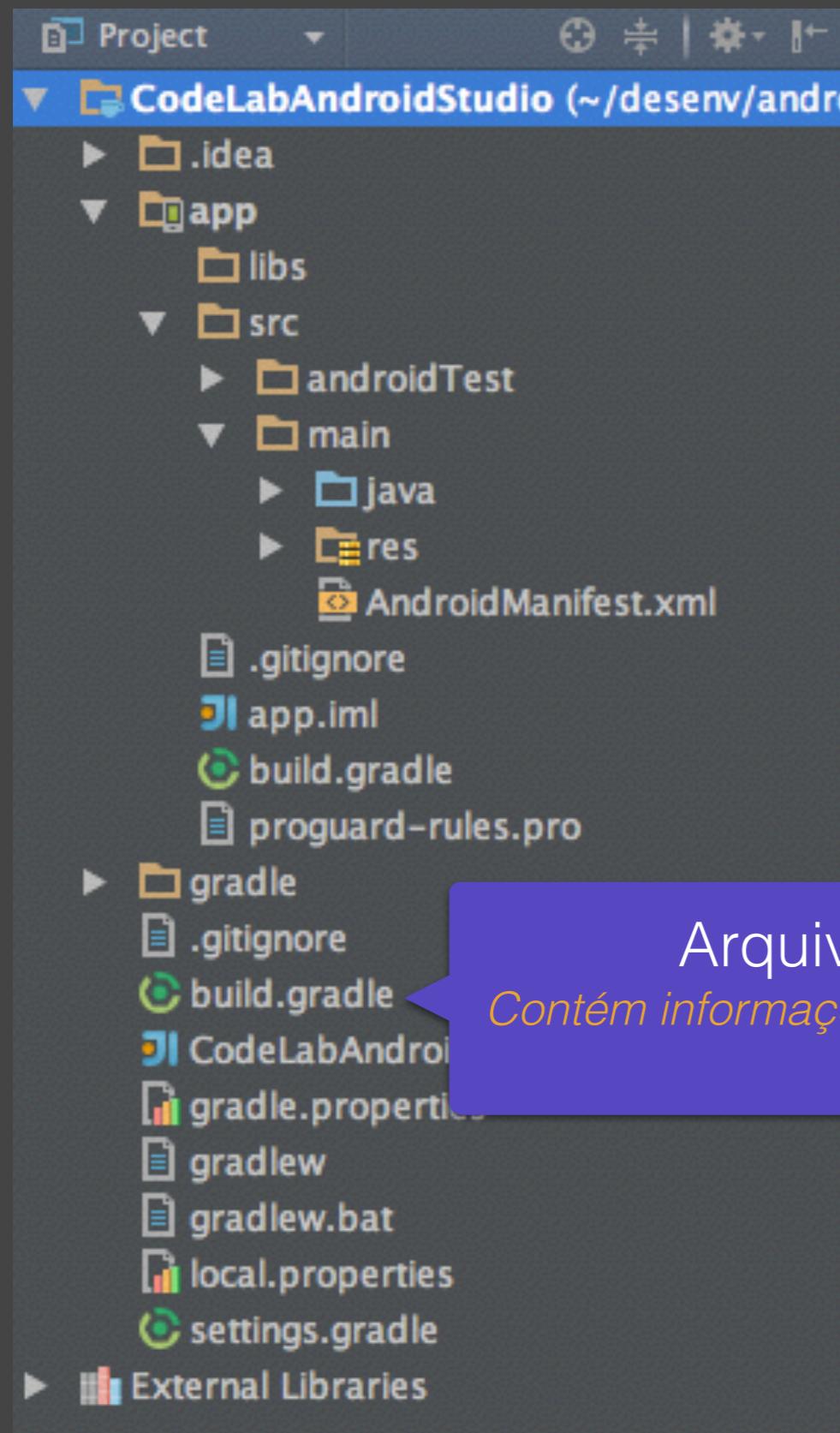
# Entendendo a estrutura de um projeto



pasta com o wrapper para gradle

No *Android Studio*, o *Gradle* já vem *embutido*, sem  
necessidade de utilizá-lo *externamente*

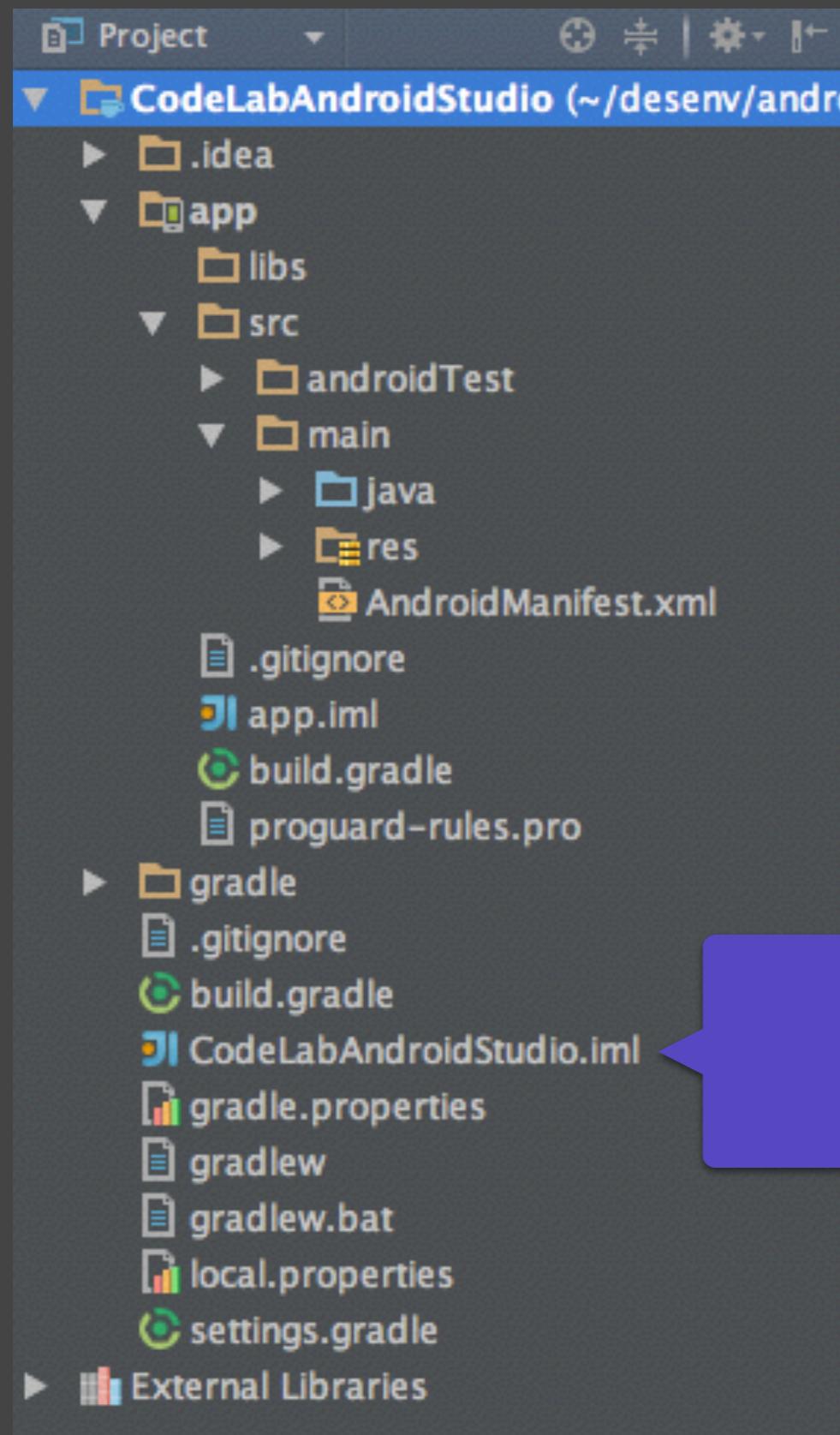
# Entendendo a estrutura de um projeto



Arquivo `build.gradle` do projeto

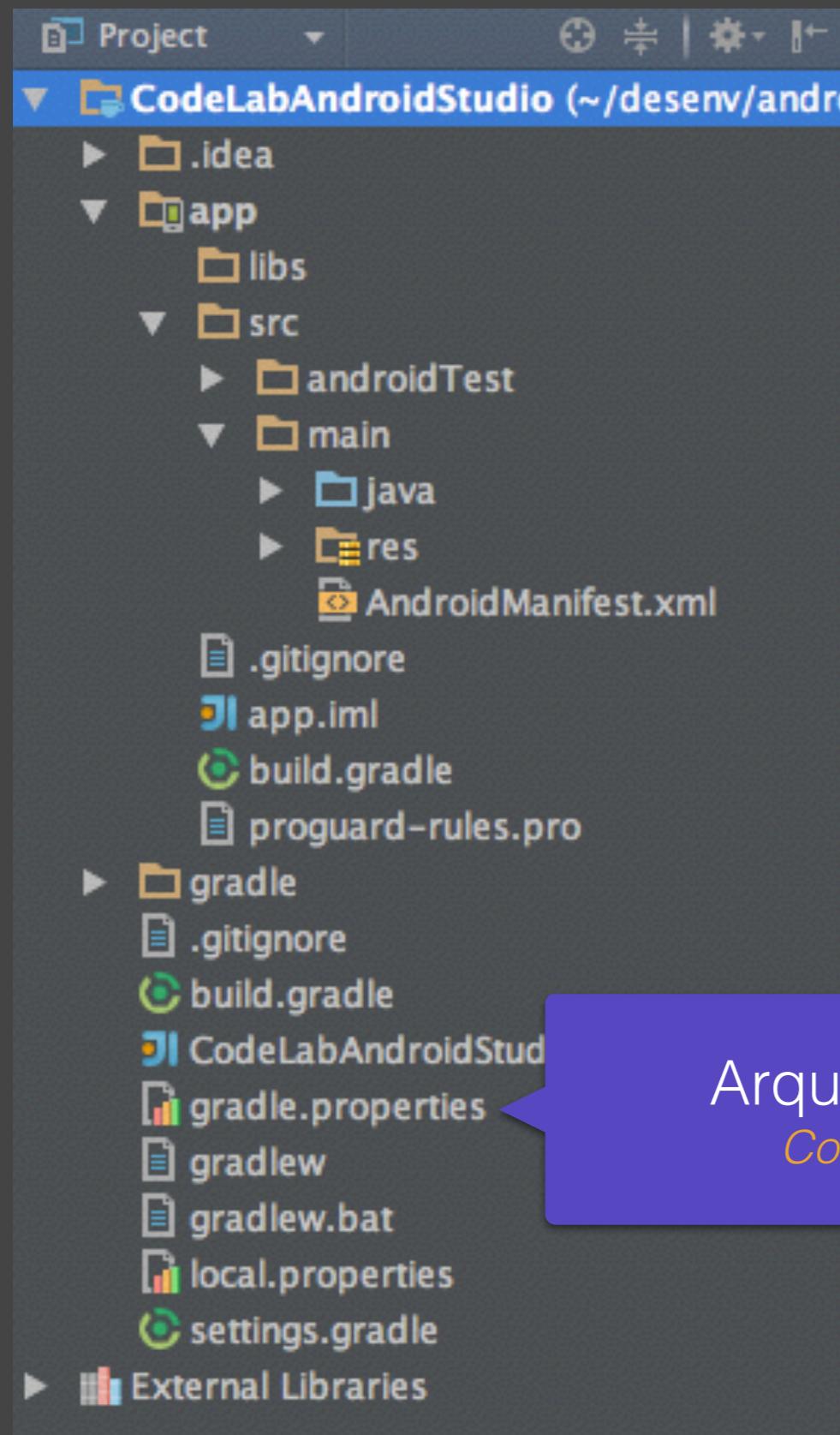
*Contém informações gerais do projeto, como dependências, repositórios, etc*

# Entendendo a estrutura de um projeto



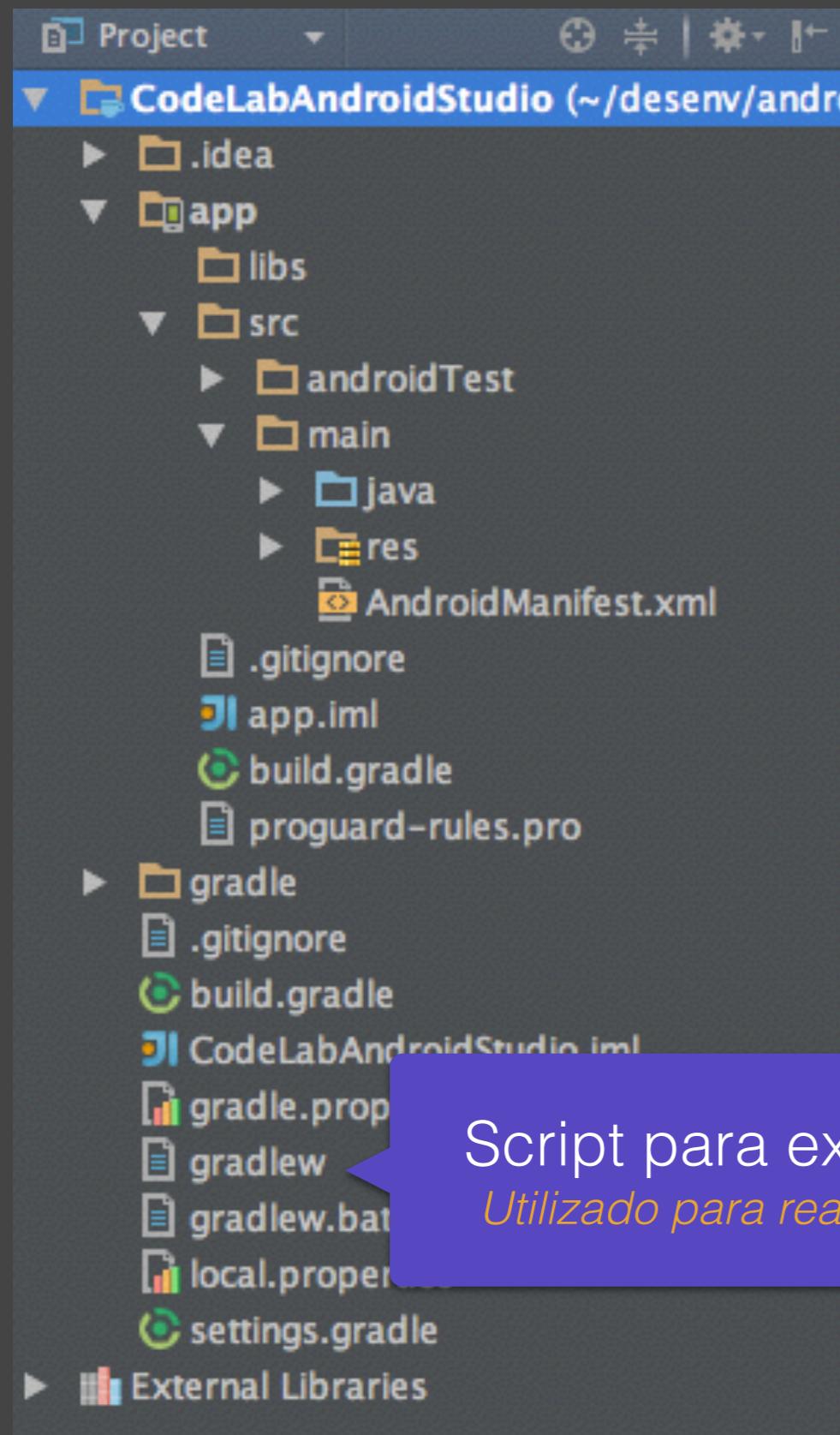
Arquivo de configuração do projeto  
*Herdado da infraestrutura do IntelliJ*

# Entendendo a estrutura de um projeto



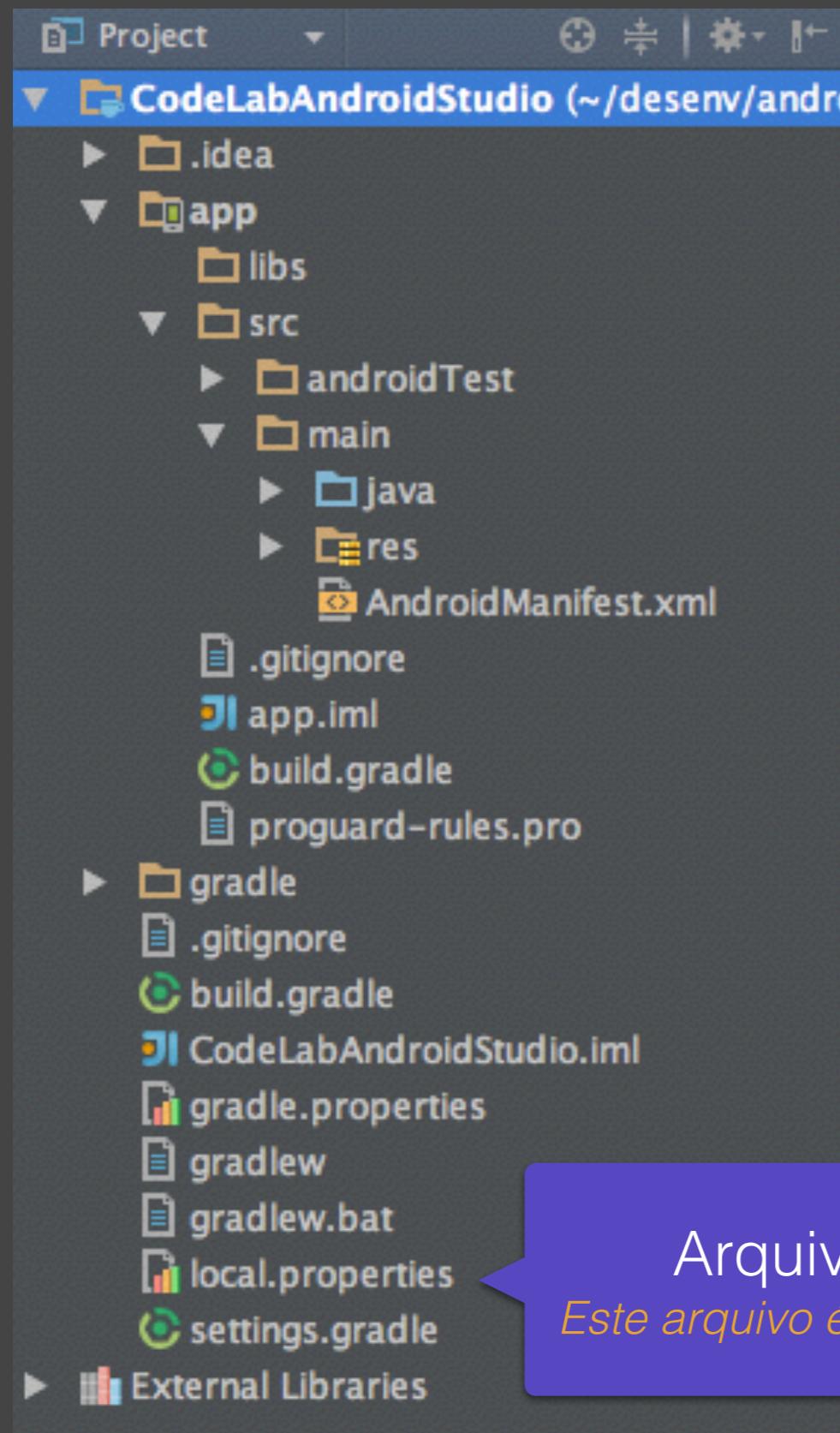
Arquivo de propriedades do Gradle  
*Configurações específicas para o Gradle*

# Entendendo a estrutura de um projeto



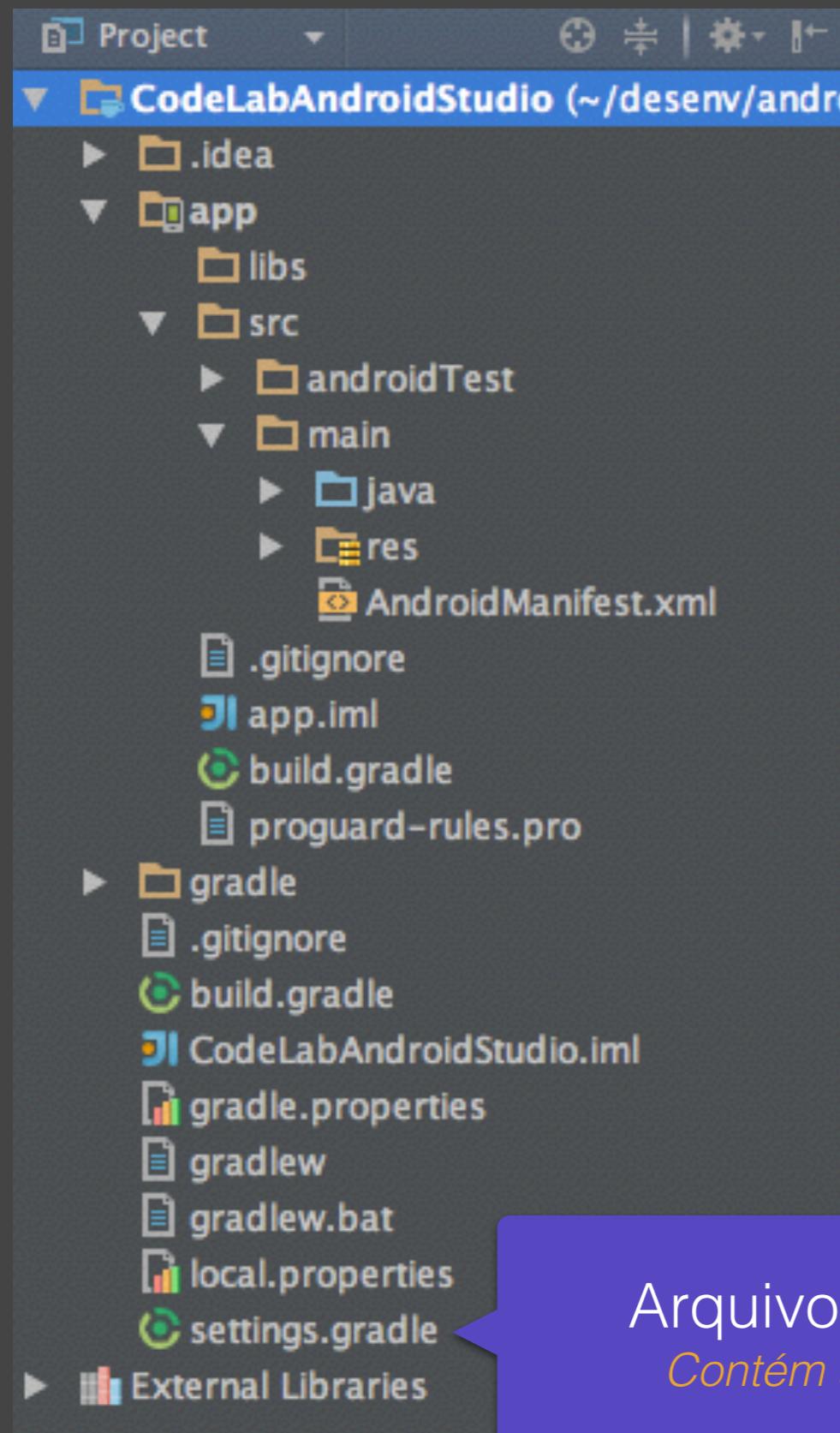
Script para execução das tasks do Gradle  
*Utilizado para realização do processo de build de apps*

# Entendendo a estrutura de um projeto



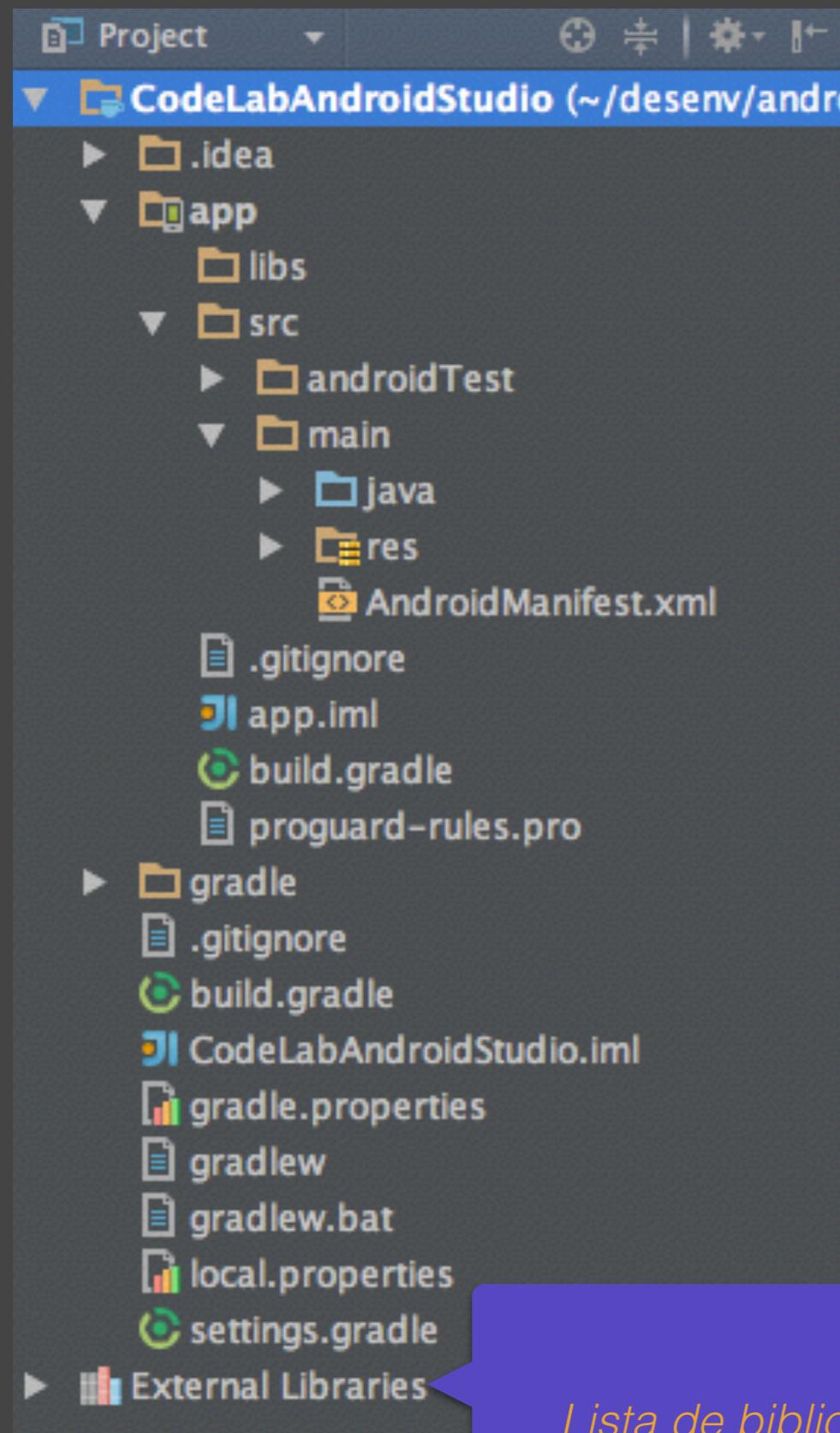
Arquivo de informações locais (SDK)  
Este arquivo é gerado pelo Android Studio. Não o modifique!

# Entendendo a estrutura de um projeto



Arquivo de configurações de módulos  
*Contém as informações dos módulos do projeto*

# Entendendo a estrutura de um projeto



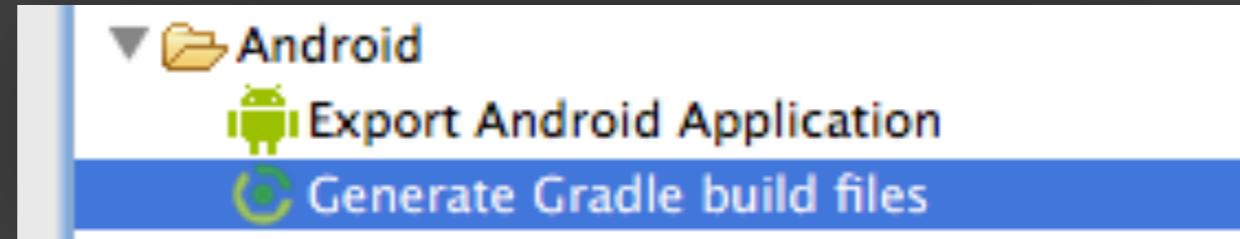
Bibliotecas externas

*Lista de bibliotecas externas ao projeto (SDK, JDK, etc)*

# Exportando projetos do ADT

1

Para importar o seu projeto do ADT no Android Studio, primeiramente você terá que exportá-lo. Esse processo irá gerar os arquivos build.gradle (para cada módulo) e o settings.gradle. Para isso, no ADT, aponte para *File > Export... > Android > Generate Gradle build files*

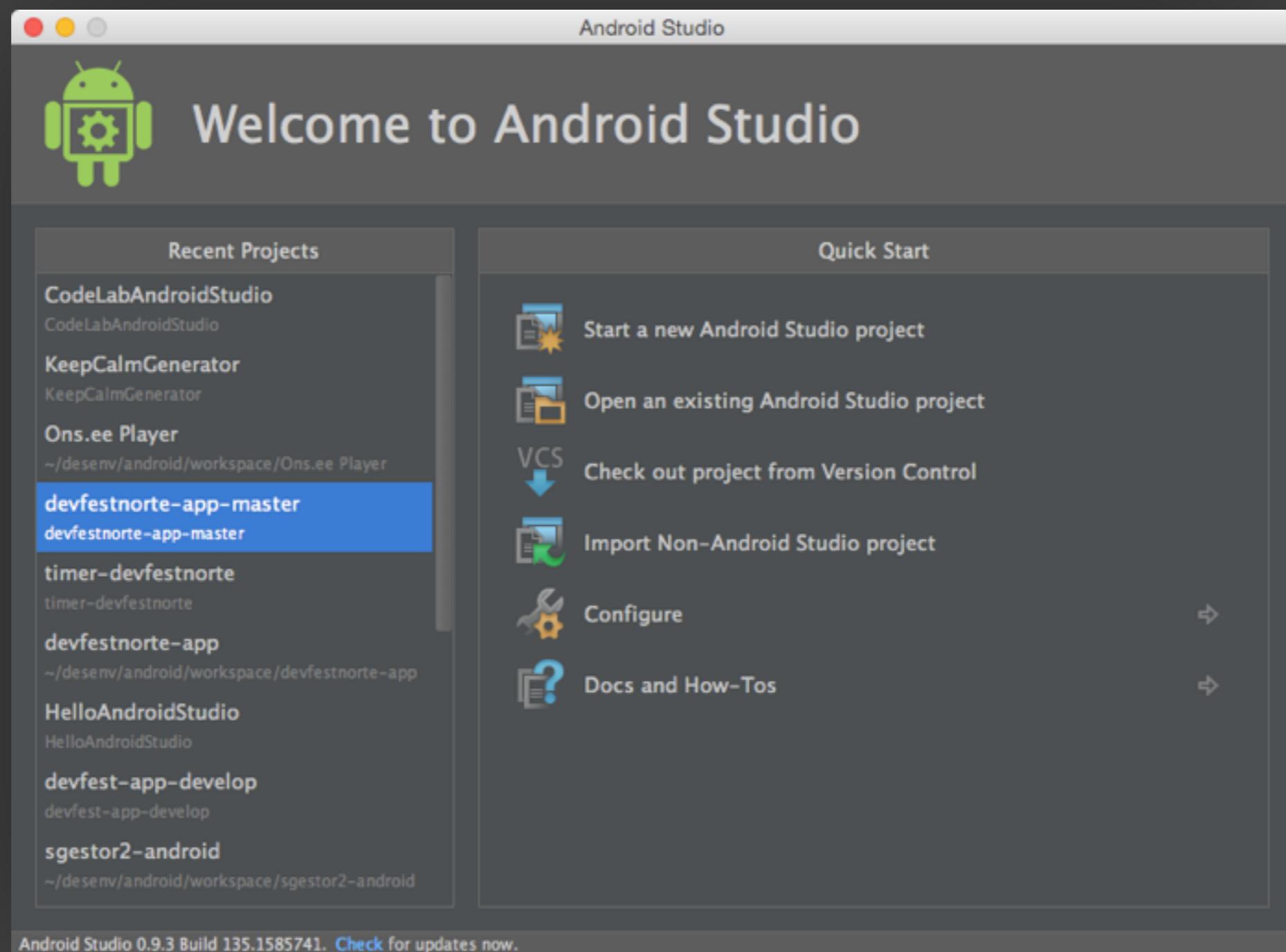


Esse processo irá alterar toda estrutura do seu projeto Eclipse antigo. Por isso, quem quiser mantê-lo funcionando no ADT, recomenda-se copiar todo o projeto para uma nova pasta antes de migrar.

# Importando seus projetos do ADT no Android Studio

1

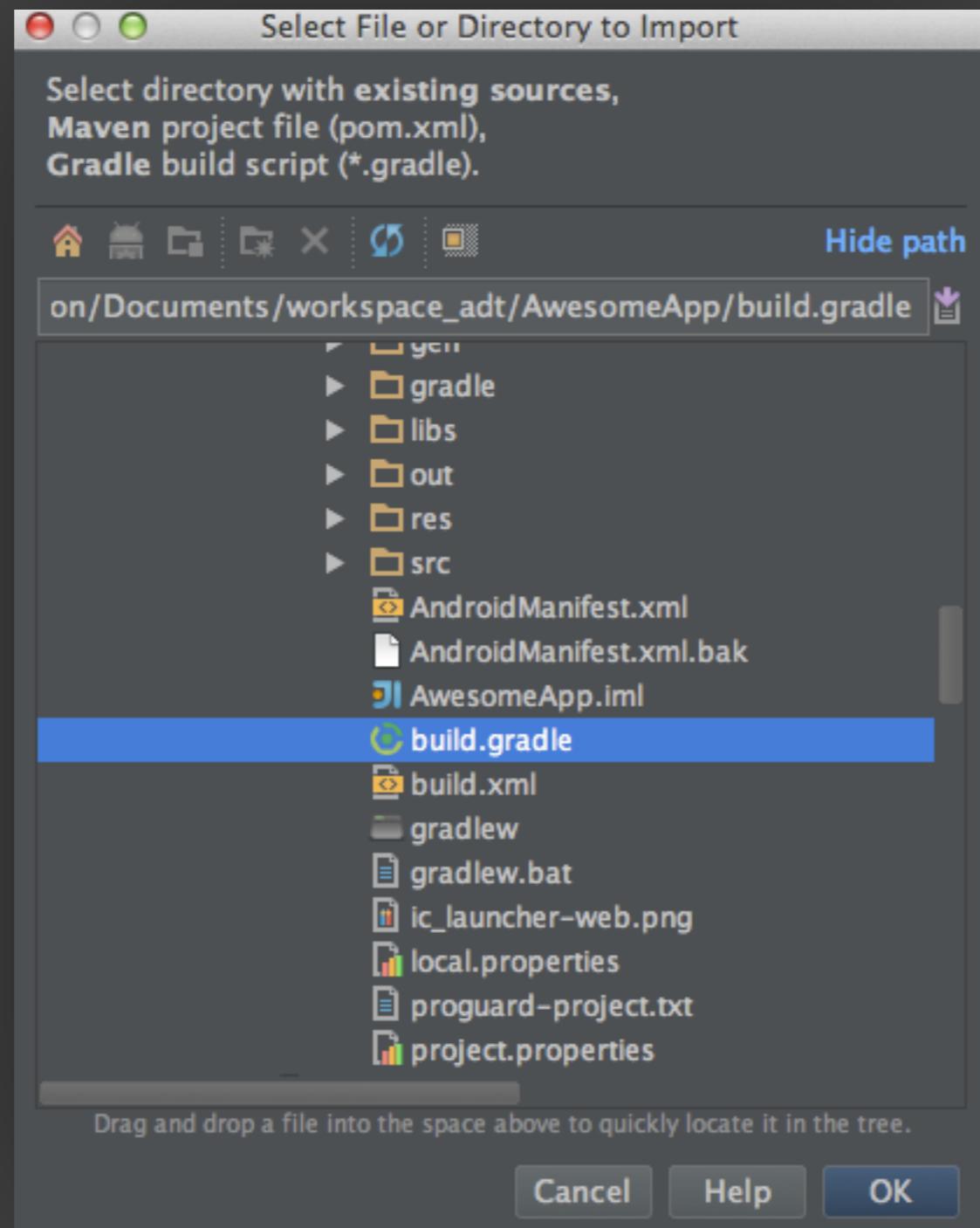
Na tela inicial do Android Studio (feche todos os projetos caso não esteja visualizando) clique em *Import Non-Android Studio Project*



# Importando seus projetos do ADT no Android Studio

2

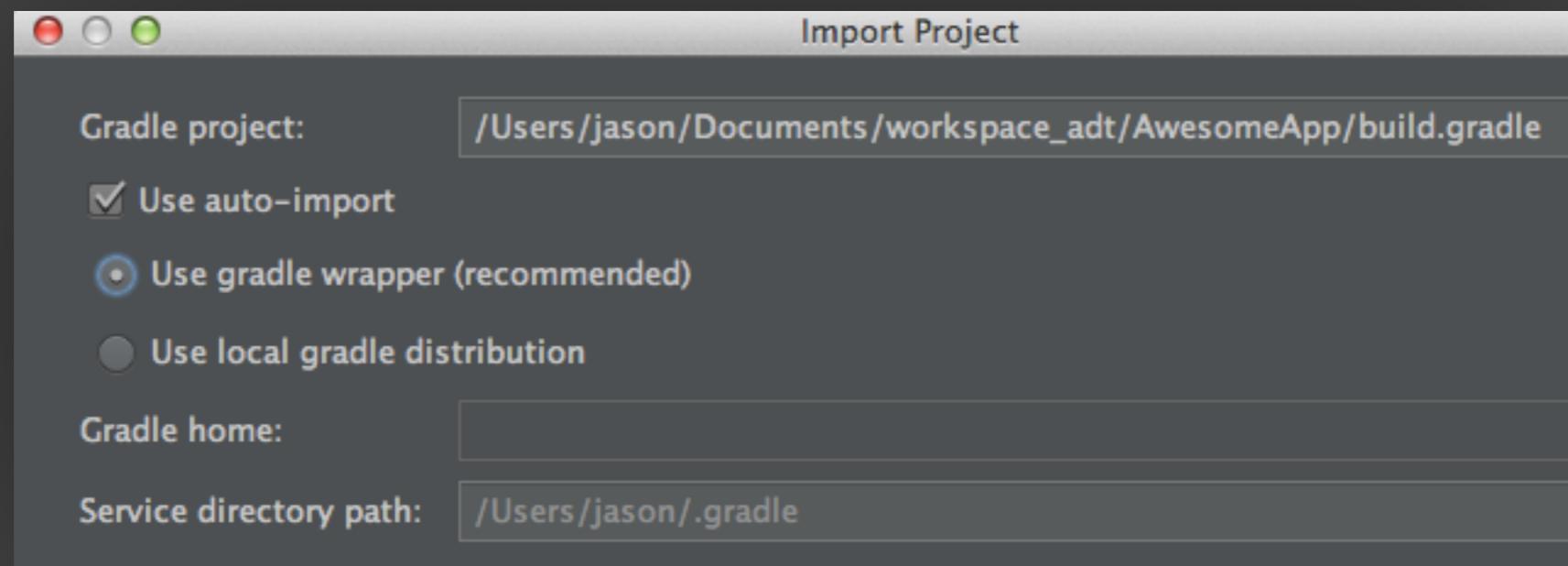
Na tela a seguir, navegue até a pasta do projeto ADT, selecione a pasta do projeto ADT, o arquivo *build.gradle* ou o arquivo *settings.gradle*. Depois clique em *OK*.



# Importando seus projetos do ADT no Android Studio

3

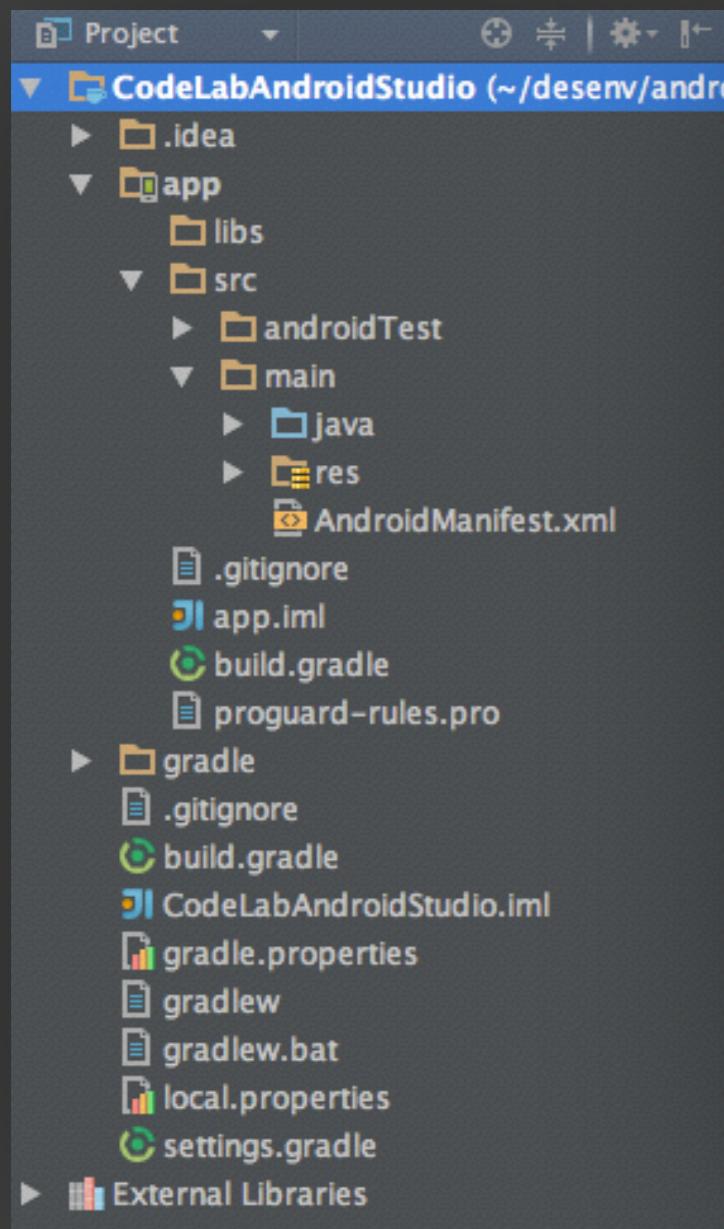
Depois, marque a opção Use auto-import e selecione “Use grade wrapper” para utilizar o Gradle que já vem embutido no Android Studio. Para finalizar a migração, clique em *Finish*.



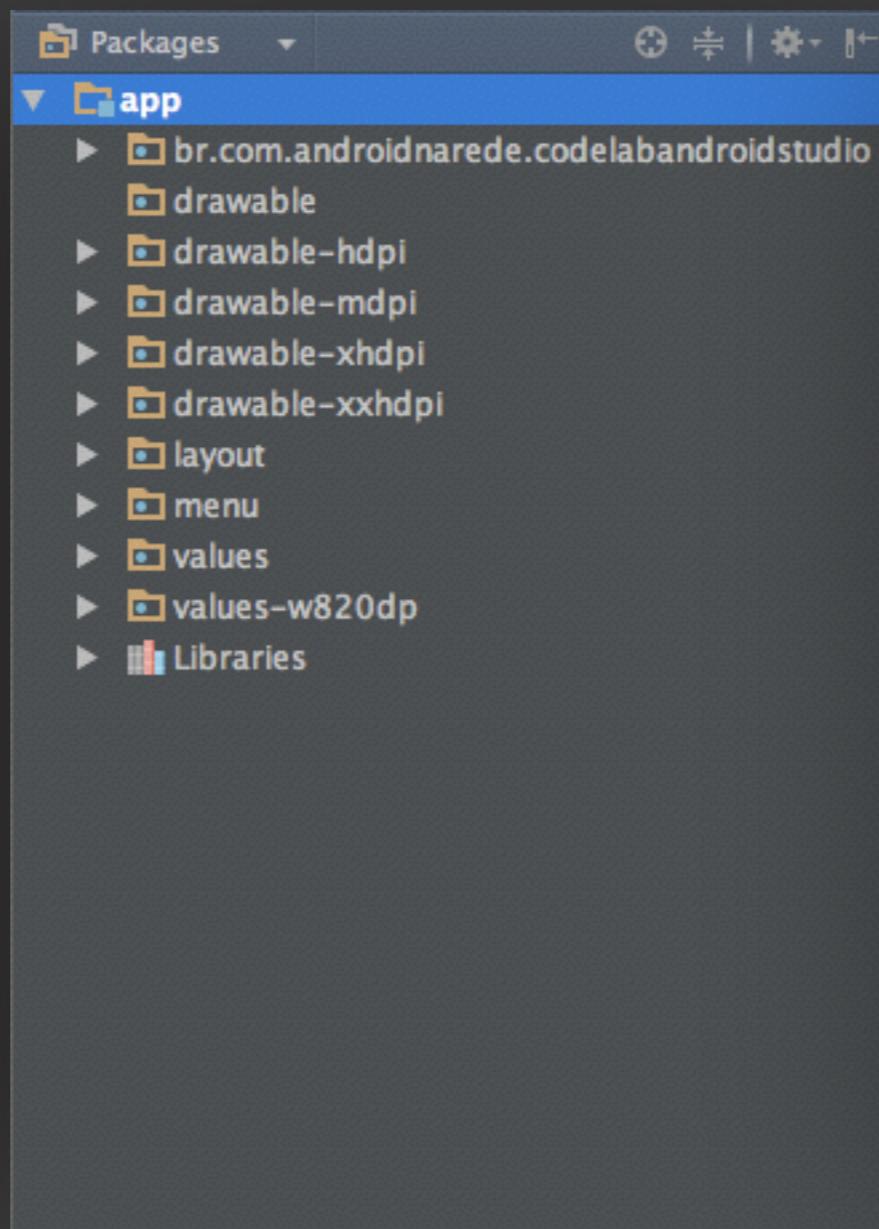
[Este link](#) mostra como é possível manter a compatibilidade do mesmo projeto sendo utilizado tanto no Android Studio quanto no Eclipse.

# Visões no Android Studio

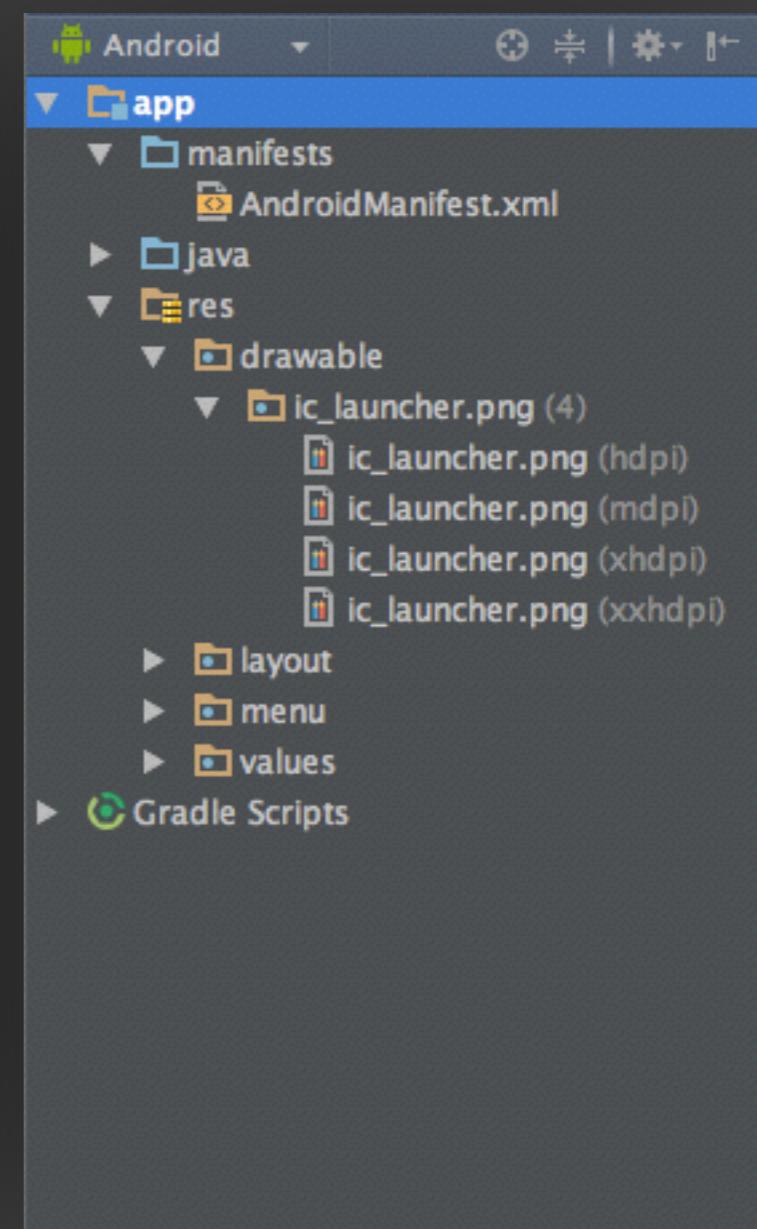
## Projeto



## Pacotes



## Android



# Apresentando Gradle

---



**Gradle** é um **sistema** de **build** ideal para **automatizar** builds, testes, publicações, deploy, empacotamentos e muito mais!

É o novo sistema de build utilizado pelo Android Studio, pois suporta várias características como **módulos**, **build variants**, **product flavors**, etc



Utiliza **Groovy** em sua essência para **criação** e **uso** de **DSLs**

# O arquivo build.gradle (geral)

---

```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:0.14.2'  
    }  
}  
  
allprojects {  
    repositories {  
        jcenter()  
    }  
}
```

# O arquivo build.gradle (módulo)

---

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "20.0.0"

    defaultConfig {
        applicationId "br.com.androidnarede.codelabandroidstudio"
        minSdkVersion 19
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }

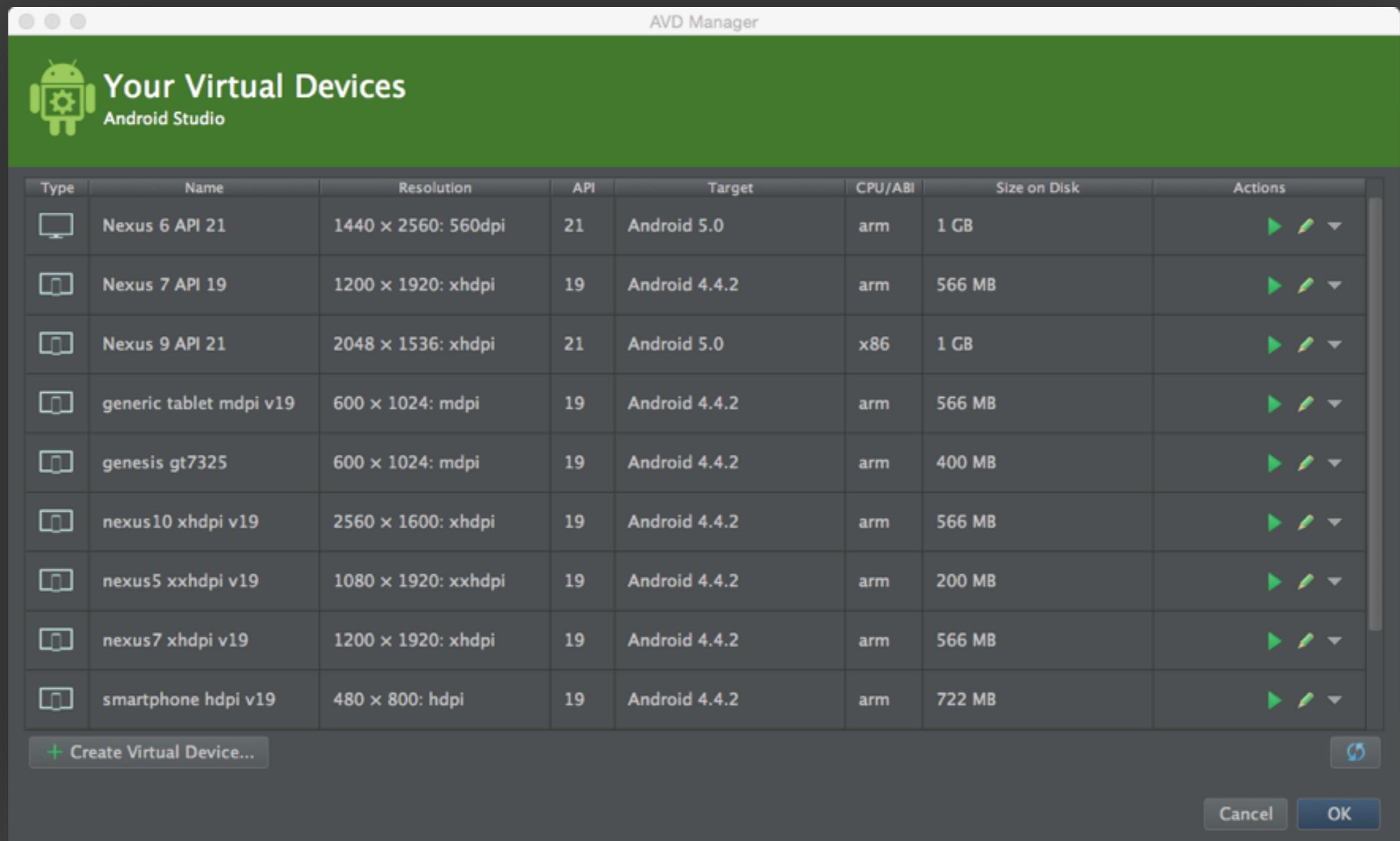
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.2'
}
```

# Executando o projeto no emulador

1

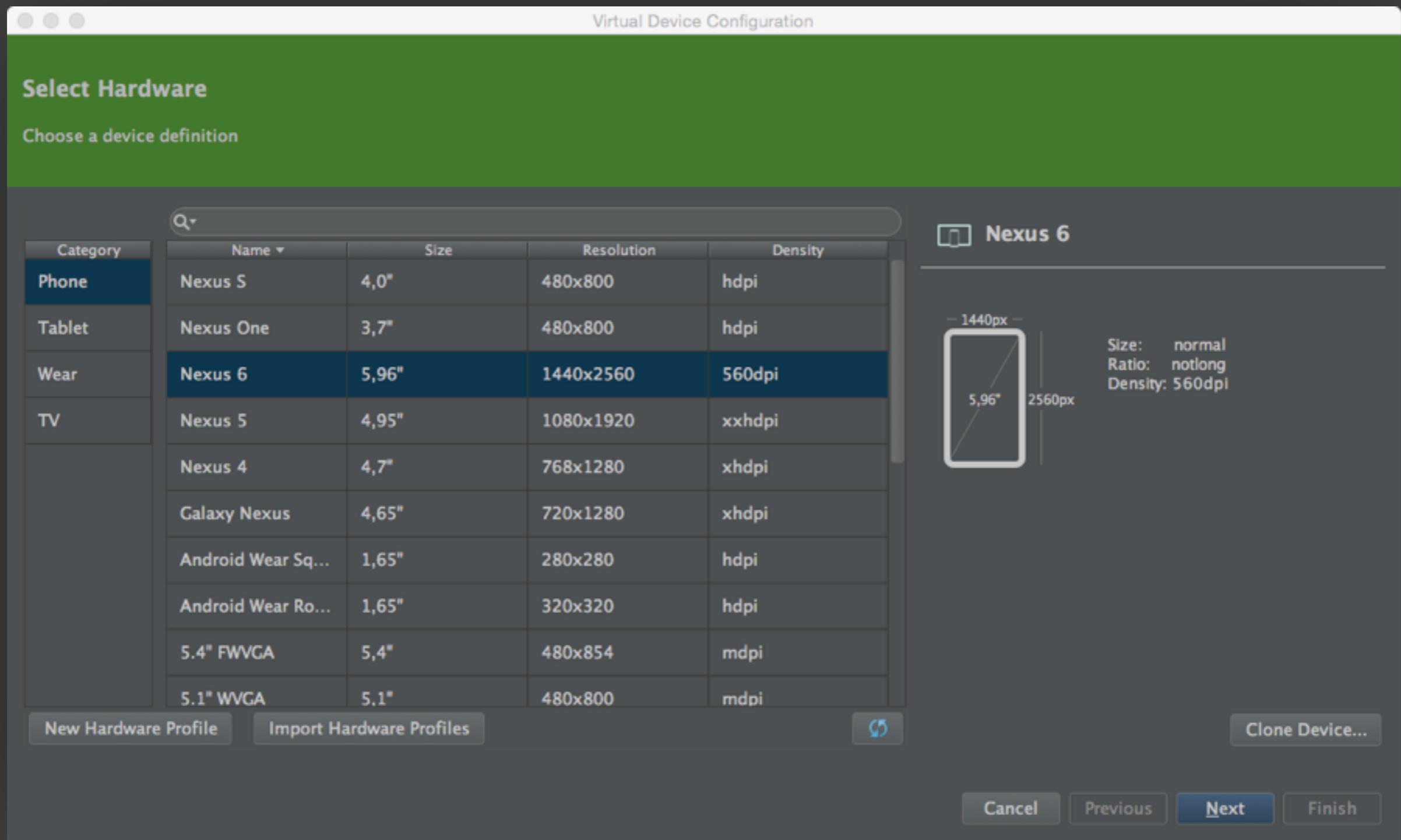
Para executar no emulador, primeiramente você terá que ter um AVD criado. Para isso, vá em *Tools > Android > AVD Manager*. Na tela que aparecer, clique no botão *Create Virtual Device...*.



# Executando o projeto no emulador

2

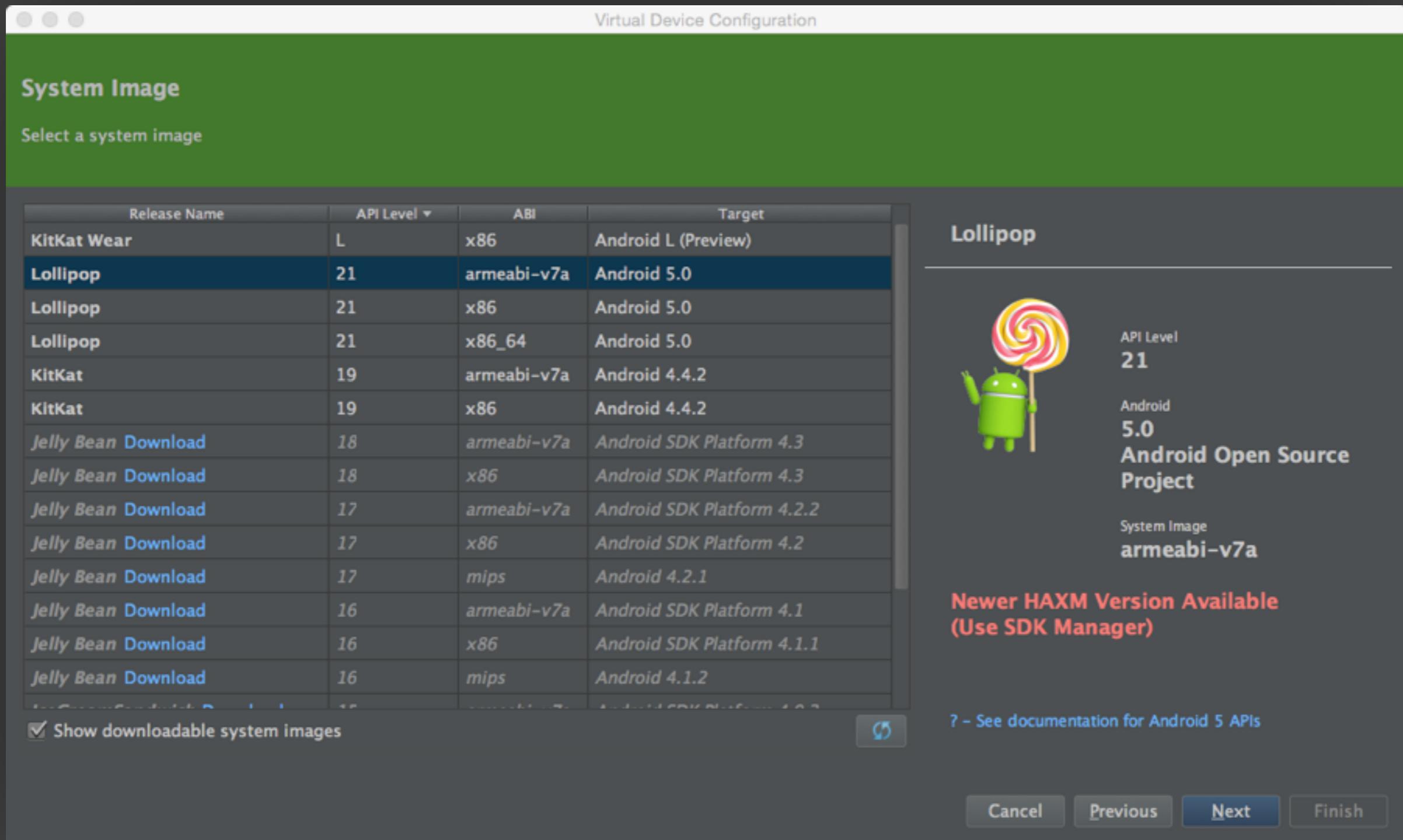
Selecione a categoria do dispositivo que deseja emular (Phone, Tablet, Wear, TV), o modelo (ex: Nexus 6) e depois clique em *Next*



# Executando o projeto no emulador

3

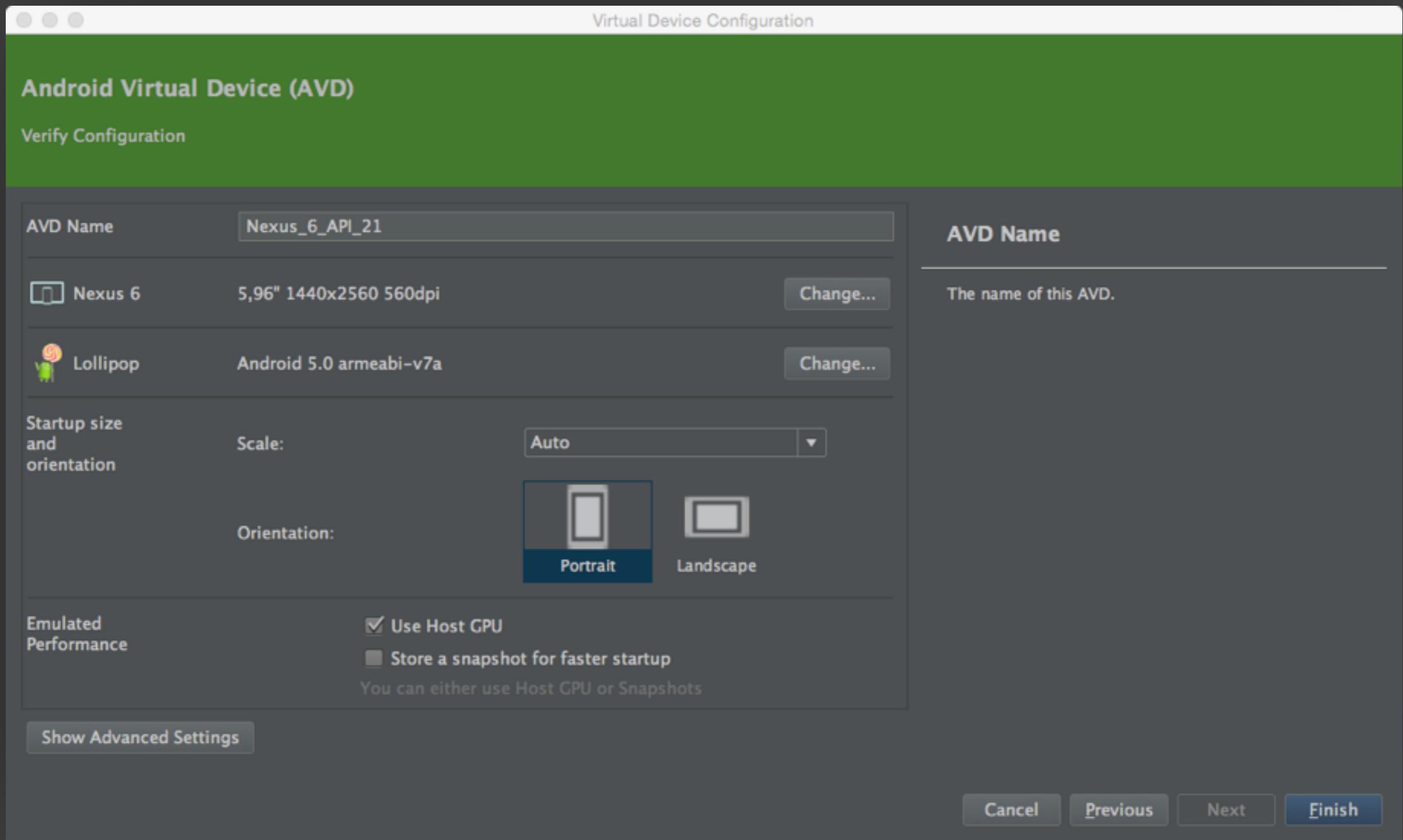
Selecione a imagem do sistema que irá emular (arm-v7a, mips, x86, x86\_64) e depois clique em *Finish*



# Executando o projeto no emulador

4

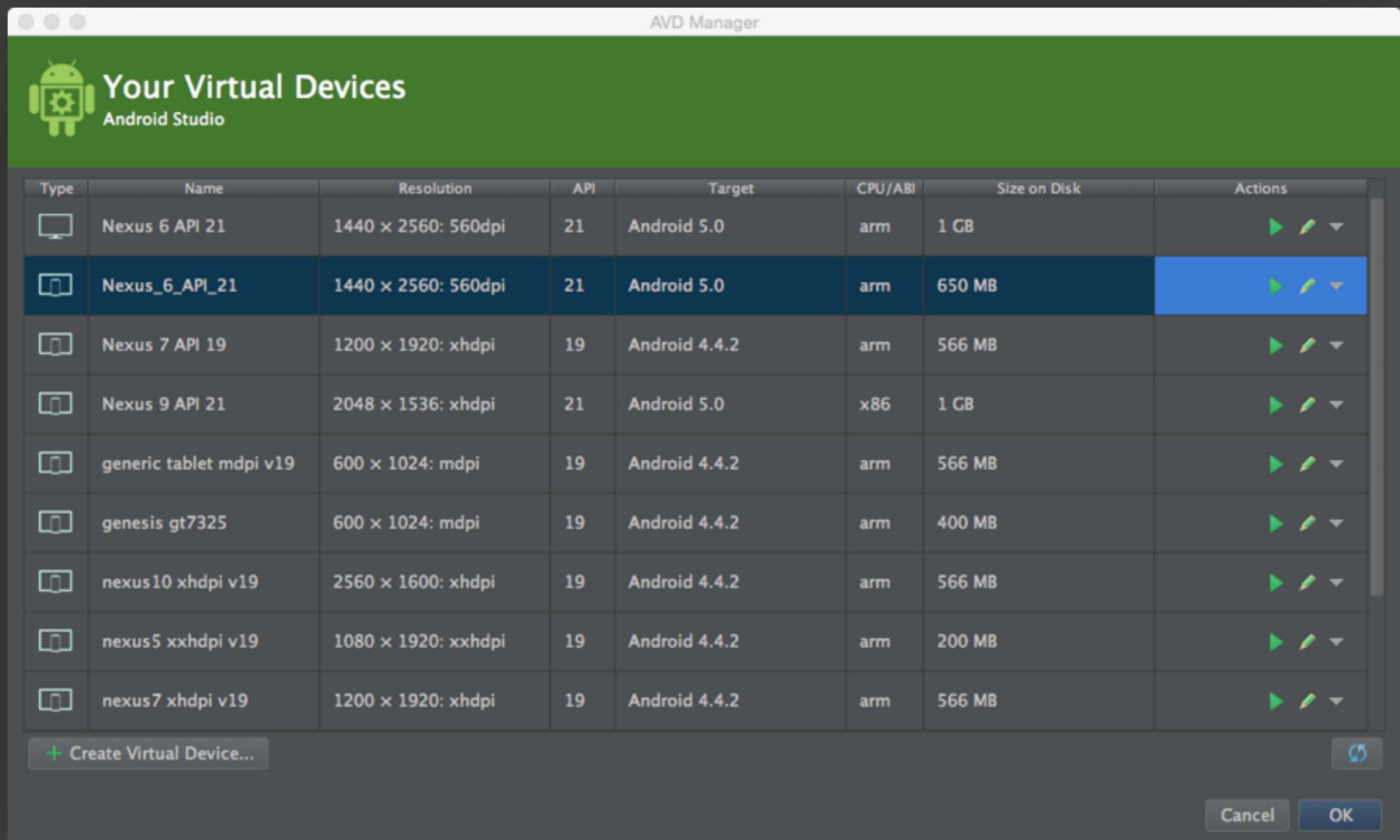
Para finalizar, configure aspectos do seu AVD (nome, orientação, escala, etc) e clique em *Finish*



# Executando o projeto no emulador

5

Selecione o AVD que você acabou de criar e clique no ícone para iniciar o emulador. Aguarde o início do emulador.



# Executando o projeto no device

1

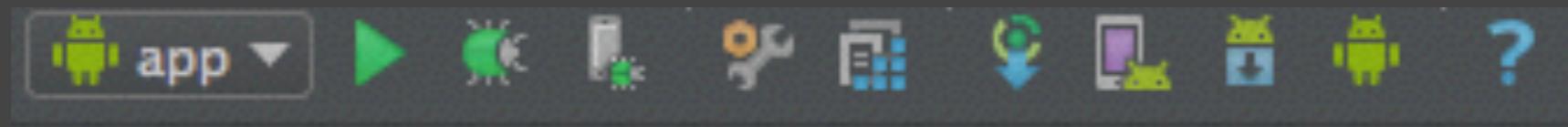
Para habilitar o seu dispositivo para execução de apps, vá em *Configurações > Opções de Desenvolvedor*. Marque “Depuração USB”

2

Depois, vá em Configurações > Segurança. Marque a opção “Fontes desconhecidas” para permitir instalação de apps que não sejam da Google Play Store

3

Depois, no Android Studio, selecione o módulo para executar e clique no botão do play ao lado. Se tudo ocorrer perfeitamente, o aplicativo irá executar no seu dispositivo.



Caso você esteja rodando Windows, será necessário instalar os drivers USB do fabricante do seu device.

#partiuCodeLab #AndroidStudio

---



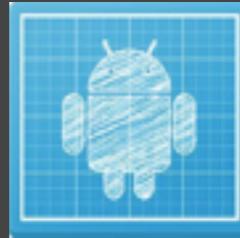
# Quer saber mais sobre Android?

---

## Siga os guidelines



[d.android.com/develop/](https://d.android.com/develop/)



[d.android.com/design/](https://d.android.com/design/)



[d.android.com/distribute/](https://d.android.com/distribute/)

# Quer saber mais sobre Android?

---

## Participe das comunidades



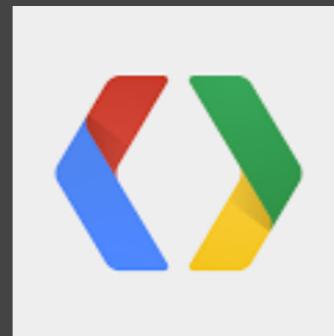
tasafo.org



+GDGBelém



+AndroidDevelopers



+DesenvolvedoresGoogle  
+GoogleDevelopers

# Obrigado

---



@ramonrabello  
[facebook.com/ramonrabello](https://facebook.com/ramonrabello)  
+RamonRabello



@androidnarede  
[facebook.com/androidnarede](https://facebook.com/androidnarede)  
[plus.androidnarede.com.br](https://plus.androidnarede.com.br)