

ANDROID

First App



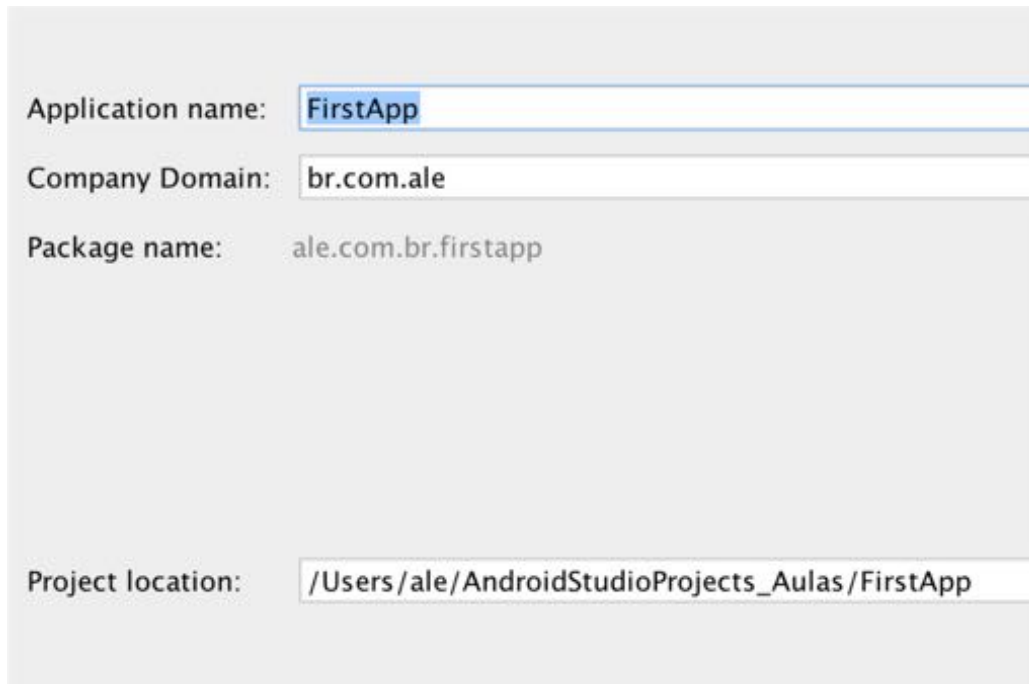
Prof. Alessandro Brawerman

Primeiro Projeto: FirstApp



Primeiro Projeto: FirstApp

- Crie um novo projeto, chame-o de FirstApp.
 - Verifique que nesta tela, temos o App name, Company Domain que forma o package name (id do App) e project location.



The screenshot shows the 'New Project' dialog in Android Studio. It contains the following fields and values:

Field	Value
Application name:	FirstApp
Company Domain:	br.com.ale
Package name:	ale.com.br.firstapp
Project location:	/Users/ale/AndroidStudioProjects_Aulas/FirstApp

Primeiro Projeto: FirstApp

- Selecione a plataforma que o app irá ser executado.
 - Phone and Tablet
 - Minimum SDK – API 19 – Android 4.4 90% dos dispositivos no mundo.

Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK

Lower API levels target more devices, but have fewer features available.
By targeting API 19 and later, your app will run on approximately 90.1% of the devices that are active on the Google Play Store.
[Help me choose](#)

☐ Wear

Minimum SDK

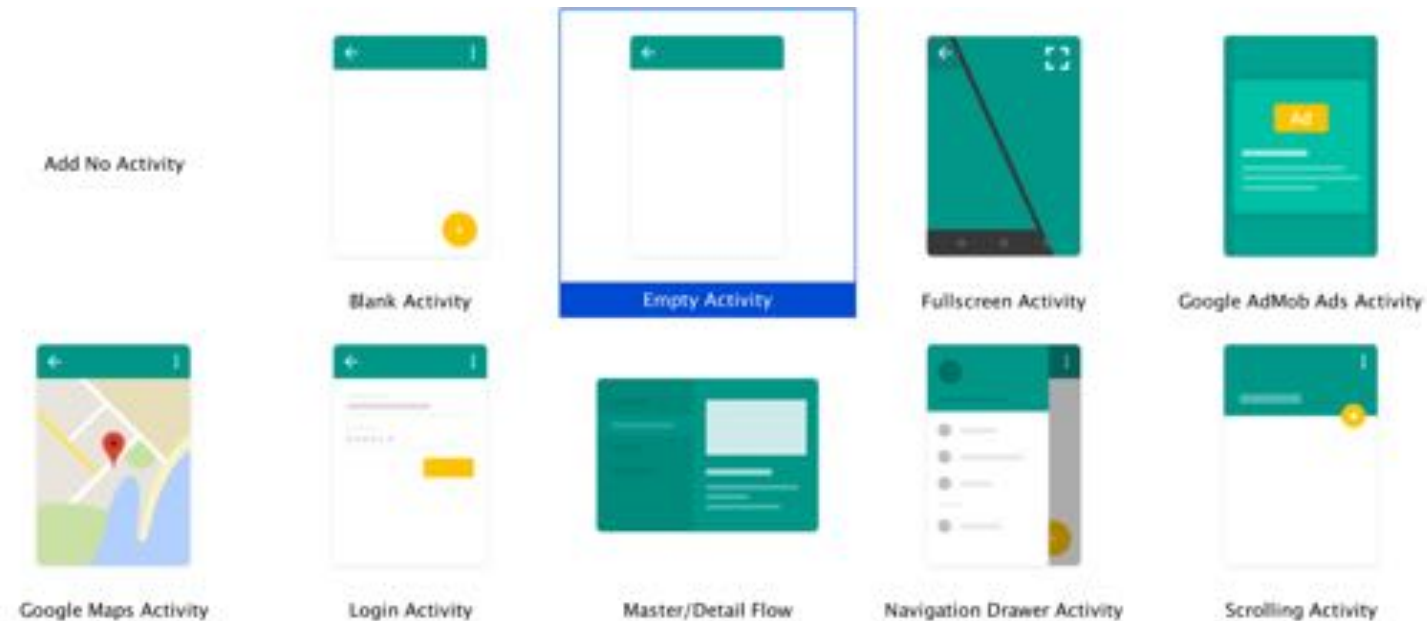
☐ TV

Minimum SDK

☐ Android Auto

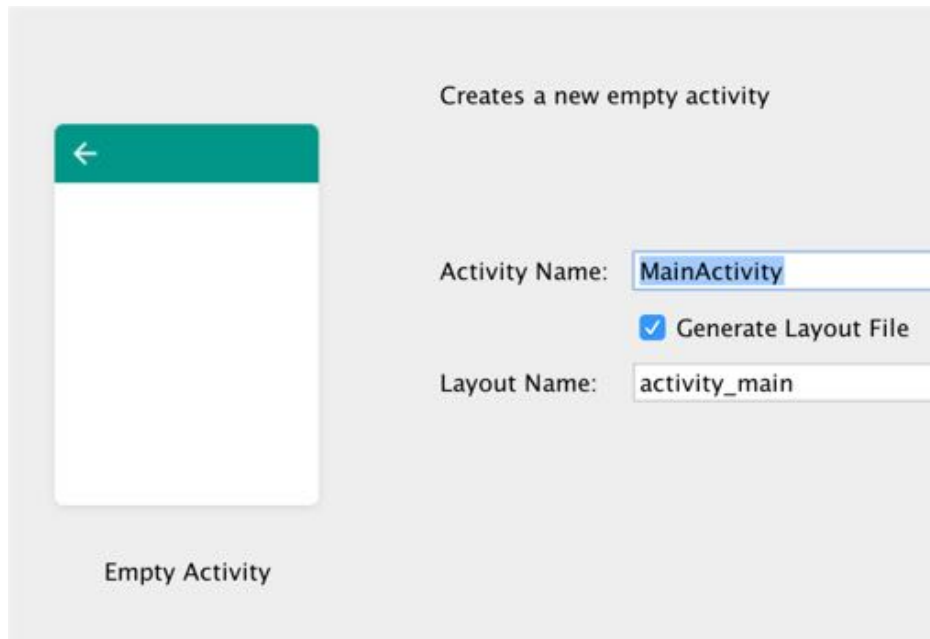
Primeiro Projeto: FirstApp

- Activity
 - Escolha o template Empty Activity.

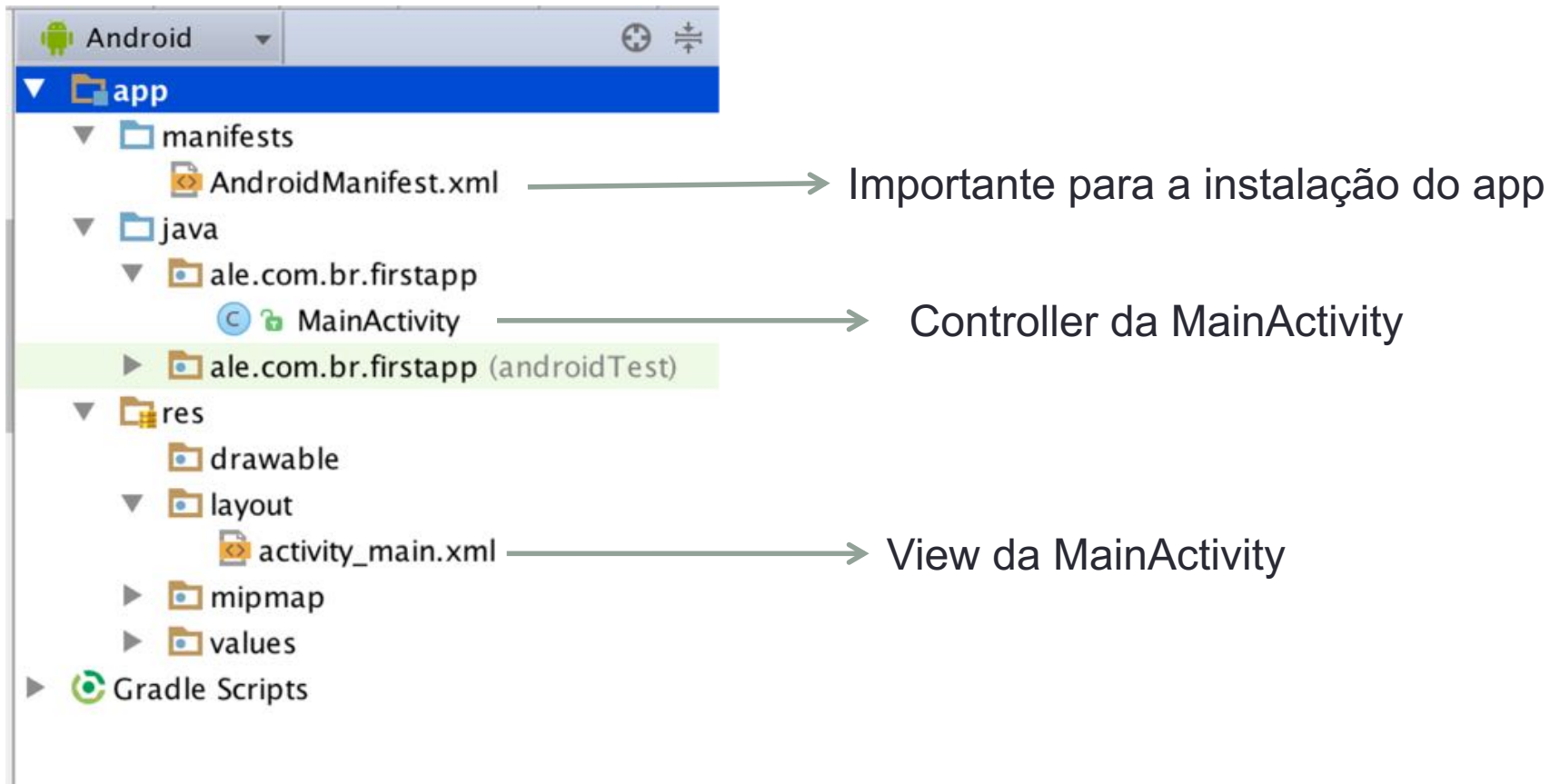


Primeiro Projeto: FirstApp

- Activity
 - Nomeie esta activity de MainActivity.
 - Ela será dividida em 2 itens importantes
 - O layout, ou seja, a View: activity_main (é um XML)
 - O código fonte, ou seja, o Controller: MainActivity.java

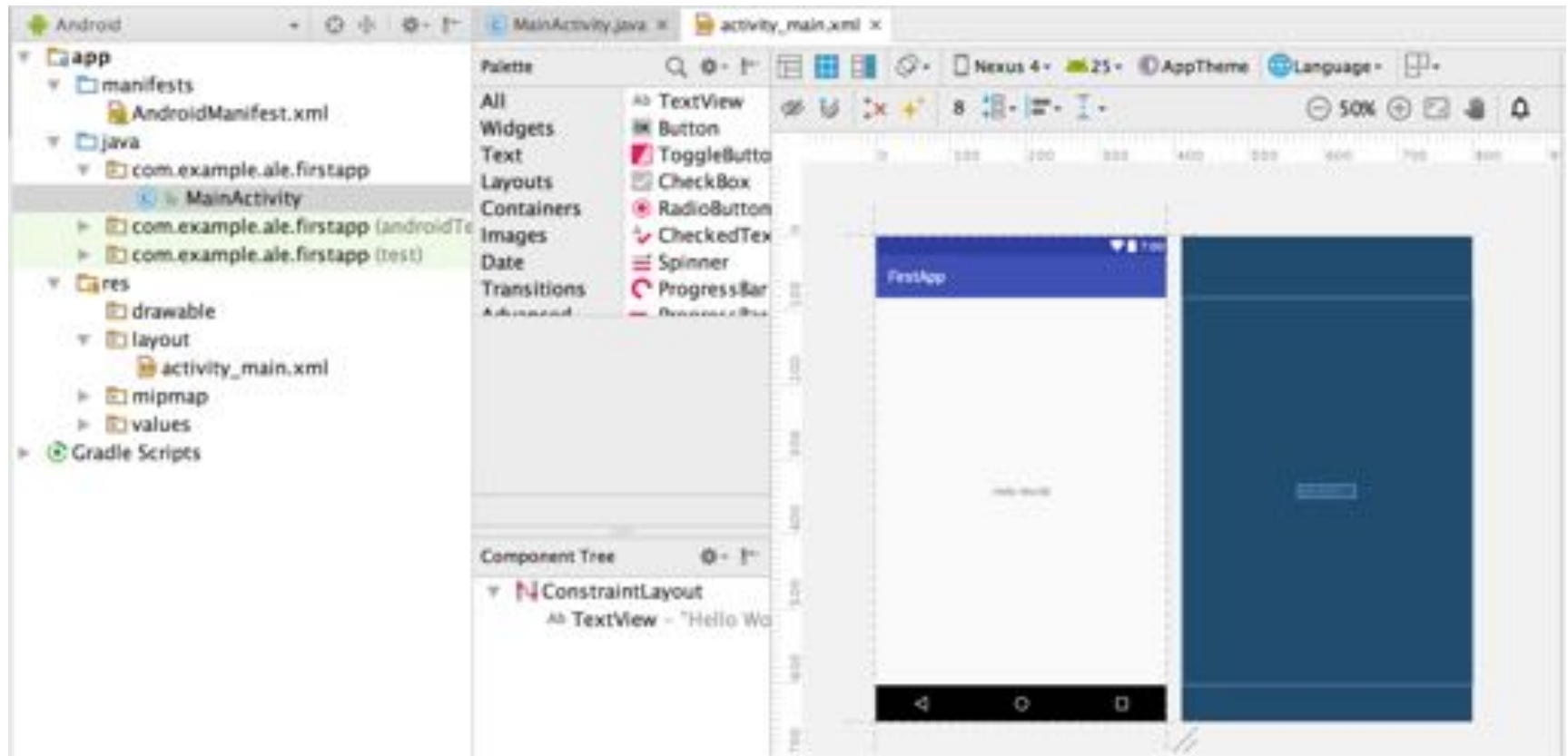


Primeiro Projeto: FirstApp



Primeiro Projeto: FirstApp

Controller View



Primeiro Projeto: FirstApp

MainActivity.java

```
package ale.com.br.firstapp;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Primeiro Projeto: FirstApp

Manifesto

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ale.com.br.firstapp" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Android SDK

- Por padrão o Android Studio não instala tudo o que está disponível para se trabalhar.
- Abra o SDK Manager no Android Studio.



Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: [Edit](#)

SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
<input type="checkbox"/>	Android 8.0 (O)	26	2	Not installed
<input type="checkbox"/>	Android 7.1.1 (Nougat)	25	3	Partially installed
<input type="checkbox"/>	Android 7.0 (Nougat)	24	2	Partially installed
<input type="checkbox"/>	Android 6.0 (Marshmallow)	23	3	Not installed
<input type="checkbox"/>	Android 5.1 (Lollipop)	22	2	Not installed
<input type="checkbox"/>	Android 5.0 (Lollipop)	21	2	Not installed
<input type="checkbox"/>	Android 4.4W (KitKat Wear)	20	2	Not installed
<input checked="" type="checkbox"/>	Android 4.4 (KitKat)	19	4	Installed
<input type="checkbox"/>	Android 4.3 (Jelly Bean)	18	3	Not installed
<input type="checkbox"/>	Android 4.2 (Jelly Bean)	17	3	Not installed
<input type="checkbox"/>	Android 4.1 (Jelly Bean)	16	5	Partially installed
<input type="checkbox"/>	Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
<input type="checkbox"/>	Android 4.0 (IceCreamSandwich)	14	4	Not installed
<input type="checkbox"/>	Android 3.2 (Honeycomb)	13	1	Not installed
<input type="checkbox"/>	Android 3.1 (Honeycomb)	12	3	Not installed
<input type="checkbox"/>	Android 3.0 (Honeycomb)	11	2	Not installed
<input type="checkbox"/>	Android 2.3.3 (Gingerbread)	10	2	Not installed
<input type="checkbox"/>	Android 2.3 (Gingerbread)	9	2	Not installed
<input type="checkbox"/>	Android 2.2 (Froyo)	8	2	Not installed


Android SDK

SDK Platforms **SDK Tools** SDK Update Sites

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

	Name	Version	Status
<input checked="" type="checkbox"/>	Android SDK Build-Tools		Installed
<input type="checkbox"/>	GPU Debugging tools		Not Installed
<input type="checkbox"/>	CMake		Not Installed
<input type="checkbox"/>	LLDB		Not Installed
<input type="checkbox"/>	Android Auto API Simulators	1	Not installed
<input type="checkbox"/>	Android Auto Desktop Head Unit emulator	1.1	Not installed
<input checked="" type="checkbox"/>	Android Emulator	26.1.2	Installed
<input checked="" type="checkbox"/>	Android SDK Platform-Tools	26.0.0	Installed
<input checked="" type="checkbox"/>	Android SDK Tools	26.0.2	Installed
<input checked="" type="checkbox"/>	Documentation for Android SDK	1	Installed
<input type="checkbox"/>	Google Play APK Expansion library	1	Not installed
<input type="checkbox"/>	Google Play Billing Library	5	Not installed
<input type="checkbox"/>	Google Play Licensing Library	1	Not installed
<input checked="" type="checkbox"/>	Google Play services	43	Installed
<input type="checkbox"/>	Google Web Driver	2	Not installed
<input type="checkbox"/>	Instant Apps Development SDK	1.0.0	Not installed
<input checked="" type="checkbox"/>	Intel x86 Emulator Accelerator (HAXM installer)	6.1.1	Installed
<input type="checkbox"/>	NDK	15.1.4119039	Not installed
<input checked="" type="checkbox"/>	Support Repository		
<input checked="" type="checkbox"/>	ConstraintLayout for Android		Installed

AVD – Android Virtual Device

- Para executar um app no emulador é necessário primeiramente criar um dispositivo compatível com a versão do Android instalada.
- Abra o AVD. 
- Crie um dispositivo virtual.
 - Escolha um dispositivo, observando o tamanho e resolução.
 - Por padrão da disciplina use o Nexus 4
 - Escolha então a API a ser instalada neste dispositivo. Deve ser uma que esteja instalada no Android Studio.
 - Por padrão da disciplina use a API 19 – Android 4.4
 - Finalize.

Android Virtual Device



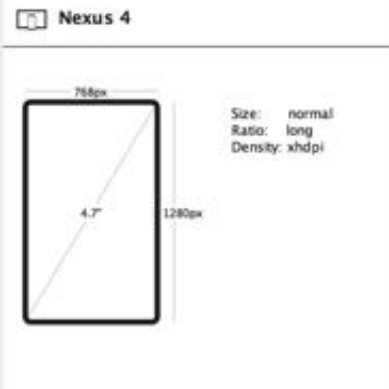
Virtual devices allow you to test your application without having to own the physical devices.

+ Create Virtual Device...

Android Virtual Device

Q-					
Category	Name	Play Store	Size	Resolution	Density
TV	Pixel		5.0"	1080x1...	xxhdpi
Wear	Nexus 5		4.0"	480x800	hdpi
Phone	Nexus One		3.7"	480x800	hdpi
Tablet	Nexus 6P		5.7"	1440x2...	560dpi
	Nexus 6		5.96"	1440x2...	560dpi
	Nexus 5X		5.2"	1080x1...	420dpi
	Nexus 5		4.95"	1080x1...	xxhdpi
	Nexus 4		4.7"	768x12...	xhdpi
	Galaxy Nexus		4.65"	720x12...	xhdpi
	5.4" FWVGA		5.4"	480x854	mdpi
	5.1" WVGA		5.1"	480x800	mdpi

New Hardware Profile Import Hardware Profiles



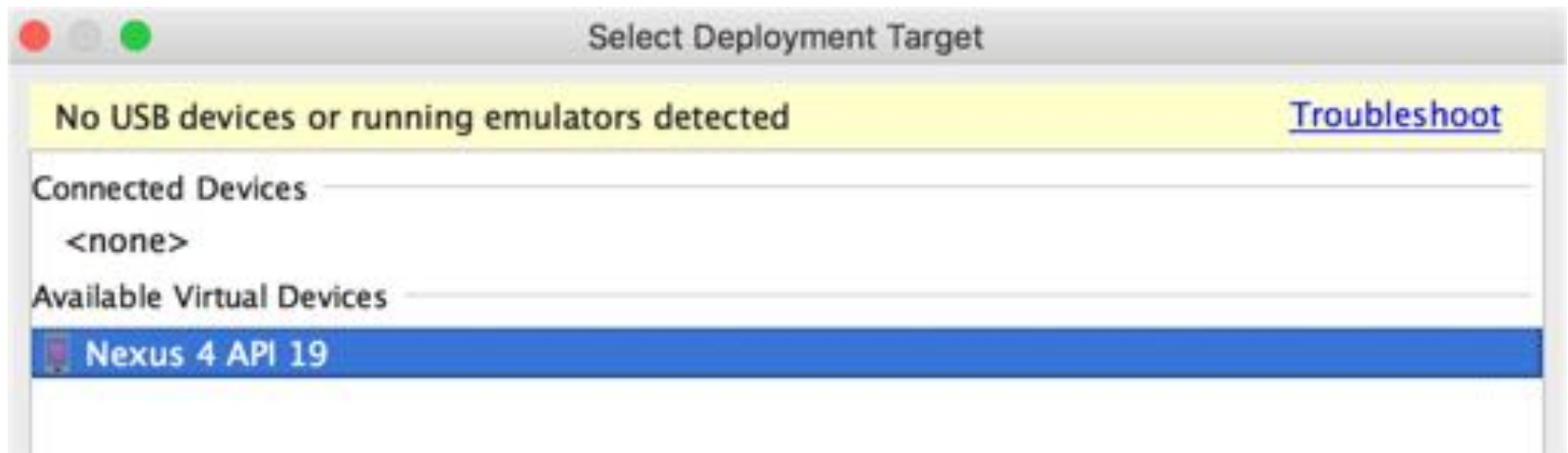
Select a system image

Recommended **x86 Images** Other Images

Release Name	API Level	ABI	Target
Lollipop Download	22	x86	Android 5.1
Lollipop Download	22	x86_64	Android 5.1
Lollipop Download	21	x86	Android 5.0 (Google APIs)
Lollipop Download	21	x86_64	Android 5.0 (Google APIs)
Lollipop Download	21	x86_64	Android 5.0
Lollipop Download	21	x86	Android 5.0
KitKat Download	19	x86	Android 4.4 (Google APIs)
KitKat Download	19	x86	Android 4.4
Jelly Bean Download	18	x86	Android 4.3 (Google APIs)
Jelly Bean Download	18	x86	Android 4.3
Jelly Bean Download	17	x86	Android 4.2 (Google APIs)
Jelly Bean Download	17	x86	Android 4.2
Jelly Bean Download	16	x86	Android 4.1 (Google APIs)
Jelly Bean	16	x86	Android 4.1

Execução

- Ao executar o app, o Android Studio permite a seleção de qual dispositivo irá realizar o download e instalação do app, para depois executa-lo.



Primeiro Projeto: FirstApp

- Ao escolher o virtual o emulador irá carregar (processo demorado).
 - Não feche mais o emulador, não é necessário.



Primeiro Projeto: FirstApp

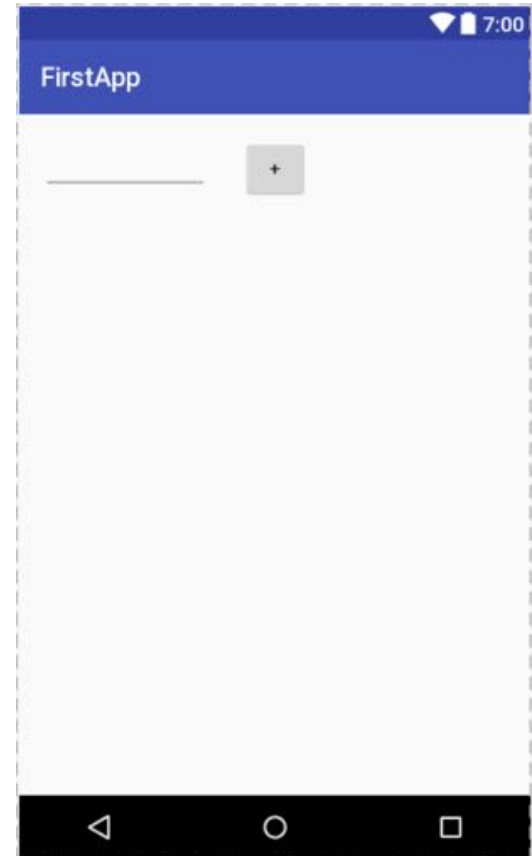
- Vamos modificar o FirstApp para fazermos uma app simples que recebe 2 números inteiros e os some.
- Vamos começar construindo a interface, ou seja, vamos começar trabalhando na View.
- Abra o xml no modo texto para alterarmos o tipo de layout.

```
<?xml version="1.0" encoding="utf-8"?>
<android.widget.RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.ale.firstapp.MainActivity"
    tools:layout_editor_absoluteY="81dp"
    tools:layout_editor_absoluteX="0dp">
```

```
</android.widget.RelativeLayout>
```

Primeiro Projeto: FirstApp

- Volte para o editor visual.
 - Inclua 1 Text Field (Number) e mude sua propriedade layout:width para 125dp, conforme a figura ao lado.
 - Mude a propriedade id de editText para num1.
 - Acrescente um botão:
 - Mude text para o símbolo +
 - Mude layout:width para 50 dp
 - Acrescente a chamada ao método sum, a ser criado, na propriedade onclick



Primeiro Projeto: FirstApp

- Acrescente outro text field numérico.
- Mude seu id para num2.
- Por fim, acrescente um Text View.
- Mude seu id para output.
- Mude a textAppearance para Material.Medium
- Posicione conforme a figura ao lado.



Primeiro Projeto: FirstApp

- Neste momento a View está pronta.
 - Temos os 2 campos de entrada de dados: num1 e num2.
 - Temos 1 botão para gerar a ação da soma, chamando o método sum
 - Temos 1 text view para preencheremos com a resposta ao somarmos os 2 número de entrada.
- Passamos agora ao Controller.

Primeiro Projeto: FirstApp

- Abra o arquivo MainActivity.java.
- Note que o método onCreate já está criado.
 - onCreate – chamado assim que a View é carregada, podemos inicializar objetos e fazer algumas operações iniciais neste método.
- Execute o app para ver a interface montada no emulador.



Primeiro Projeto: FirstApp

- Mapeando componentes GUI da View para objetos no controller:
 - Para podermos manipular os componentes de interface, temos que mapeá-los como objetos para dentro do nosso controller.
 - O mapeamento pode ser global a classe ou local a um método, dependendo da necessidade do escopo desejado.
 - O método `findViewById` é responsável por mapear um componente de interface para um objeto no controller.
 - Ele mapeia usando o ID que nomeia o componente, para um objeto de mesmo tipo no controller.
 - Este método usa a classe `R` como intermediária para auxiliar no mapeamento.

Primeiro Projeto: FirstApp

- A classe R
 - Esta classe conhece todos os componentes da interface, ou seja, conhece tudo o que uma View contém.
 - Além disto, ela também conhece todos os recursos do projeto, sejam eles arquivos de imagens, de sons, de vídeos ou menus.
 - Enfim, ela mapeia tudo o que projeto contém em termos de recursos, independente do número de Views e arquivos.
 - Esta classe é gerada automaticamente toda a vez que uma View é modificada e gravada ou quando se insere novos recursos no projeto.
 - Caso uma View tenha algum erro no seu XML, a classe R não é atualizada.

Primeiro Projeto: FirstApp

- Vamos mapear o text view output com escopo global, já que iremos precisar dele mais de um método.

```
public class MainActivity extends ActionBarActivity {  
    TextView out;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        out = (TextView)findViewById(R.id.output);  
    }  
}
```

- O objeto out agora possui escopo global e a linha `TextView out = ...`, no `onCreate`, faz o mapeamento.
 - Estamos mapeando um componente de interface que é um Text View, então criamos um objeto TextView e fazemos o casting.
 - O nome do objeto e do componente GUI pode ou não ser o mesmo. É uma escolha do desenvolvedor.

Primeiro Projeto: FirstApp

- Agora podemos manipular o componente output através do objeto out.
- Lembre que o output está mostrando um texto “Medium Text” ao rodarmos o app, vamos começar por inicializar este componente com texto vazio.
- O método onCreate, que é chamado ao executarmos o app, é um bom lugar para inicializarmos o componente.
- Modifique o onCreate e execute o app.

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    out = (TextView)findViewById(R.id.output);  
    out.setText("");  
}
```

Primeiro Projeto: FirstApp

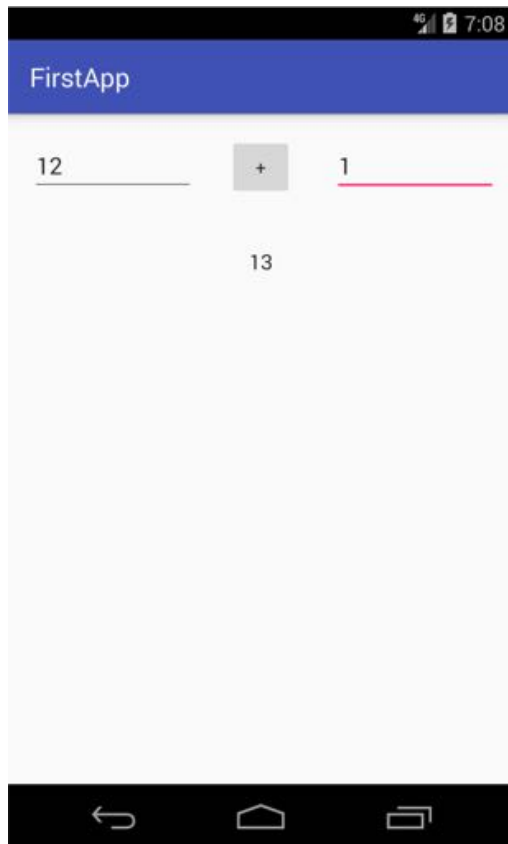
- O próximo passo é criarmos o método sum, que é chamado pela propriedade onClick de nosso botão, ou seja, ao ser clicado, o botão chama o método sum.
- Crie o método sum, que pode ser colocado logo abaixo do onCreate, conforme código abaixo.

```
public void sum(View view) {  
    // Componentes GUI x Objetos  
    EditText num1 = (EditText)findViewById(R.id.num1);  
    EditText num2 = (EditText)findViewById(R.id.num2);  
  
    //Captura número digitado e converte para inteiro  
    int n1 = Integer.parseInt(num1.getText().toString());  
    int n2 = Integer.parseInt(num2.getText().toString());  
  
    //Soma e escreve na interface o resultado  
    out.setText(String.valueOf(n1+n2));  
}
```

- Note que é necessário receber uma view no método, já que ele é chamado pelo clique do botão.

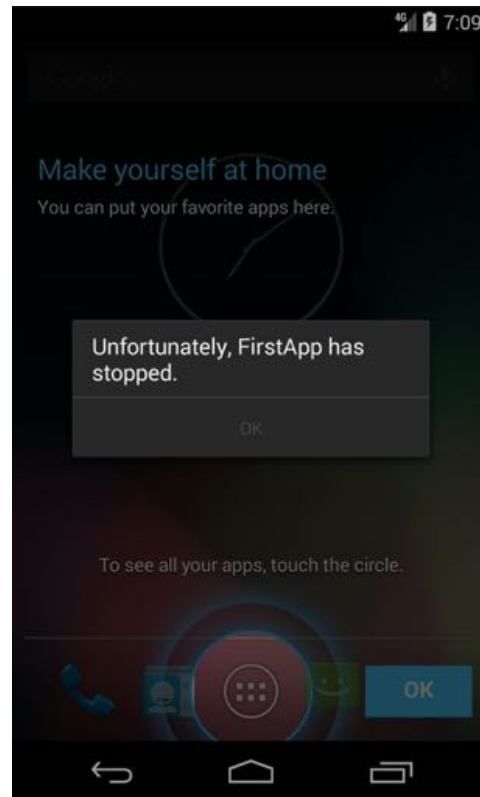
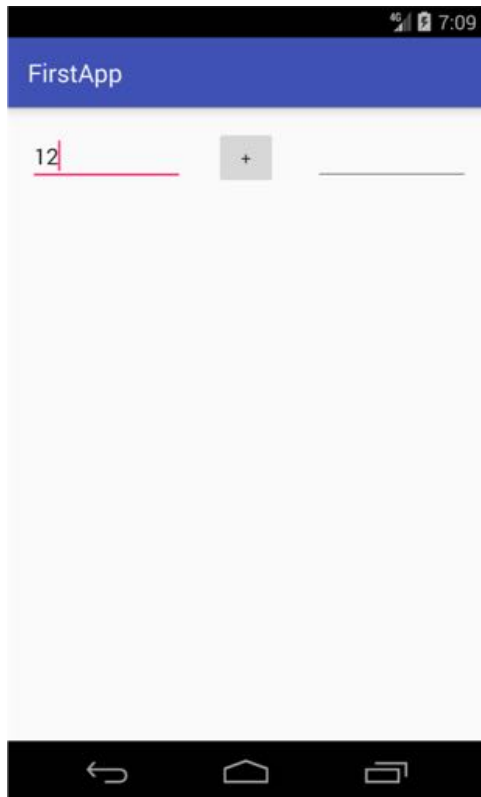
Primeiro Projeto: FirstApp

- Execute o app.



Primeiro Projeto: FirstApp

- Note que se deixarmos um campo de entrada em branco, teremos um erro.



Primeiro Projeto: FirstApp

- Podemos corrigir o erro no código.

```
public void sum(View view) {  
    EditText num1 = (EditText)findViewById(R.id.num1);  
    EditText num2 = (EditText)findViewById(R.id.num2);  
  
    if(num1.length() !=0 && num2.length() !=0 ) {  
        int n1 = Integer.parseInt(num1.getText().toString());  
        int n2 = Integer.parseInt(num2.getText().toString());  
        out.setText(String.valueOf(n1 + n2));  
    }  
}
```

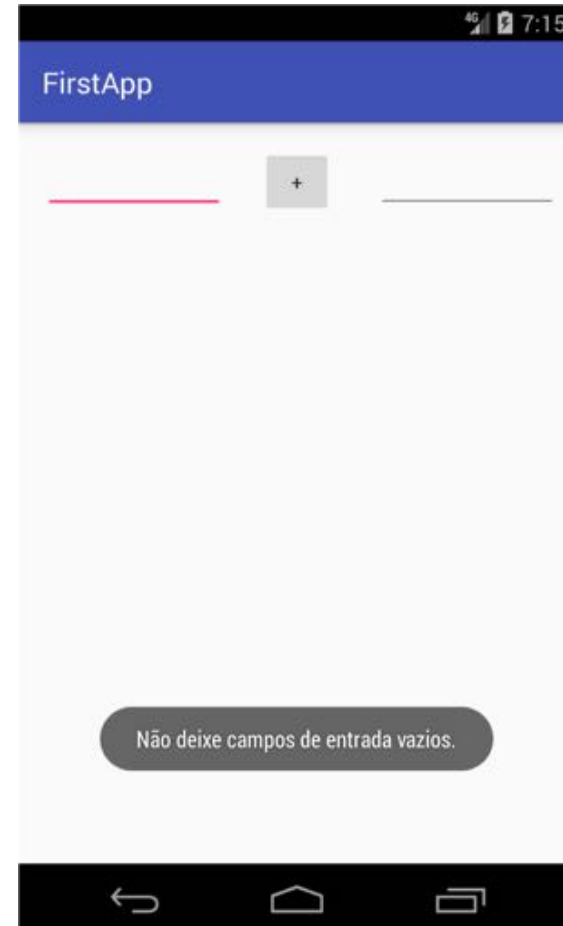
Primeiro Projeto: FirstApp

- Podemos informar nosso usuário do erro com um Toast.

```
public void sum(View view) {  
    EditText num1 = (EditText)findViewById(R.id.num1);  
    EditText num2 = (EditText)findViewById(R.id.num2);  
  
    if(num1.length() !=0 && num2.length() !=0 ) {  
        int n1 = Integer.parseInt(num1.getText().toString());  
        int n2 = Integer.parseInt(num2.getText().toString());  
        out.setText(String.valueOf(n1 + n2));  
    } else {  
        Toast msg =  
            Toast.makeText(this, "Não deixe campos de entrada vazios.", Toast.LENGTH_SHORT);  
        msg.show();  
    }  
}
```

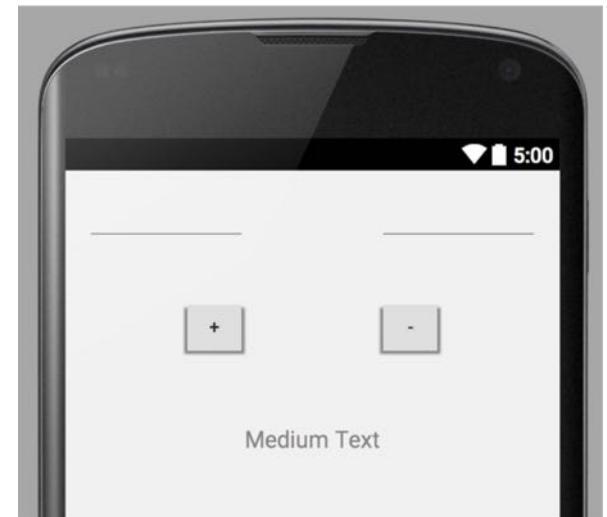
Primeiro Projeto: FirstApp

- Podemos informar nosso usuário do erro com um Toast.



Primeiro Projeto: FirstApp

- Incrementando um pouco mais o app.
- Vamos acrescentar mais um botão para subtrair.
- Mude a propriedade onClick deste novo botão para também chamar o método sum.
- Mude o id do botão soma para bSum e do subtrair para bSub.
- Rearranje a interface conforme a figura ao lado.



Primeiro Projeto: FirstApp

- Modifique o método sum para descobrir qual dos botões foi clicado e realizar a operação certa.
- Como descobrir?
 - Pelo id do botão.

```
public void sum(View view) {  
    // Componentes GUI x Objetos  
    EditText num1 = (EditText)findViewById(R.id.num1);  
    EditText num2 = (EditText)findViewById(R.id.num2);  
  
    //Captura número digitado e converte para inteiro  
    int n1 = Integer.parseInt(num1.getText().toString());  
    int n2 = Integer.parseInt(num2.getText().toString());  
  
    if(view.getId() == R.id.bSum)  
        //Soma e escreve na interface o resultado  
        out.setText(String.valueOf(n1+n2));  
    else if(view.getId() == R.id.bSub)  
        //Subtrai e escreve na interface o resultado  
        out.setText(String.valueOf(n1-n2));  
}
```