

Aula 3: Hello World

```
object to mirror
mirror_mod.mirror_object
operation = "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
operation = "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
operation = "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True
```

```
selection at the end -add
mirror_mod.select= 1
mirror_mod.select=1
bpy.context.scene.objects.active = mirror_mod
("Selected" + str(modifier))
mirror_mod.select = 0
bpy.context.selected_objects.append(data.objects[one.name])
data.objects[one.name].select = 1
```

```
print("please select exactly one object")
```

```
-- OPERATOR CLASSES --
```

```
types.Operator):
    "x mirror to the selected object.mirror_mirror_x"
    "mirror X"
```

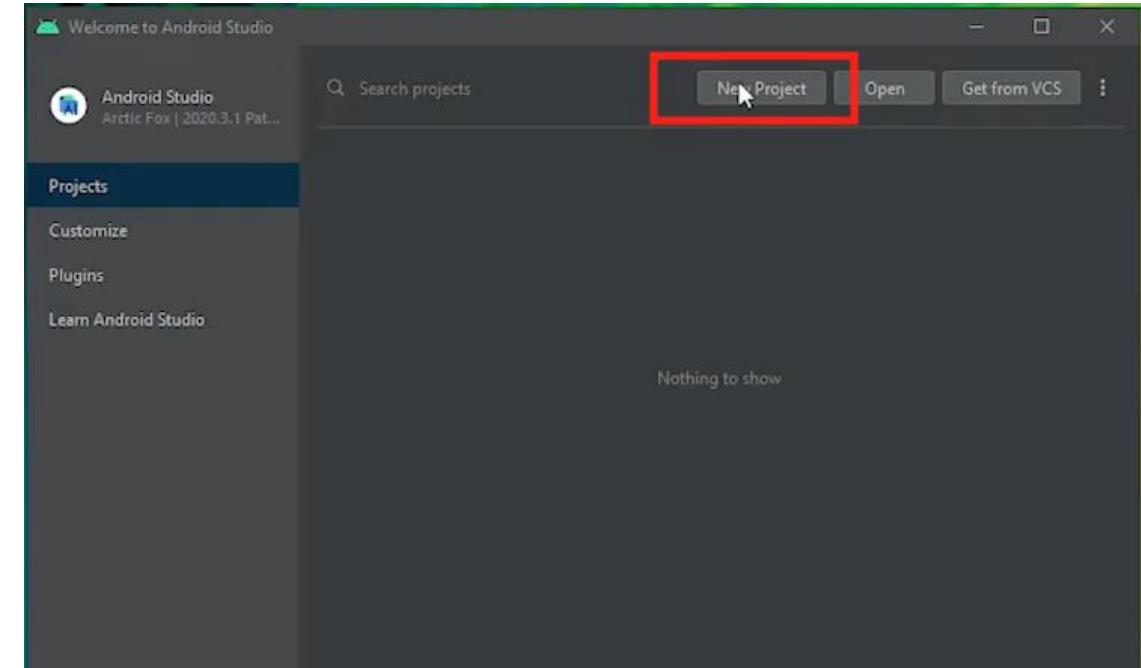
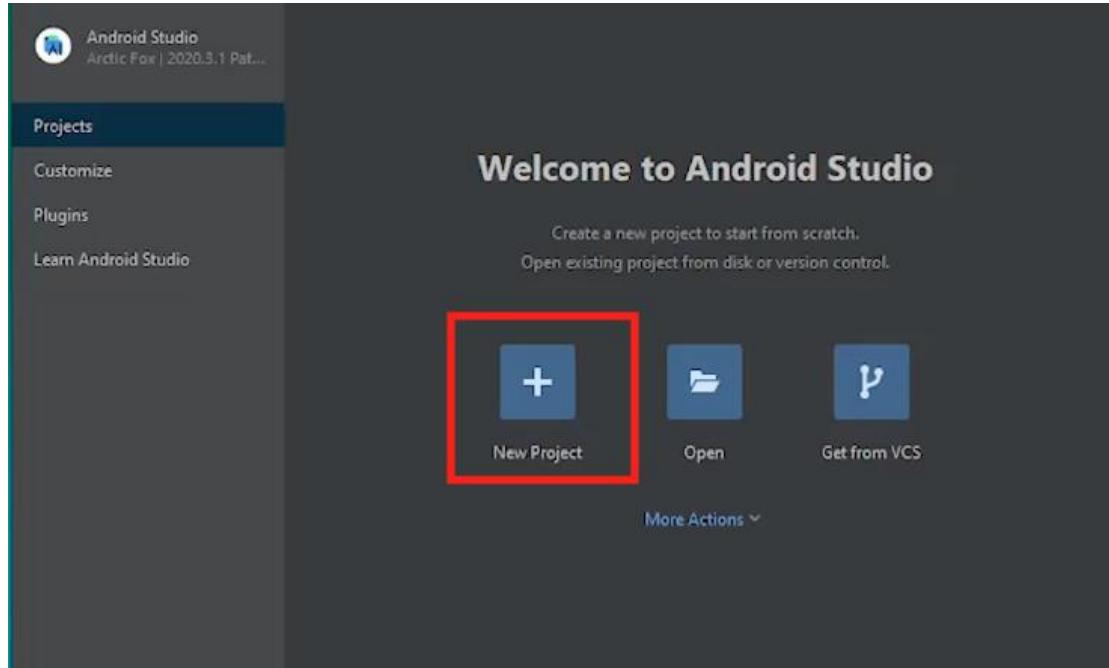


Introdução

- Nesta aula vamos construir o nosso primeiro projeto Android
- O objetivo é familiarizar vocês com o Android Studio e as principais opções na criação de um projeto novo

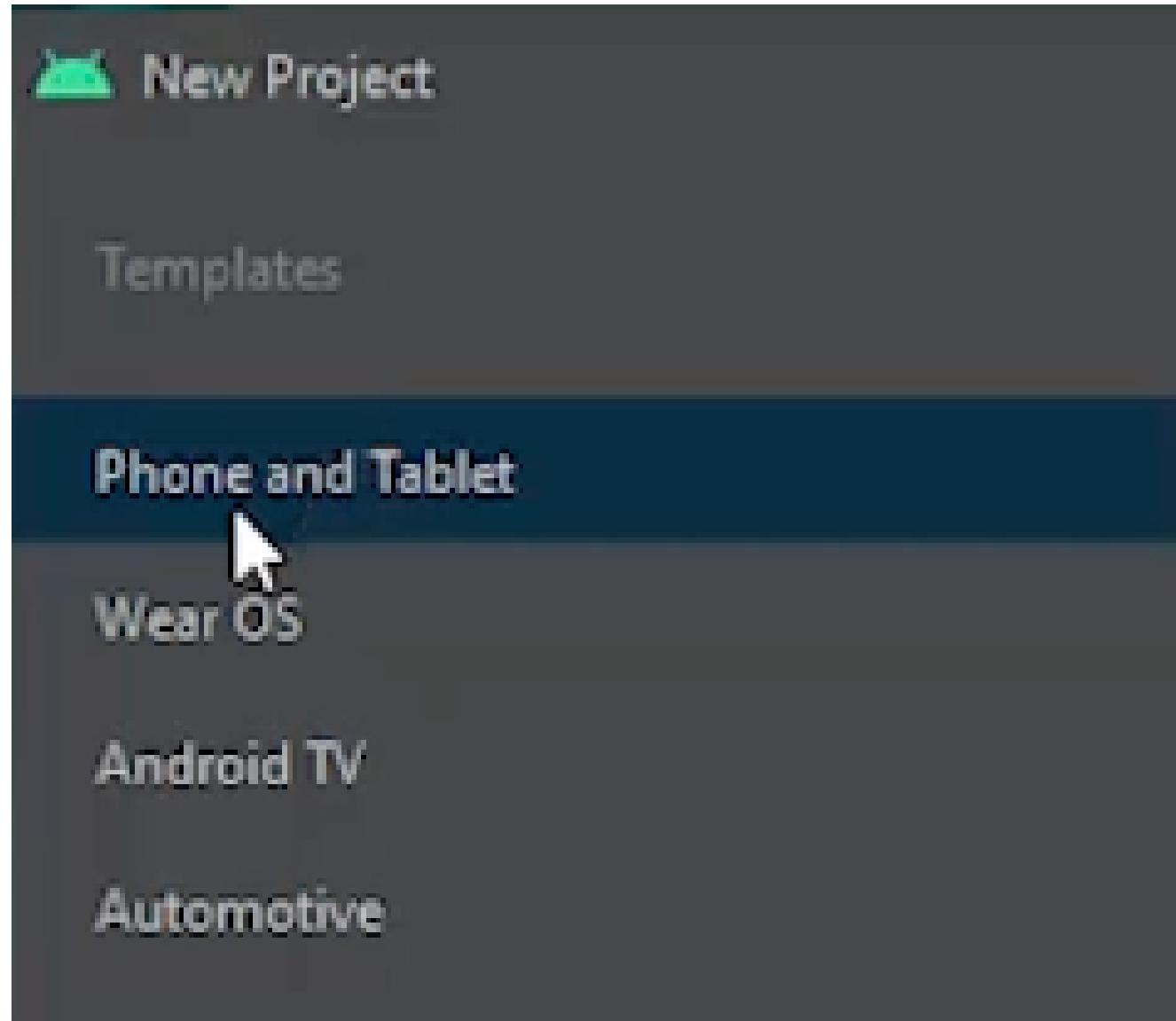
Criando um projeto novo

- A opção de criar um novo projeto pode aparecer nas formas abaixo



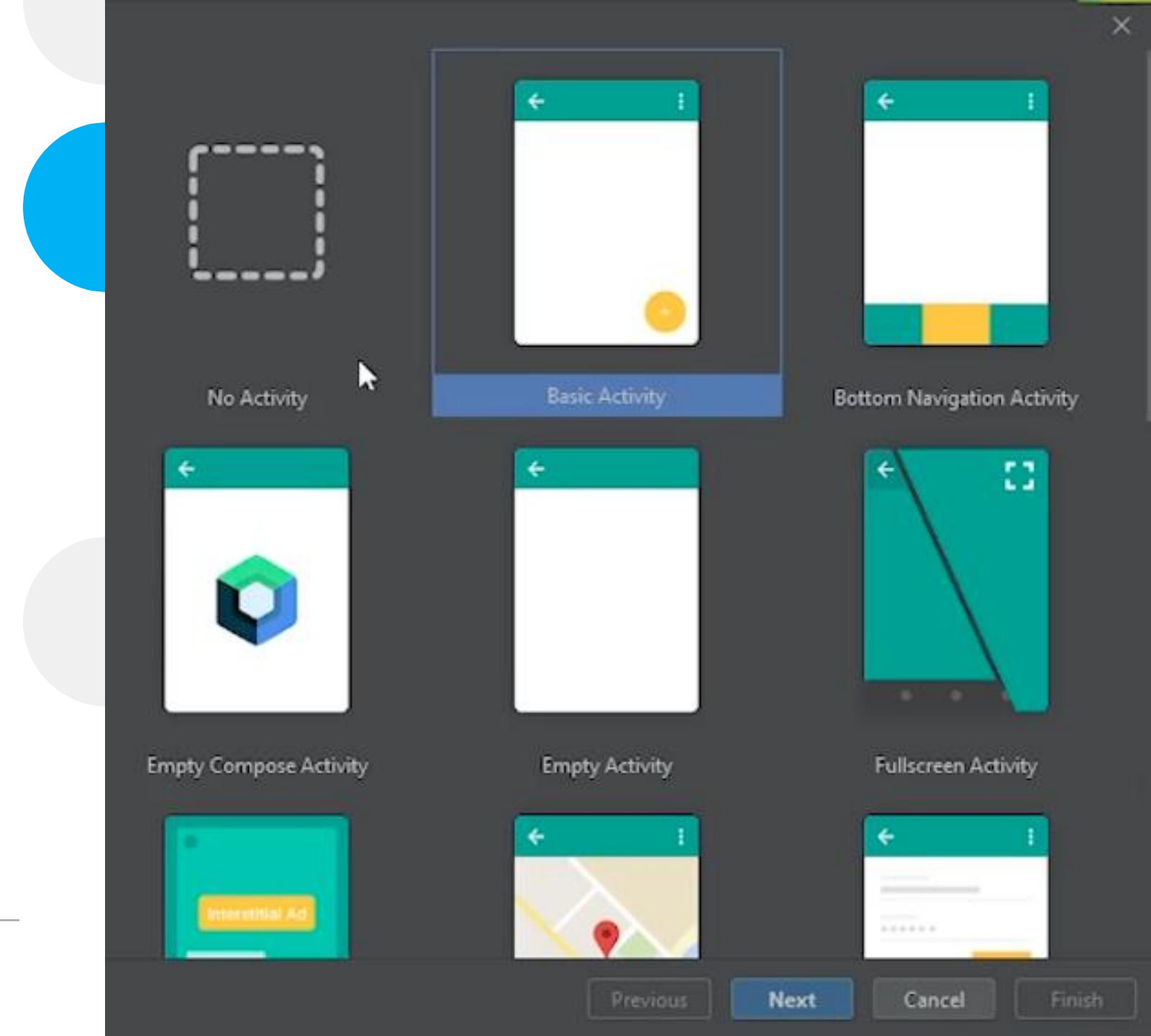
Seleção da versão Android

- O android Studio oferece templates para as versões do Android disponível
- Para este curso vamos utilizar somente os templates de telefone e tablet, mas se tiverem um tempo adicional aproveitem para explorar as outras opções



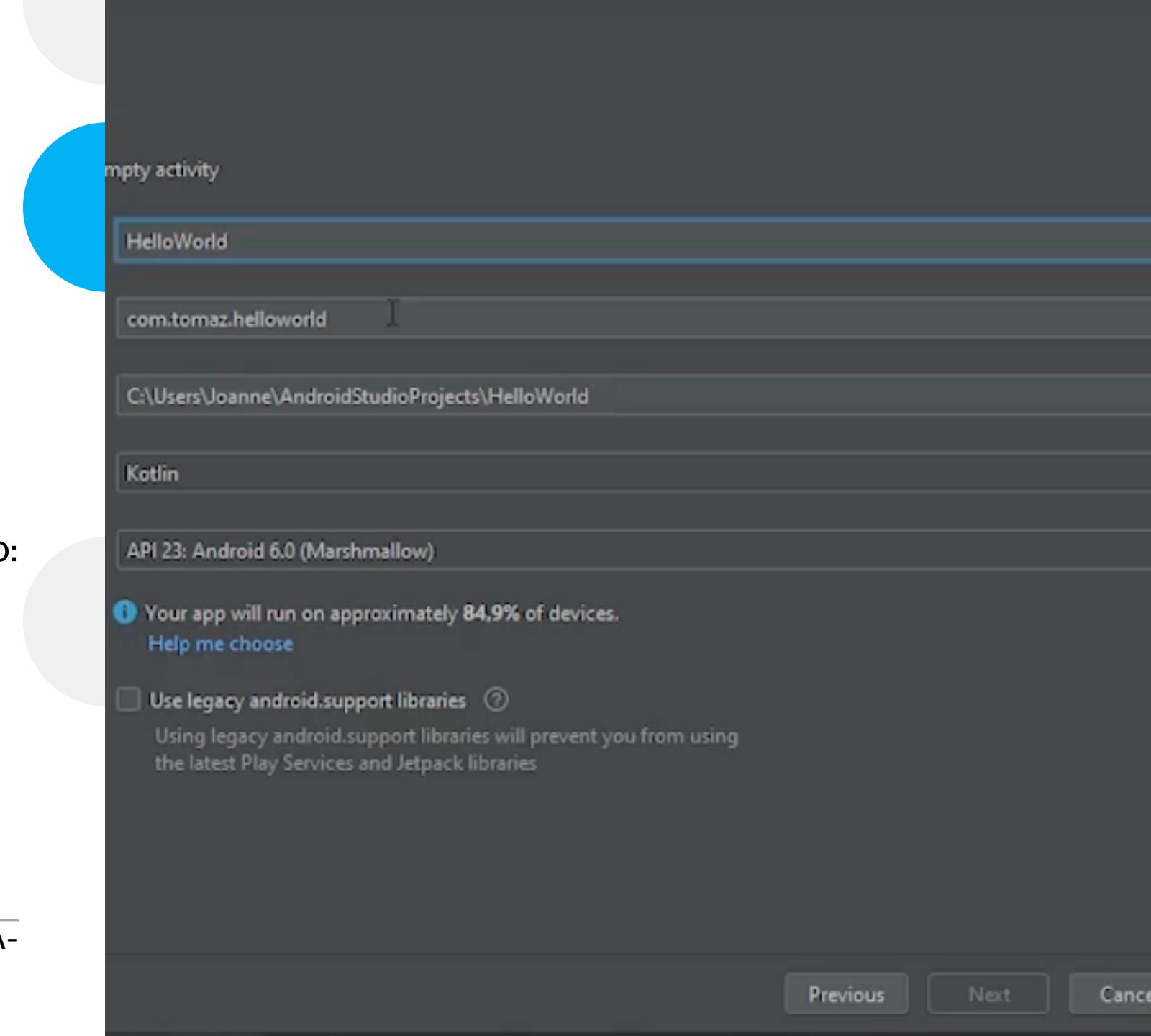
Seleção do template

- Existem vários templates diferentes disponíveis, cada um deles ajuda a começar um projeto novo com alguma característica específica
- Nesta aula vamos utilizar o Basic Activity, mas como atividade opcional; fica a sugestão de utilizar outros templates e ver o resultado



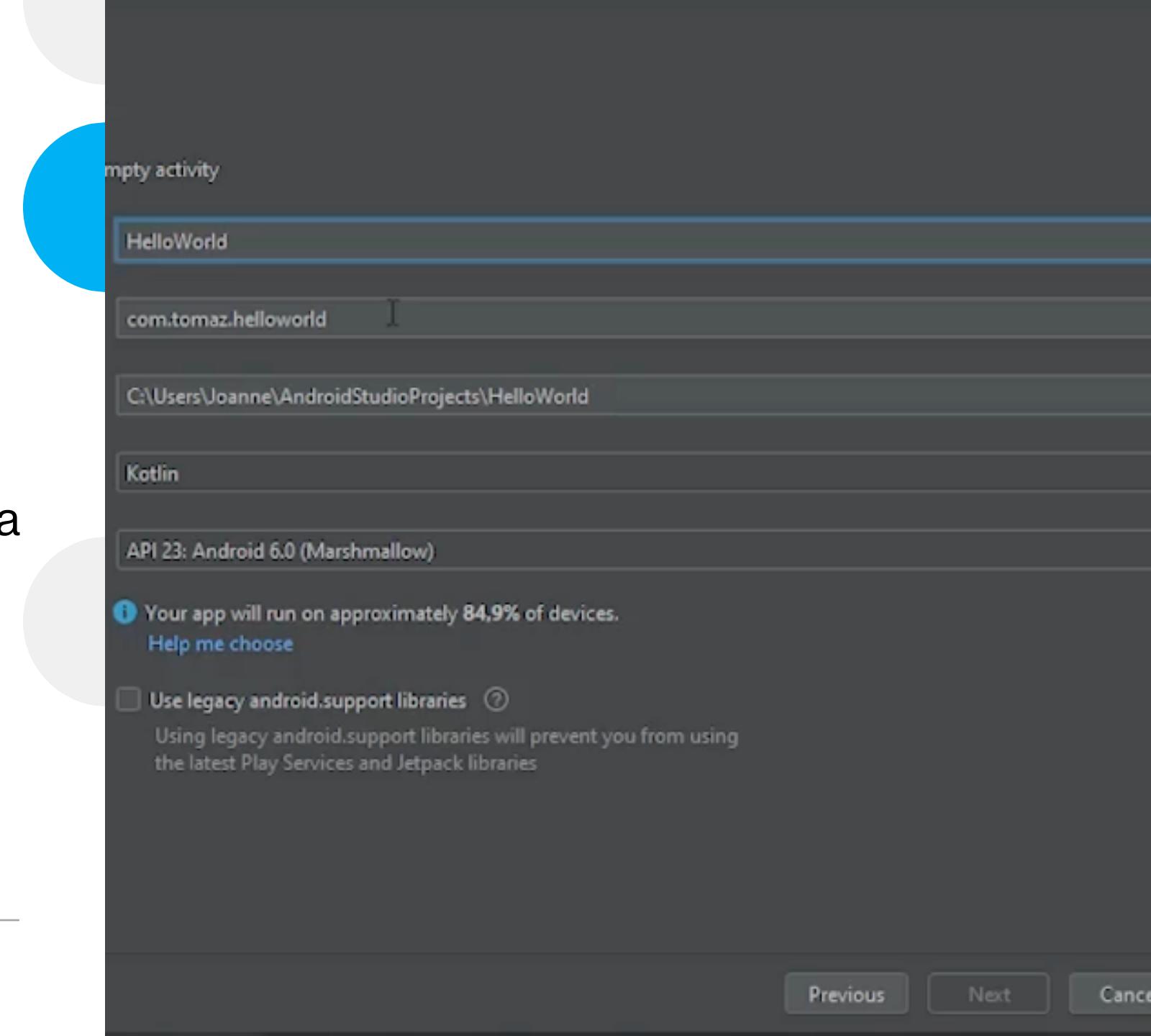
Criando o primeiro projeto – Nome e packageName

- Nome: é o nome do seu app
- PackageName: é o identificador do seu aplicativo na loja do google. Se você for submter seu ap para o google este nome precisa ser único
- Geralmente segue o seguinte padrão: *com. + sua empresa ou seu nome + nome do app*
- *Alem disso precisa seguir as seguintes regras:*
 - Deve ter pelo menos dois segmentos (um ou mais pontos).
 - Cada segmento deve começar com uma letra.
 - Todos os caracteres devem ser alfanuméricos ou um underscore [a-zA-Z0-9_].



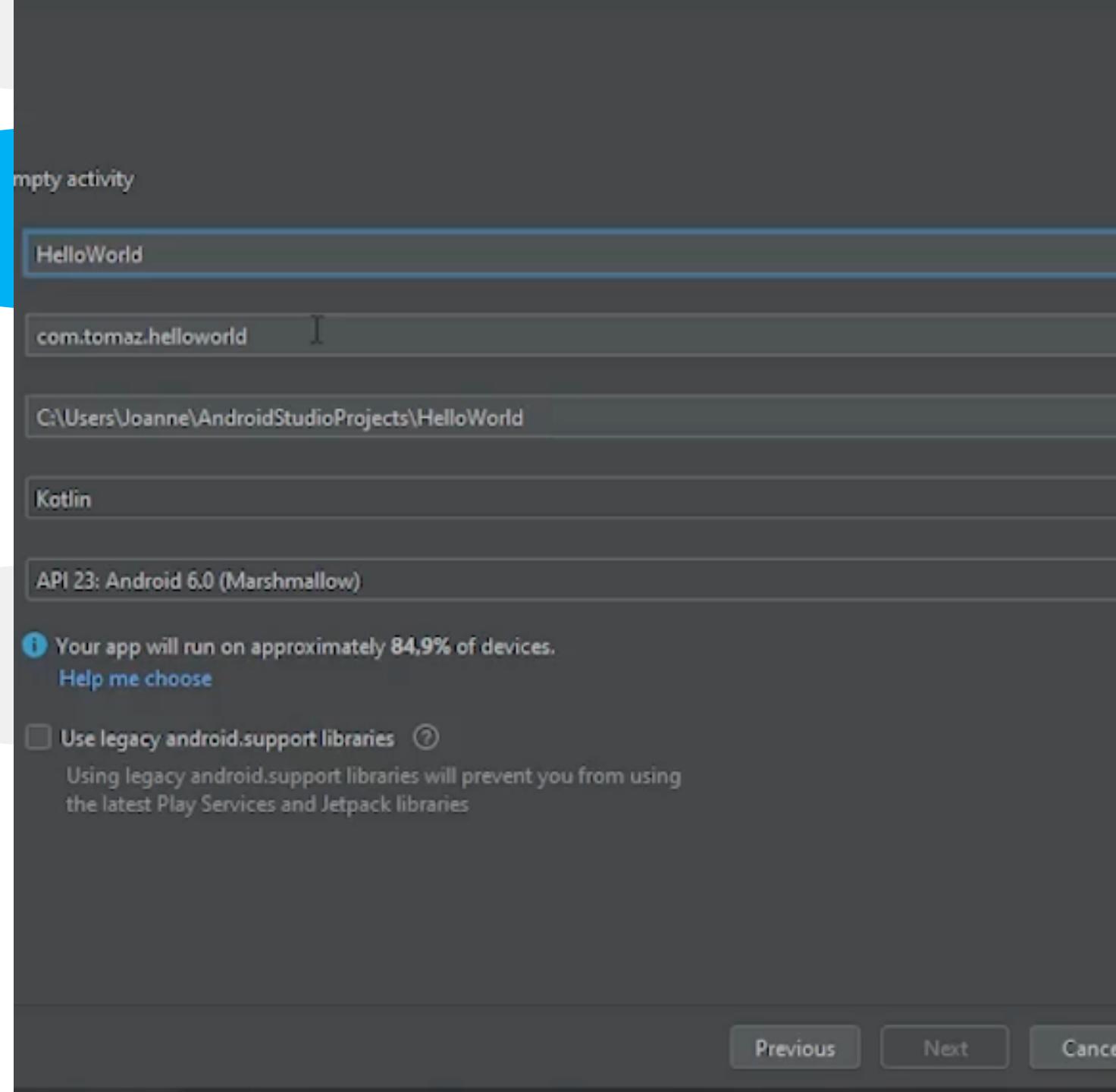
Criando o primeiro projeto – Outras opções

- Location: Pasta aonde será salvo seu app. Por padrão o Android Studio cria uma pasta com o nome do app na pasta AndroidStudioProject
- Linguagem: é possível criar app em Android utilizando Java ou Kotlin, neste curso usaremos somente kotlin



Criando o primeiro projeto – Mínimo SDK suportado

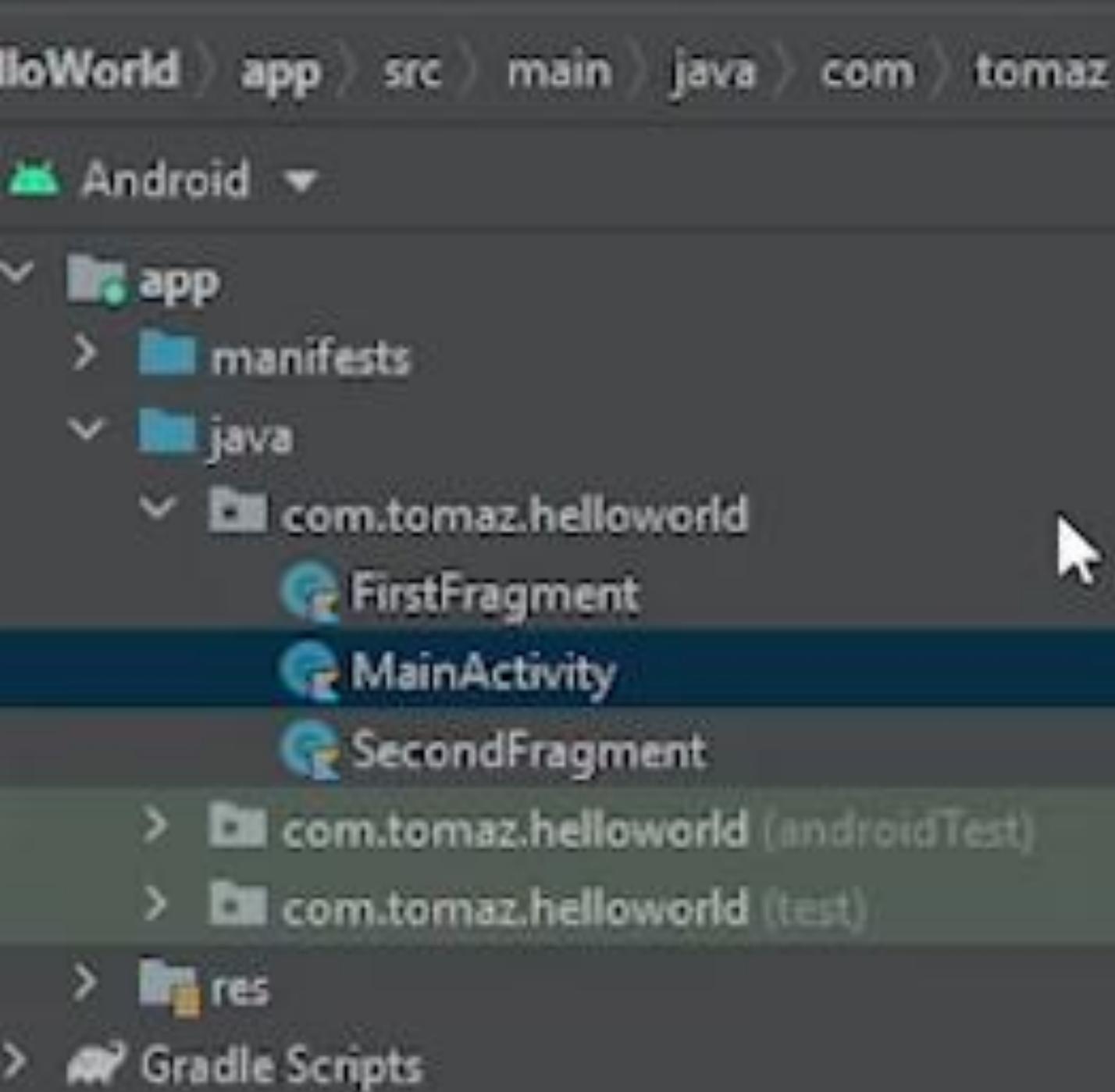
- Esta é uma das opções mais importantes, ele indica qual a menor versão do Android necessária para rodar sua aplicação
- Geralmente qualquer valor que garanta uma base de pelo 70% dos usuários é uma boa escolha
- Eventualmente será necessário escolher uma versão mais alta do android devido a uma funcionalidade disponível somente nesta versão



ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION	Marshmallow
4.0 Ice Cream Sandwich	15		Security
4.1 Jelly Bean	16	99,8%	Fingerprint Authentication Confirm Credential
4.2 Jelly Bean	17	99,2%	System
4.3 Jelly Bean	18	98,4%	App Linking Adoptable Storage
4.4 KitKat	19	98,1%	Multimedia
5.0 Lollipop	21	94,1%	4K Display Mode Support for MIDI
5.1 Lollipop	22	92,3%	Create digital audio playback objects APIs to associate audio devices
6.0 Marshmallow	23	84,9%	List of all audio devices Updated video processing
7.0 Nougat	24	73,7%	Flashlight API Reprocessing Camera
7.1 Nougat	25	66,2%	Updated ImageWriter API Image Reader class
8.0 Oreo	26	60,8%	User Input
8.1 Oreo	27	53,5%	Voice Interactions Assist API Bluetooth Stylus Support
9.0 Pie	28	39,5%	
10. Android 10	29	8,2%	

Mínimo SDK suportado – me ajude a escolher

- Quando selecionamos a opção me ajude a escolher o Android studio lista cada versão do android com a porcentagem da base que possui aquela versão além das principais inovações introduzidas naquela versão



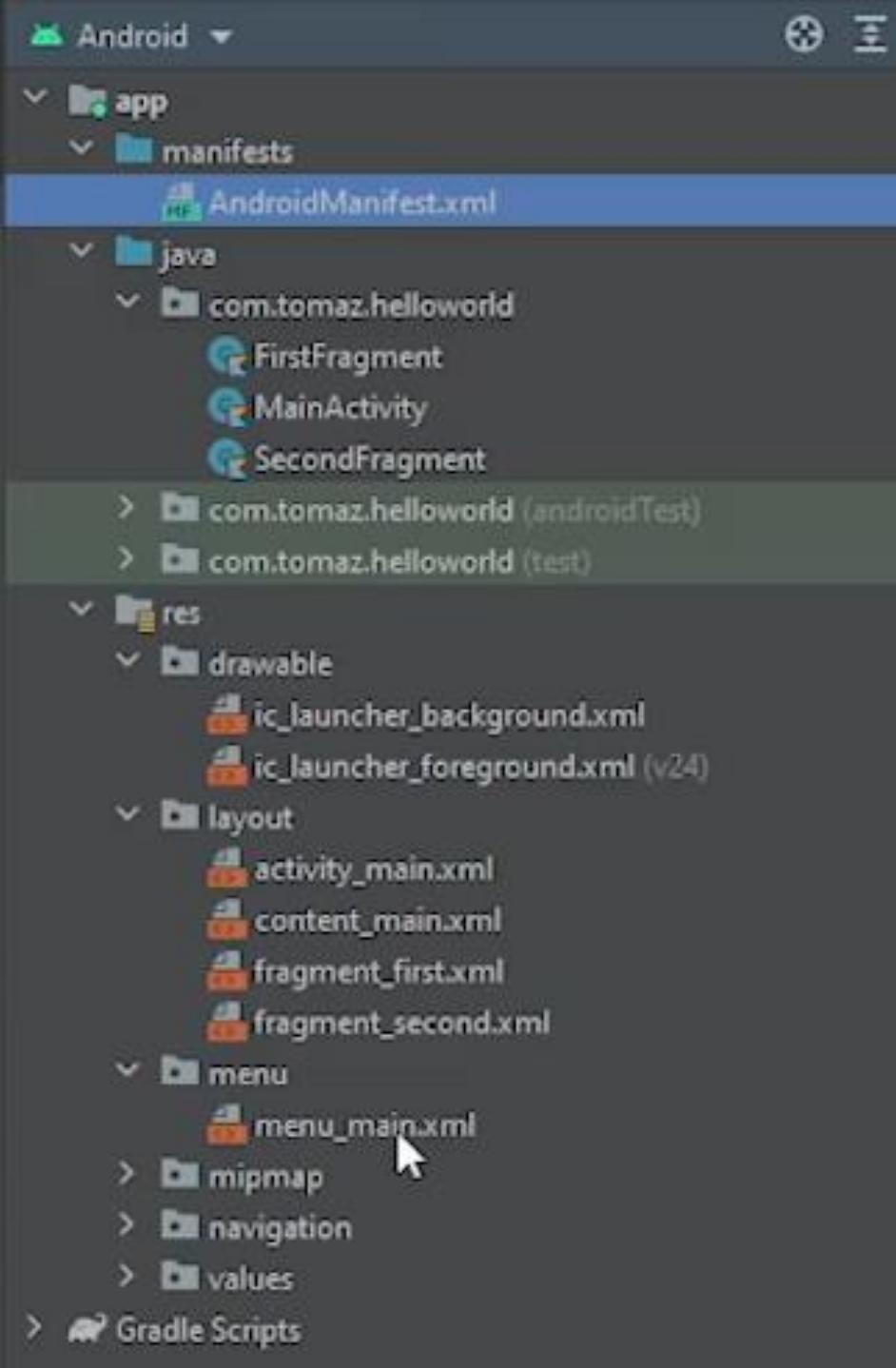
Estrutura do app

- O código fonte do nosso app ficará sempre na pasta com o nosso packageName, dentro da pasta **Java**
- Por razões históricas sempre aparecerá a pasta Java, mesmo no nosso caso aonde utilizamos kotlin.
- Todo app terá três pastas com o packageName, um com código fonte, outra para testes funcionais e a última para testes unitários

Arquivo manifest

- Todo aplicativo precisa ter um arquivo manifest, nele estão contidas informações essenciais sobre o aplicativo como:
- O nome do pacote do aplicativo
- Permissões que o app utiliza, como por exemplo localização, uso de microfone, etc
- Os principais componentes do app como:
 - Activities, serviços, broadcast receivers

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/re
3      package="com.tomaz.helloworld">
4
5      <application
6          android:allowBackup="true"
7          android:icon="@mipmap/ic_launcher"
8          android:label="HelloWorld"
9          android:roundIcon="@mipmap/ic_launcher_round"
10         android:supportsRtl="true"
11         android:theme="@style/Theme.HelloWorld">
12             <activity
13                 android:name=".MainActivity"
14                 android:exported="true"
15                 android:label="HelloWorld"
16                 android:theme="@style/Theme.HelloWorld.NoAction
17                     <intent-filter>
18                         <action android:name="android.intent.action
19
20                         <category android:name="android.intent.cat
21                     </intent-filter>
22             </activity>
23         </application>
24
25     </manifest>
```

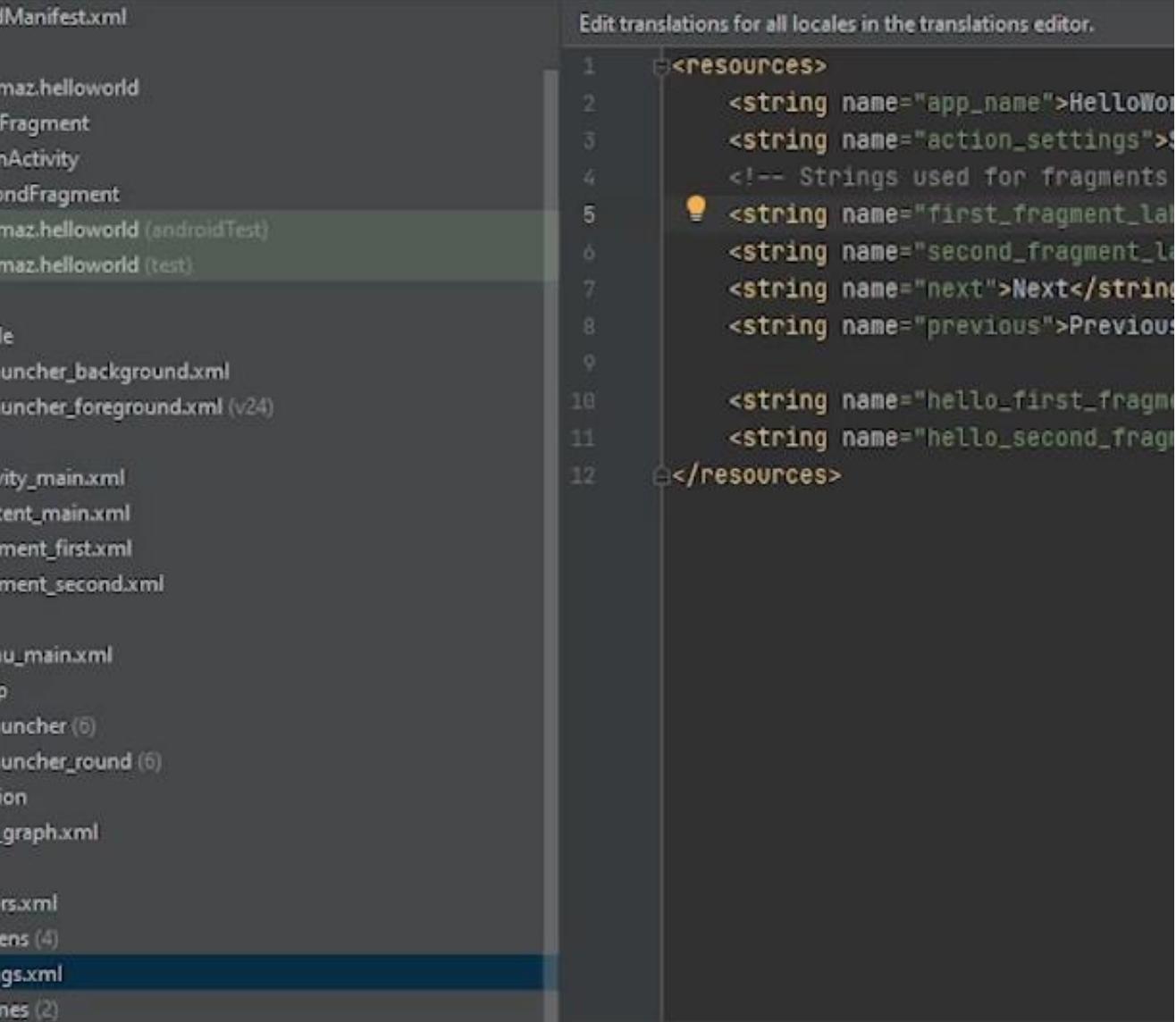


Pasta Res - resources

- Recursos são os arquivos adicionais e o conteúdo estático usado pelo seu código, como bitmaps, definições de layout, strings da interface do usuário, instruções de animação, entre outras coisas.
- Estes recursos ficam sempre na pasta **RES**
- **Drawable:** Contem todas as imagens usadas no app
- **Layout:** Contém todas as telas do app
- **Mipmap:** Armazena os ícones da aplicação
- **Values:** São arquivos XML que contêm valores simples, como strings, números inteiros e cores.

Arquivo strings.xml

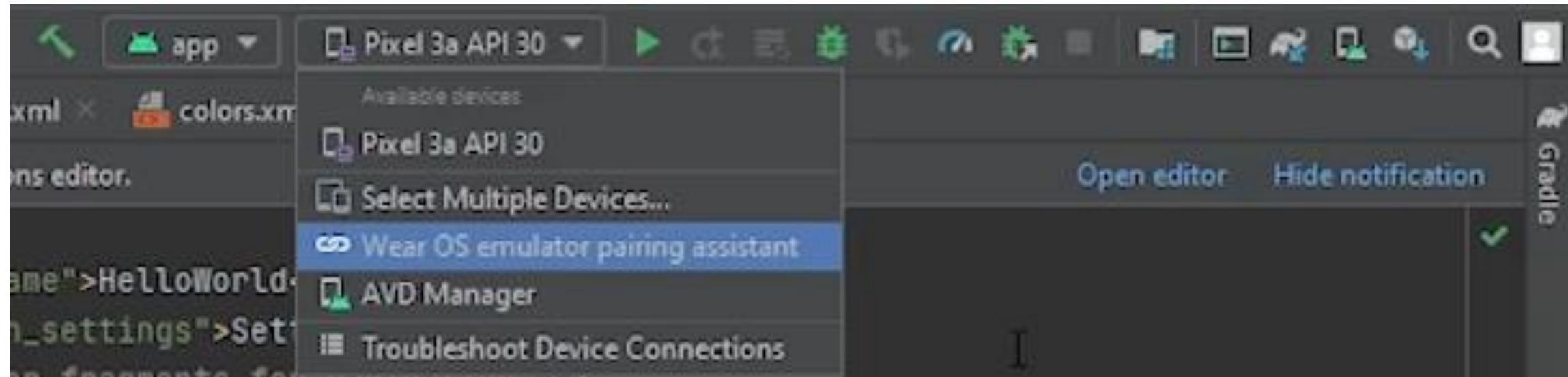
- Uma boa prática no desenvolvimento Android é nunca colocar os textos que seu aplicativo vai usar diretamente no código fonte ou no layout da tela
- O ideal é sempre utilizar o arquivo strings.xml
- Isso é super importante quando você estiver trabalhando em um aplicativo que vai ser traduzido para diversas línguas.



The screenshot shows the Android Studio code editor with the strings.xml file open. The file contains XML code for resource strings, including translations for various fragments and navigation actions. The code is as follows:

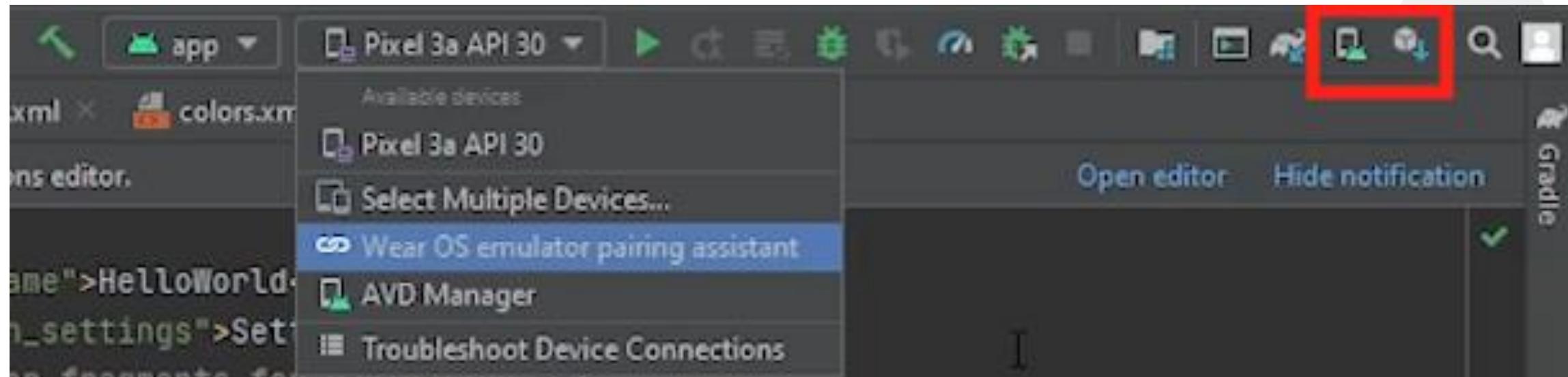
```
<resources>
    <string name="app_name">HelloWorld</string>
    <string name="action_settings">Settings</string>
    <!-- Strings used for fragments -->
    <string name="first_fragment_label">First Fragment</string>
    <string name="second_fragment_label">Second Fragment</string>
    <string name="next">Next</string>
    <string name="previous">Previous</string>
    <string name="hello_first_fragment">Hello First Fragment</string>
    <string name="hello_second_fragment">Hello Second Fragment</string>
</resources>
```

The code editor interface includes a sidebar with file navigation and a status bar at the bottom.



Menu Android Studio

- No menu aonde aparece "Pixel 3a", selecionamos em qual dispositivos queremos executar nossa aplicação
- Na "seta verde" é botão de rodar, ele vai gerar um novo build da nossa aplicação e se tudo estiver certo vai instalar no dispositivo selecionado
- Se o emulador selecionado não estiver executando, quando clicar no botão rodar não tem problema, ele automaticamente vai inicializar o emulador e depois vai instalar o app



Menu Android Studio – parte 2

Os dois últimos ícones são atalhos para o AVD manager e o SDK manager respectivamente

Hello world!

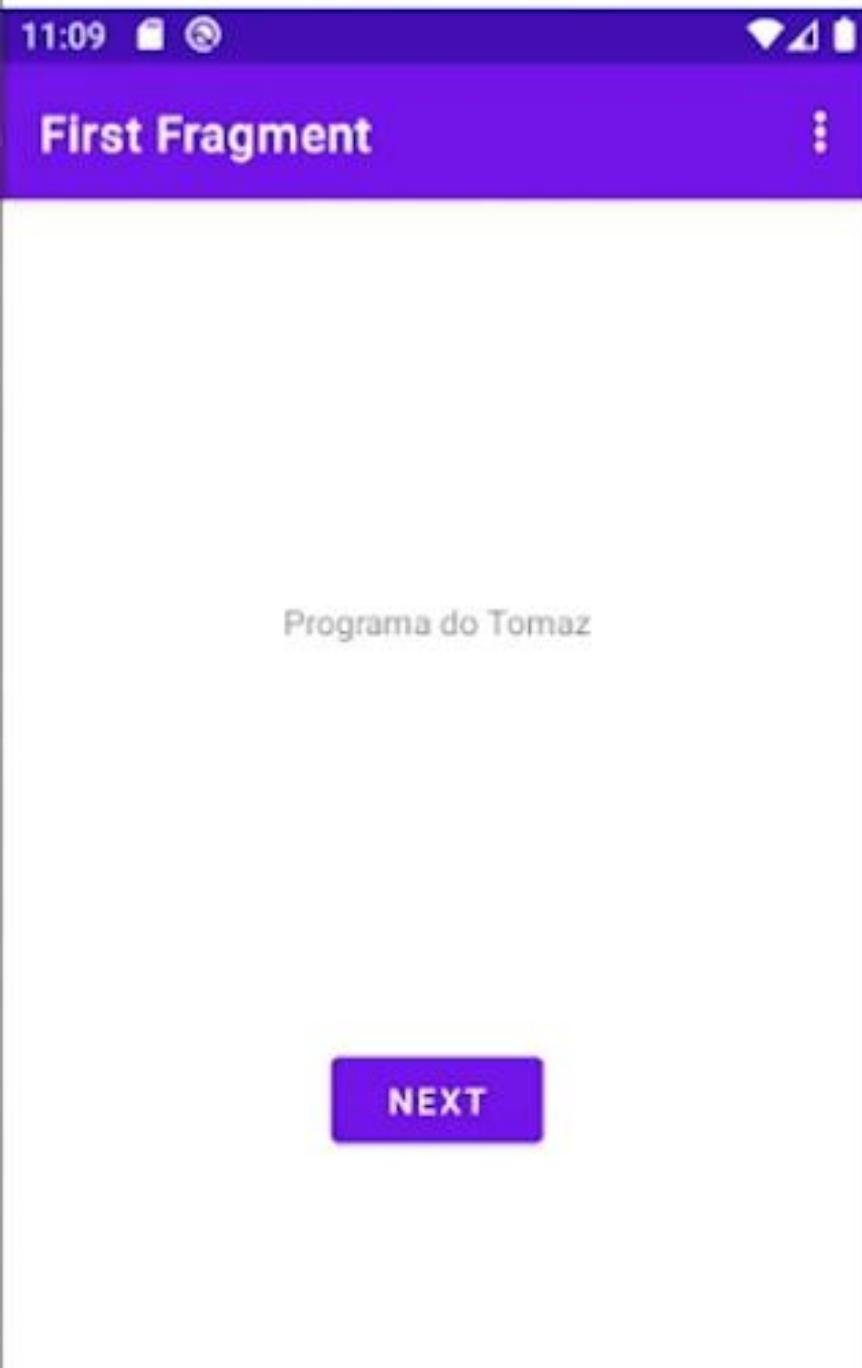
Hello first fragment

NEXT

Customizar a mensagem na tela do app

- Para alterarmos a mensagem exibida na tela precisamos alterar o conteúdo da propriedade "hello_first_fragment"
- No meu exemplo alterei para "Programa do Tomaz"
- Altere para incluir seu nome
- Altere as outras strings e veja o resultado no app

```
<resources>
    <string name="app_name">HelloWorld</string>
    <string name="action_settings">Settings</string>
    <!-- Strings used for fragments for navigation -->
    <string name="first_fragment_label">First Fragment</string>
    <string name="second_fragment_label">Second Fragment</string>
    <string name="next">Next</string>
    <string name="previous">Previous</string>
    <string name="hello_first_fragment">Hello first fragment</string>
```



Meu programa