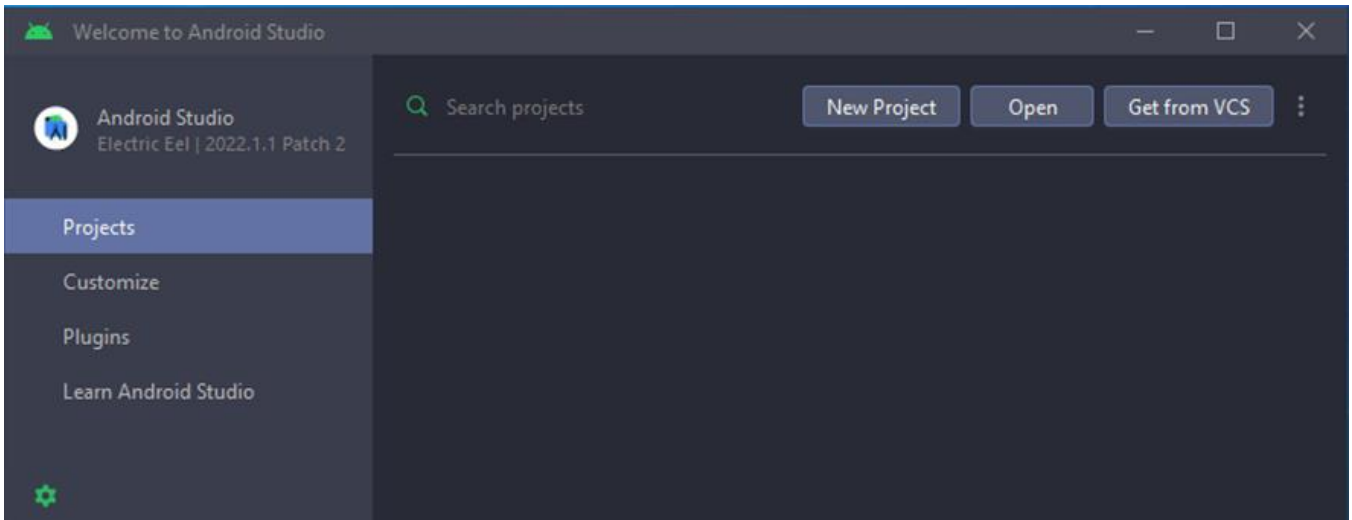


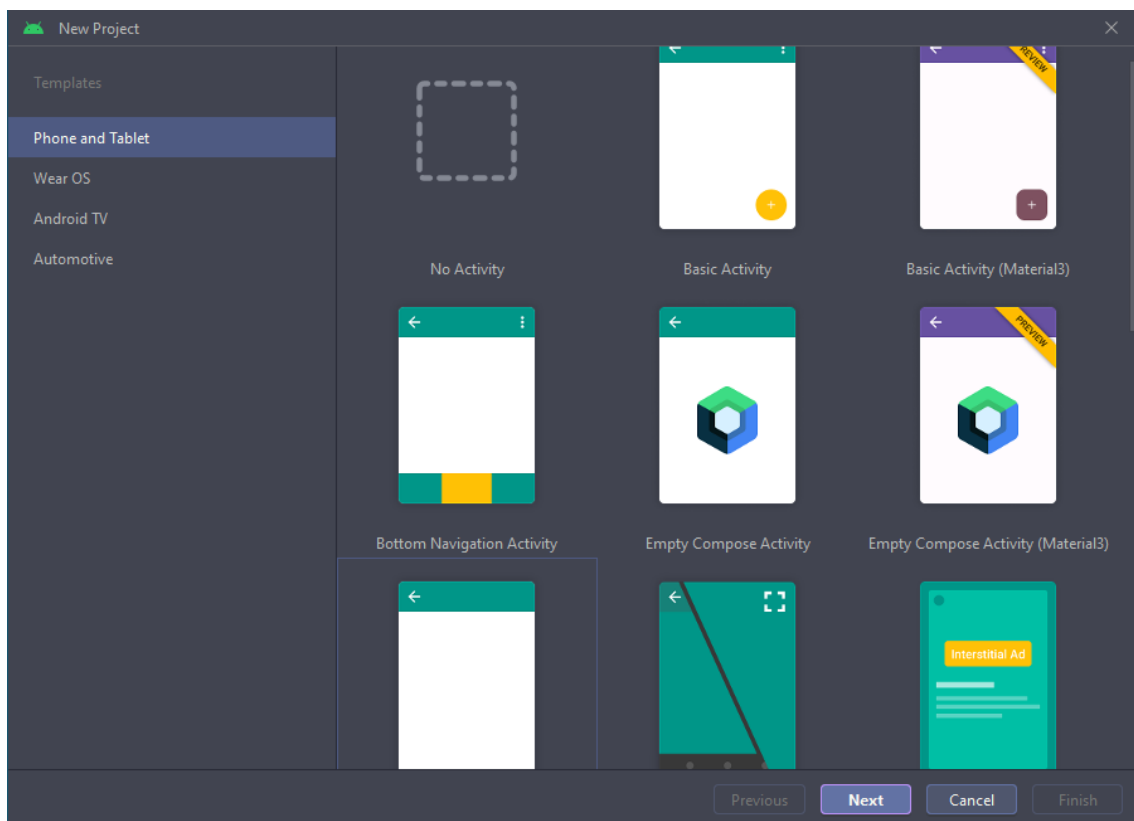
Programação Para Dispositivos Moveis

❖ *Vamos começar do 0 ...*

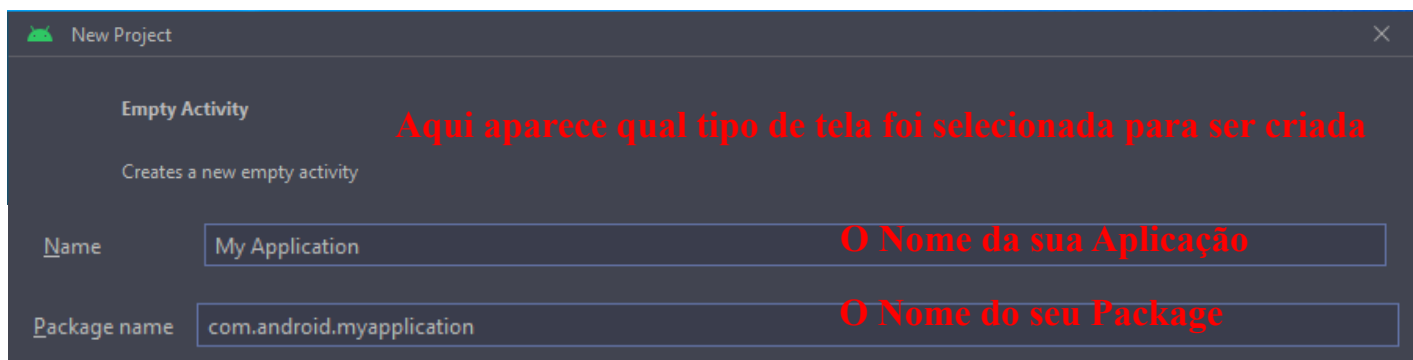
- Criando um novo Projeto



Selecionando um estilo de tela (Para o primeiro projeto recomendo **Basic Activity** ou **Empty Activity**)



- Agora nessa Tela vamos em Partes:



New Project

Empty Activity

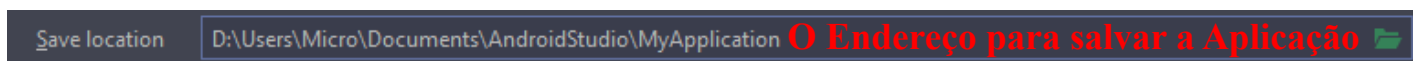
Creates a new empty activity

Name: My Application

Package name: com.android.myapplication

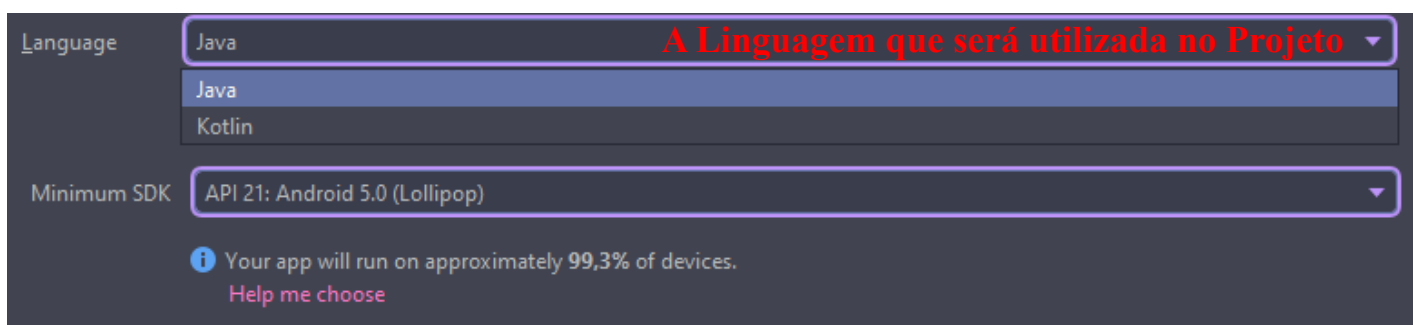
O **Package** serve para quem quer Publicar o seu aplicativo na **Play Store**, no caso se utiliza o seu domino (Site) de traz para frente.

Segue o Exemplo: **com.meusite.myapplication**



Save location: D:\Users\Micro\Documents\AndroidStudio\MyApplication

A pasta de destino não pode conter espaço “ ”.



Language: Java

Minimum SDK: API 21: Android 5.0 (Lollipop)

Your app will run on approximately 99,3% of devices. Help me choose

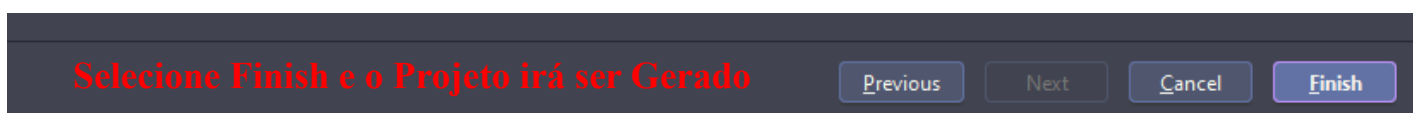
O **Minimum SDK** serve para escolher a versão Android base para a execução do projeto. Por padrão pegue a **5.0 (Lollipop)**.

A ferramenta consegue calcular quanto público consegue abranger com essa Versão.

- Mas qual é a melhor versão para se escolher?

A resposta é: Depende, se sua aplicação for mais simples pode escolher versões mais antigas do Android, já se for necessário alguma funcionalidade mais nova, requer um Android mais recente.

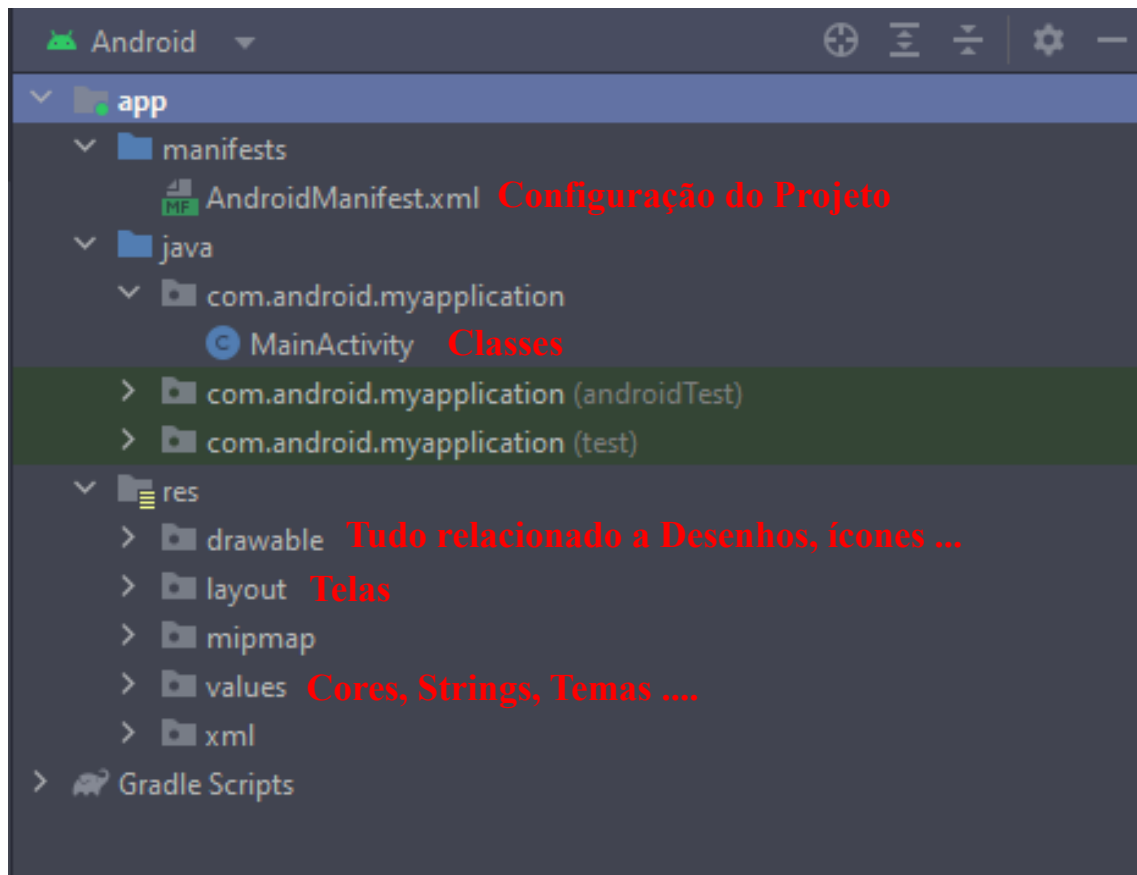
A proporção é o seguinte Quanto **Menor** a versão do Android, maior público alvo vai poder abranger, aplicações mais básicas. Agora Quanto **Maior** a versão do Android, menor público alvo vai poder abranger, mais aplicações e funções poderá ter para desenvolver o Projeto.



Selecione Finish e o Projeto irá ser Gerado

Previous Next Cancel Finish

Por ser a primeira vez é normal que o projeto demore para ser gerado, pois irá terminar de ser configurado, se necessário baixar algumas funções, tudo isso irá variar de acordo com o seu Hardware (Pc / Not) e Internet.



O Código por trás da Tela


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10    <TextView
11        android:layout_width="wrap_content"
12        android:layout_height="wrap_content"
13        android:text="Hello World!"
14        app:layout_constraintBottom_toBottomOf="parent"
15        app:layout_constraintEnd_toEndOf="parent"
16        app:layout_constraintStart_toStartOf="parent"
17        app:layout_constraintTop_toTopOf="parent" />
18
19 </androidx.constraintlayout.widget.ConstraintLayout>
```

A Tela

Hello World!

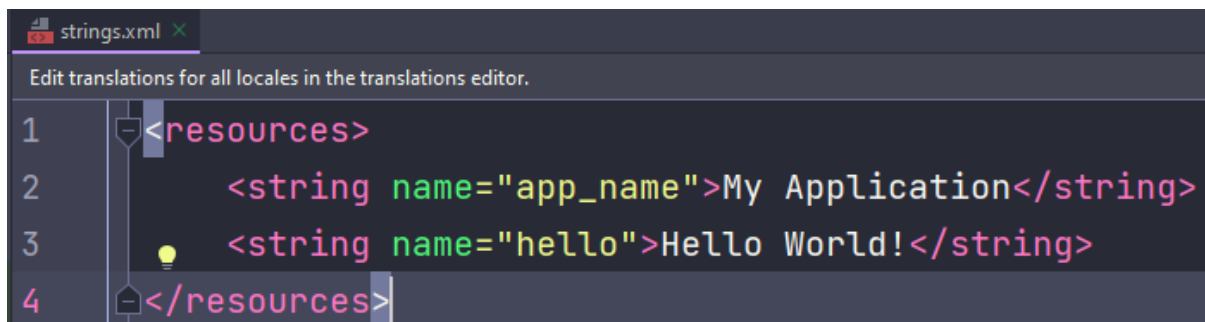
❖ *O que é um TextView?*

<TextView> Permite exibir texto em um aplicativo, pode ser usado para formatar o texto usando recursos de formatação de **String** do Android.



❖ *O que é uma String?*

Ela disponibiliza a Internacionalização do seu programa, de forma que ele pega esse texto e quando exibido para o usuário ele identifica a sua região (**Ex:** BR) e o traduz para essa língua. (serve para fazer a internacionalização, é tornar o aplicativo disponível em vários idiomas).



❖ *O que é um Button?*

<Button> Permite que os usuários interajam com o aplicativo, clicando nele para realizar uma ação específica. Ele é usado para criar botões clicáveis que os usuários podem pressionar para executar uma ação ou fazer algo acontecer no aplicativo Android.

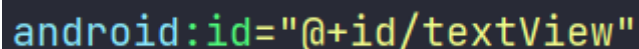


❖ *O que é PlainText?*

<EditText> é um elemento de interface do usuário no Android Studio que permite que o usuário insira e edite texto.



android:id: Ele é usado para identificar de maneira única e permitir que você se refira a ela no código do seu aplicativo.



android:layout_width/height: é usado para definir a **largura/altura** de uma **View** em um layout **XML** no Android Studio. Esse atributo define a largura da **View** dentro do layout e é uma das propriedades básicas de layout do Android.

atributos XML	
<code>android:layout_height</code>	Especifica a altura básica da exibição.
<code>android:layout_width</code>	Especifica a largura básica da exibição.

Pode ser do tipo **wrap_content** ou **match_parent**

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"
```

FirstFragment: é um exemplo de um **Fragment** no Android Studio, que é uma classe que representa um componente de interface do usuário reutilizável em um aplicativo Android.

onViewCreated:

Ex: `view.findViewById(R.id.lista)`: Recupera os componentes da interface)

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    nome = findViewById(R.id.nome);  
    valor = findViewById(R.id.valor);  
    spinContribuinte = findViewById(R.id.spinContribuinte);  
    btEnviar = findViewById(R.id.btEnviar);  
    textView = findViewById(R.id.textView);  
}
```

onClick: código a ser executado quando o elemento é clicado pelo usuário.

setOnClickListener: é um método utilizado para definir um ouvinte de clique em um elemento de interface do usuário no Android Studio, como um botão.

```
btEnviar.setOnClickListener(new View.OnClickListener()
```

Para mais conhecimento ascese: <https://developer.android.com/studio/intro?hl=pt-br>