

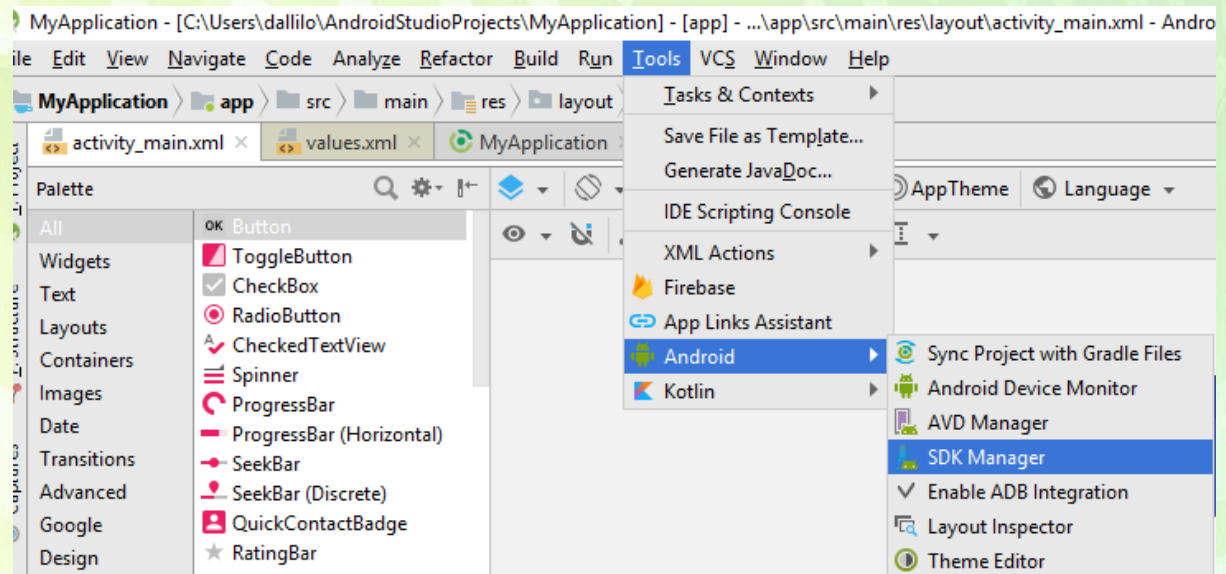
DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS Parte II

PROF. MSC. FELIPE DINIZ DALLILO



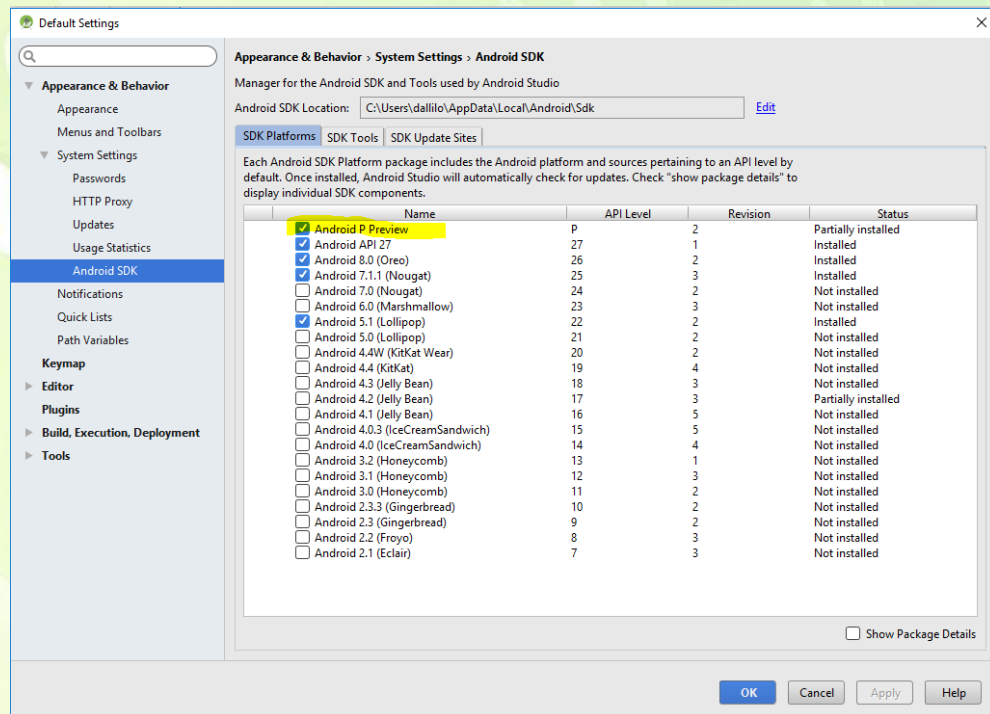
Activity_main.xml

- Um plugin do Android facilita a implementação das Interfaces com o usuário.
- Acesse o menu: Tools > Android > SDK Manager
- E selecione a opção **“Android P Preview”**



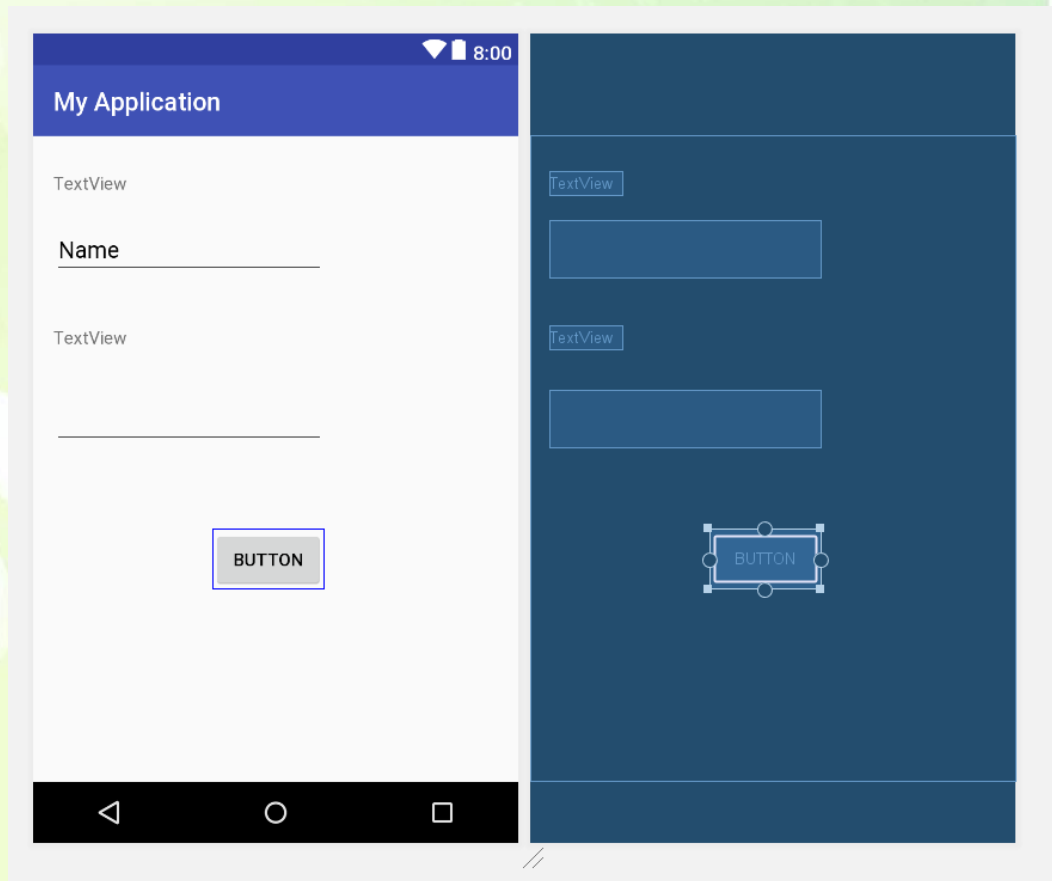
Android p preview

Ao seleccionar qualquer opção da SDK Plataforms (que não estava seleccionada anteriormente), será efetuado um download e posteriormente instalado.



Activity_main.xml

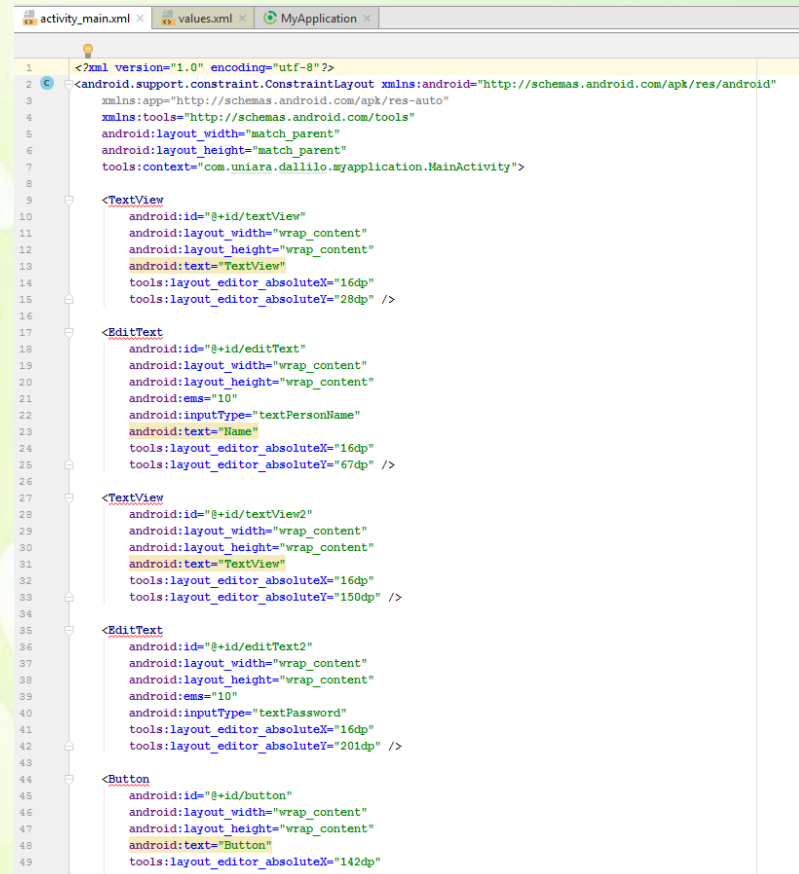
- Com o plugin, é possível visualizar o resultado final no smartphone, como apresentado na imagem ao lado:
- Para o mesmo, foram adicionados (Drag & Drop) dois componentes de Text View, dois componentes de EditText e um botão.



Atividade XML

Define o Layout da Atividade em XML.

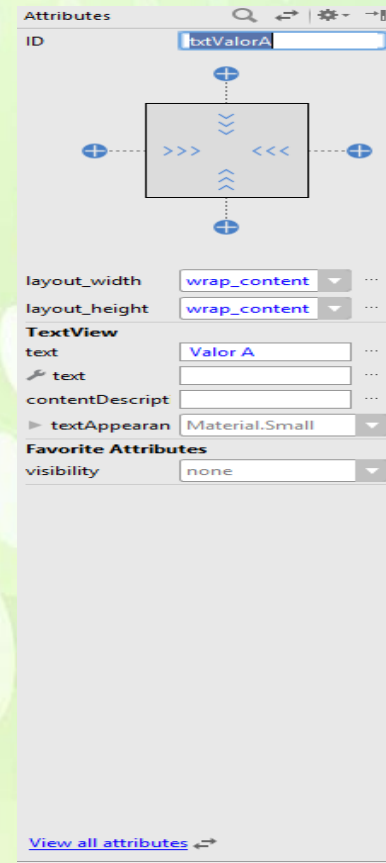
Cada atributo possui uma tag e as mesmas podem estar contidas (Semelhante ao HTML)



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context="com.uniarra.dalilio.myapplication.MainActivity">
8
9     <TextView
10         android:id="@+id/textView"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="TextView"
14         tools:layout_editor_absoluteX="16dp"
15         tools:layout_editor_absoluteY="28dp" />
16
17     <EditText
18         android:id="@+id/editText"
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:ems="10"
22         android:inputType="textPersonName"
23         android:text="Name"
24         tools:layout_editor_absoluteX="16dp"
25         tools:layout_editor_absoluteY="67dp" />
26
27     <TextView
28         android:id="@+id/textView2"
29         android:layout_width="wrap_content"
30         android:layout_height="wrap_content"
31         android:text="TextView"
32         tools:layout_editor_absoluteX="16dp"
33         tools:layout_editor_absoluteY="150dp" />
34
35     <EditText
36         android:id="@+id/editText2"
37         android:layout_width="wrap_content"
38         android:layout_height="wrap_content"
39         android:ems="10"
40         android:inputType="textPassword"
41         tools:layout_editor_absoluteX="16dp"
42         tools:layout_editor_absoluteY="201dp" />
43
44     <Button
45         android:id="@+id/button"
46         android:layout_width="wrap_content"
47         android:layout_height="wrap_content"
48         android:text="Button"
49         tools:layout_editor_absoluteX="142dp"
50         tools:layout_editor_absoluteY="231dp" />
```

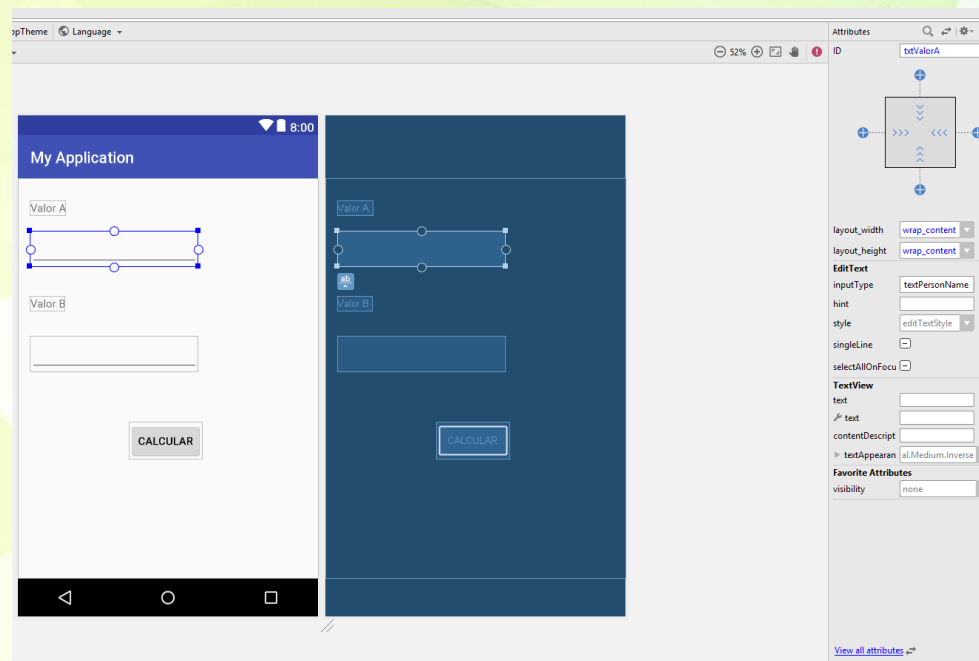
Propriedades dos componentes

- Como visto no Slide 2, ao clicar e arrastar o componente, uma tag XML é criada para ele e diversos atributos são definidos automaticamente.
- Os mesmos podem ser alterados diretamente no xml ou pelo menu “attributes”



Propriedades dos componentes

- Ao selecionar um componente, é alterado a aba de atributos automaticamente.



Text X Design

The image shows the Android Studio IDE with the XML editor on the left and the design preview on the right. The XML editor displays the layout for `activity_main.xml`, which includes a `ConstraintLayout` containing several `TextView` and `EditText` components, and a `Button`.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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.uniara.dallilo.myapplication.MainActivity">

    <TextView
        android:id="@+id/lblValorA"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Valor A"
        tools:layout_editor_absoluteX="16dp"
        tools:layout_editor_absoluteY="28dp" />

    <EditText
        android:id="@+id/txtValorA"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        tools:layout_editor_absoluteX="16dp"
        tools:layout_editor_absoluteY="67dp" />

    <TextView
        android:id="@+id/lblValorB"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Valor B"
        tools:layout_editor_absoluteX="16dp"
        tools:layout_editor_absoluteY="150dp" />

    <EditText
        android:id="@+id/txtValorB"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPassword"
        tools:layout_editor_absoluteX="16dp"
        tools:layout_editor_absoluteY="201dp" />

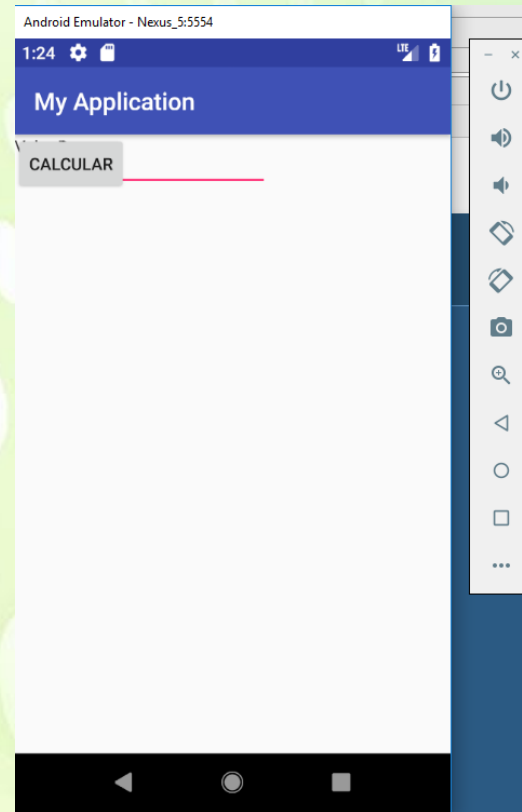
    <Button
        android:id="@+id/btnCalcular"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Calcular"
        tools:layout_editor_absoluteX="142dp"
        tools:layout_editor_absoluteY="311dp" />

</android.support.constraint.ConstraintLayout>
```

The design preview on the right shows the visual representation of the layout on a Nexus 4 device. It features a blue header with the title "My Application". Below the header, there are two text labels: "Valor A" and "Valor B". Under "Valor A" is a text input field. Under "Valor B" is a password input field. At the bottom right is a button labeled "CALCULAR". The status bar at the top shows the time as 8:00 and battery level at 59%.

Rebuild and Run

- Visão do Layout ao executar o Android.



Componentes Sobrepostos

- Ao definir uma atividade na última versão do android, é selecionado por padrão o componente de Layout “android.support.constraint.ConstraintLayout”, no entanto, com o mesmo, os componentes acabam ficando sobrepostos.

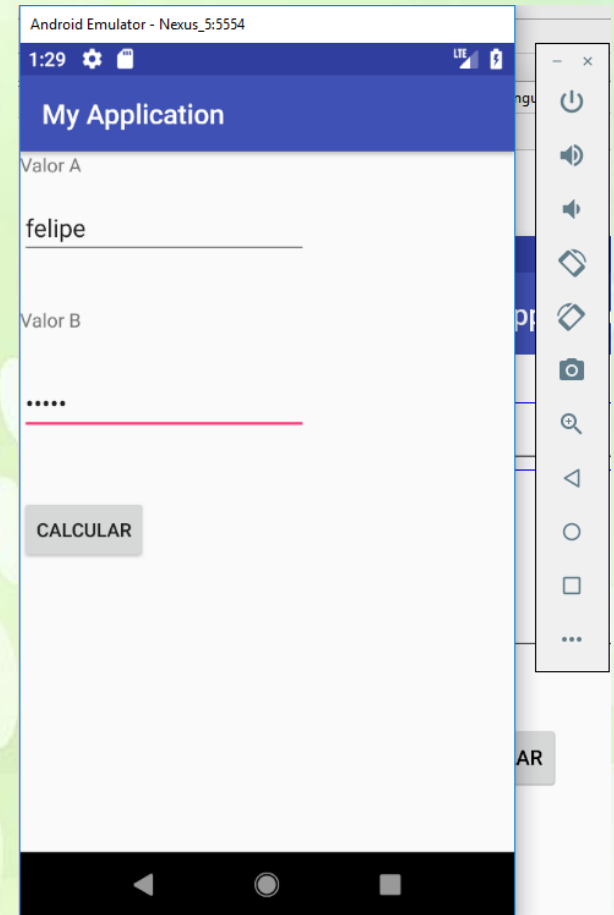
Relative Layout

- Ao alterar o layout do activity_main para “RelativeLayout”, é possível organizar da forma correta os componentes do formulário.

```
<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.uniara.dallilo.myapplication.MainActivity">

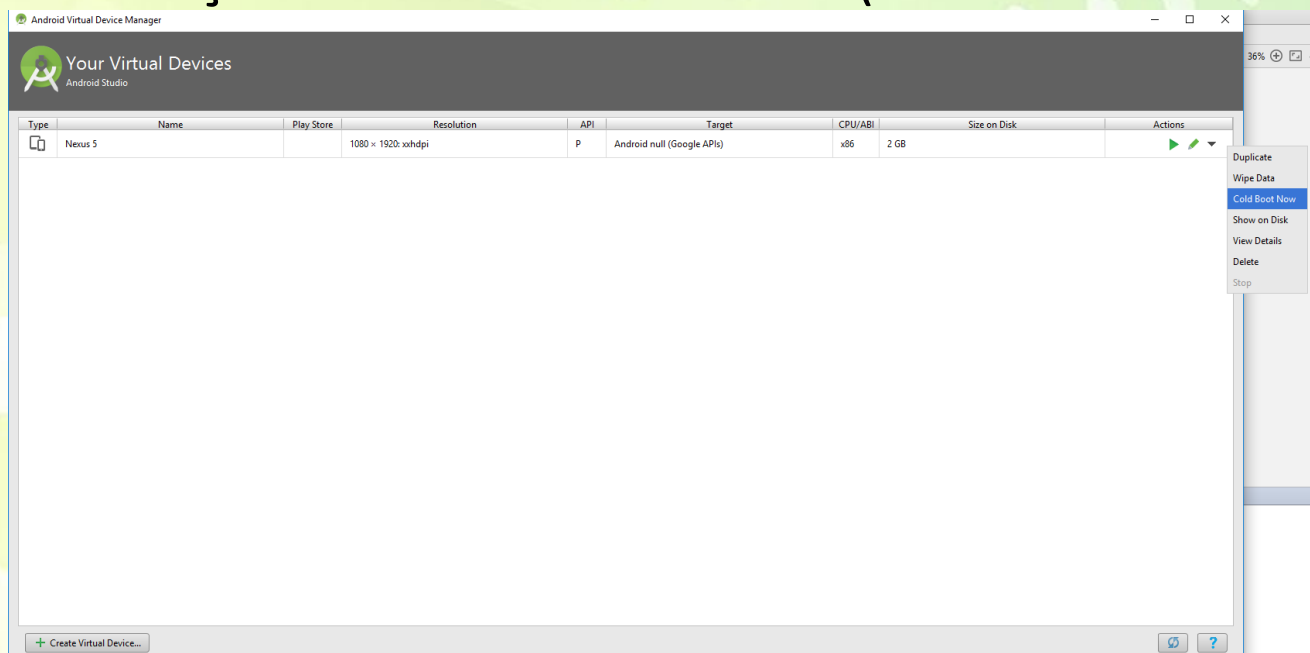
    <TextView
        android:id="@+id/lblValorA"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Valor A"
        tools:layout_editor_absoluteX="16dp"
        tools:layout_editor_absoluteY="39dp" />

    <EditText
        android:id="@+id/txtValorA"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```



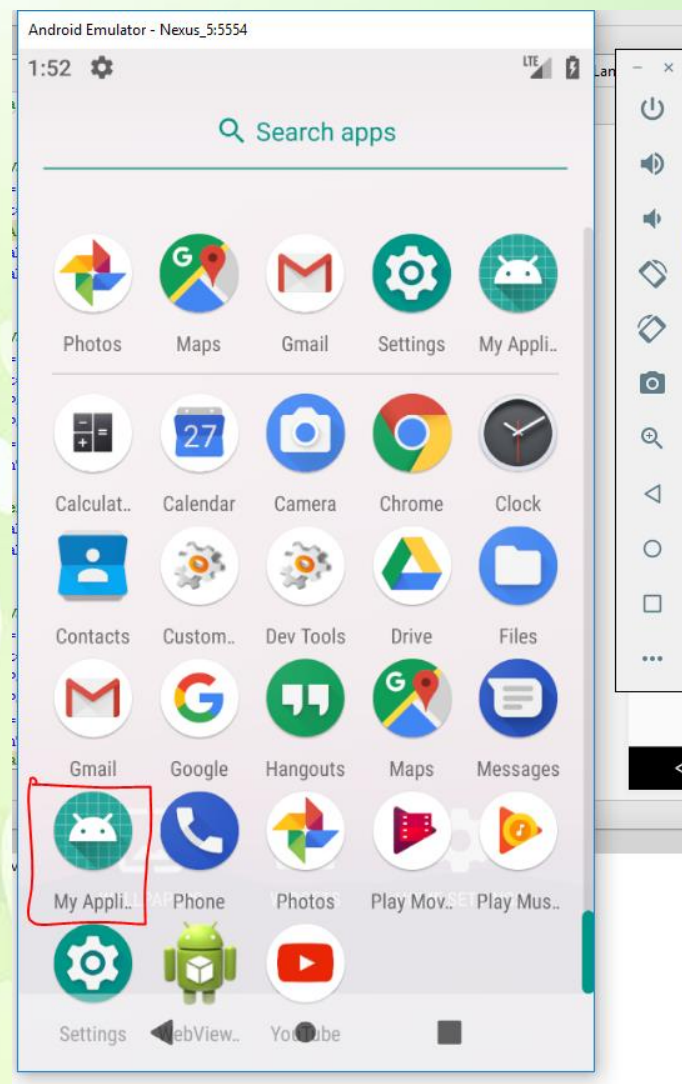
Problema ao dar boot no emulador

Se ao emular seu APP, permacer com a versão anterior, uma opção de emulação é o “Cold Boot Now” (Iniciar o android inteiro)



Acessar APP

Caso o aplicativo não seja iniciado automaticamente, é criado um ícone para o seu aplicativo onde é possível acessar pela navegação convencional do Android.



Adicionar implementação ao formulário

- Na classe MainActivity.java é possível prosseguir com o fluxo de desenvolvimento após as instruções definidas no método onCreate.

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```


Adicionar implementação ao formulario

- No exemplo abaixo, foi criado um método “addBotaoCalcularClickListener()” com o objetivo de implementar uma ação ao botão de calcular do formulário.

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

    addBotaoCalcularClickListener();
}
```

Adicionar implementação ao formulario

Através do método “findViewById” foi recuperado o componente botão e adicionado um Listener de click.

Na implementação do Listener, foi adicionado uma linha na API de log do Android.

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

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

        addBotaoCalcularClickListener();
    }

    public void addBotaoCalcularClickListener(){
        Button btnCalcular = (Button) findViewById(R.id.btnCalcular);
        btnCalcular.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {
                Log.i( tag: "Meu app", msg: "Meu botão foi clicado");
            }
        });
    }
}
```

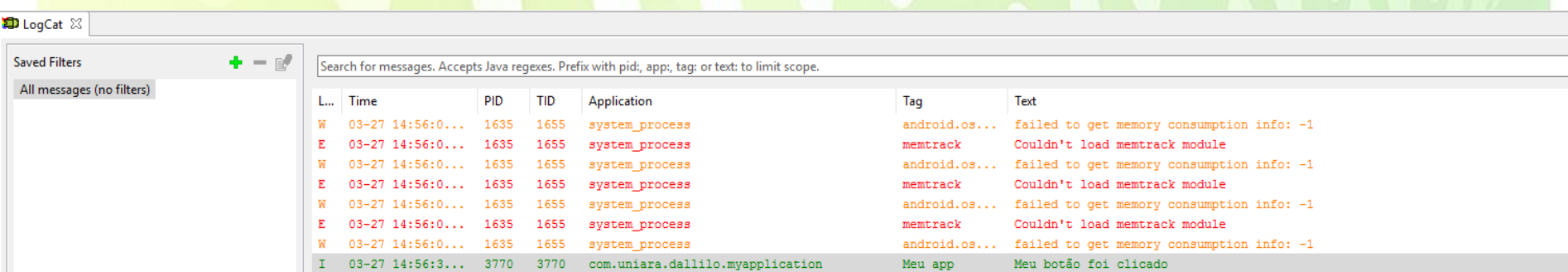
Log - Métodos

A prioridade é um dos seguintes valores:

- **V** — Detalhada (prioridade mais baixa)
- **D** — Depuração
- **I** — Informação
- **W** — Aviso
- **E** — Erro
- **A** — Afirmação

Android device monitor

Ao clicar no botão, uma linha de log é adicionado, informando o horário, PID, TID, a aplicação, Tag e o Texto.



LogCat

Search for messages. Accepts Java regexes. Prefix with pid; app; tag; or text; to limit scope.

L...	Time	PID	TID	Application	Tag	Text
W	03-27 14:56:0...	1635	1655	system_process	android.os...	failed to get memory consumption info: -1
E	03-27 14:56:0...	1635	1655	system_process	memtrack	Couldn't load memtrack module
W	03-27 14:56:0...	1635	1655	system_process	android.os...	failed to get memory consumption info: -1
E	03-27 14:56:0...	1635	1655	system_process	memtrack	Couldn't load memtrack module
W	03-27 14:56:0...	1635	1655	system_process	android.os...	failed to get memory consumption info: -1
E	03-27 14:56:0...	1635	1655	system_process	memtrack	Couldn't load memtrack module
W	03-27 14:56:0...	1635	1655	system_process	android.os...	failed to get memory consumption info: -1
I	03-27 14:56:3...	3770	3770	com.uniara.dallilo.myapplication	Meu app	Meu botão foi clicado

Adicionar implementação ao formulario

Também é possível adicionar um Toast para mostrar no dispositivo uma janela informativa

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

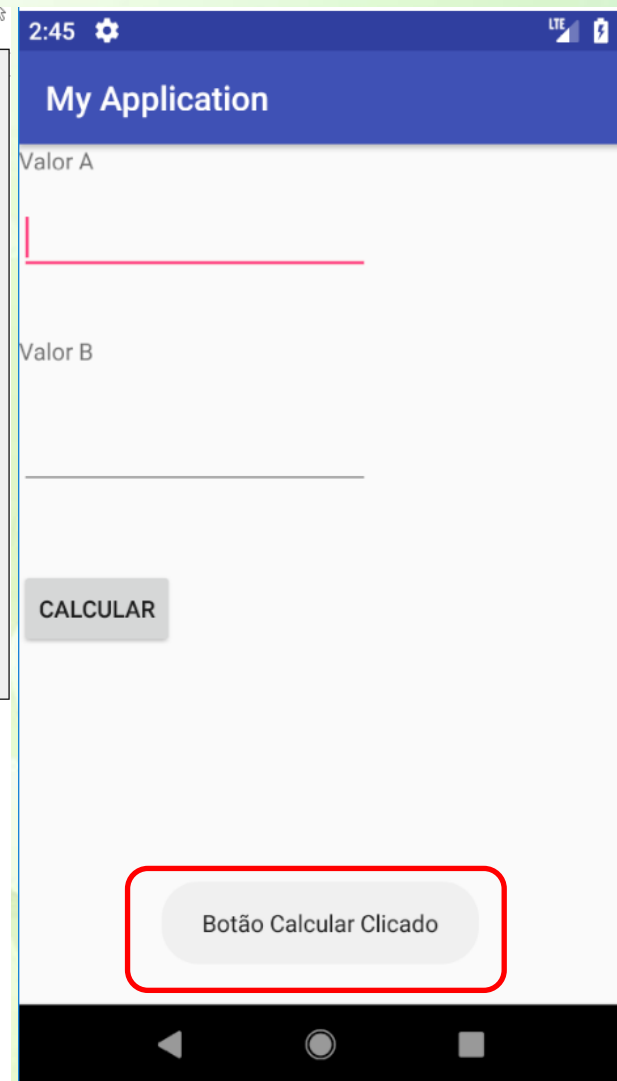
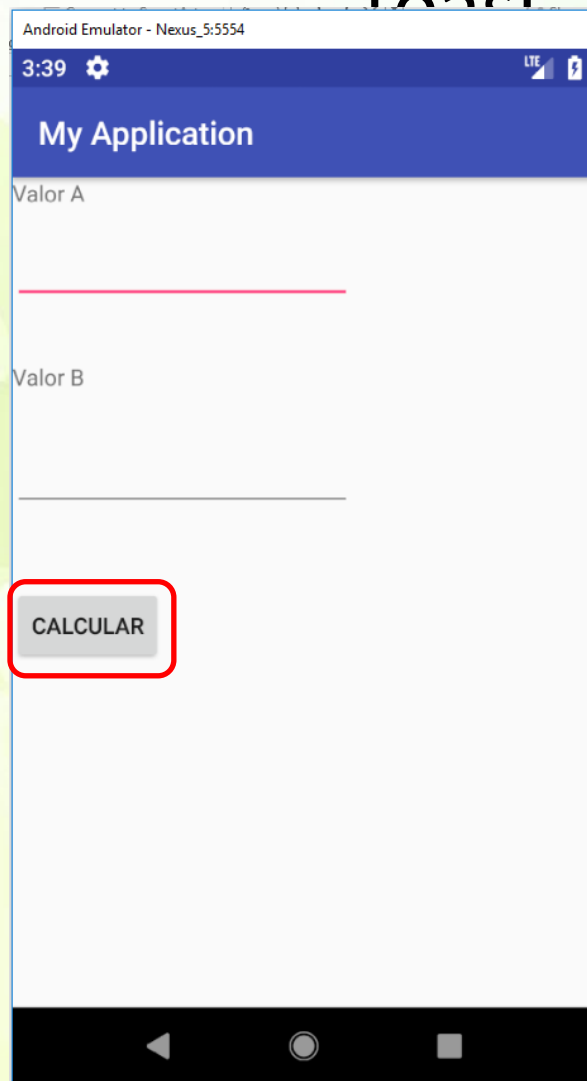
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        addBotaoCalcularClickListener();
    }

    public void addBotaoCalcularClickListener(){
        Button btnCalcular = (Button) findViewById(R.id.btnCalcular);
        btnCalcular.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {
                Log.i( tag: "Meu app", msg: "Meu botão foi clicado");
                Toast.makeText(getApplicationContext(), text: "Botão Calcular Clicado", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Ao clicar no botão calcular, a mensagem “Botão Calcular Clicado” aparece com sucesso no dispositivo.



Recuperar valor de um edittext

- Para recuperar o valor de um componente EditText, é necessário primeiramente recuperar o Widget no Java e posteriormente, chamar o método getText() e na sequência o método toString, como exemplificado no exemplo abaixo.

```
EditText txtValorA = (EditText) findViewById(R.id.txtValorA);  
String valorA = txtValorA.getText().toString();
```

Somar

- O código a seguir, coleta dois valores dos campos EditText, converte os mesmos para Int e efetuam uma soma, apresentando o resultado no Toast

```
public class MainActivity extends AppCompatActivity {

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

        addBotaoCalcularClickListener();
    }

    public void addBotaoCalcularClickListener(){
        Button btnCalcular = (Button) findViewById(R.id.btnCalcular);
        btnCalcular.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {
                Log.i( tag: "Meu app", msg: "Iniciado o processamento do botão Calcular");

                EditText txtValorA = (EditText) findViewById(R.id.txtValorA);
                EditText txtValorB = (EditText) findViewById(R.id.txtValorB);

                String valorA = getTextFromElement(txtValorA);
                String valorB = getTextFromElement(txtValorB);

                int intValA = Integer.parseInt(valorA);
                int intValB = Integer.parseInt(valorB);

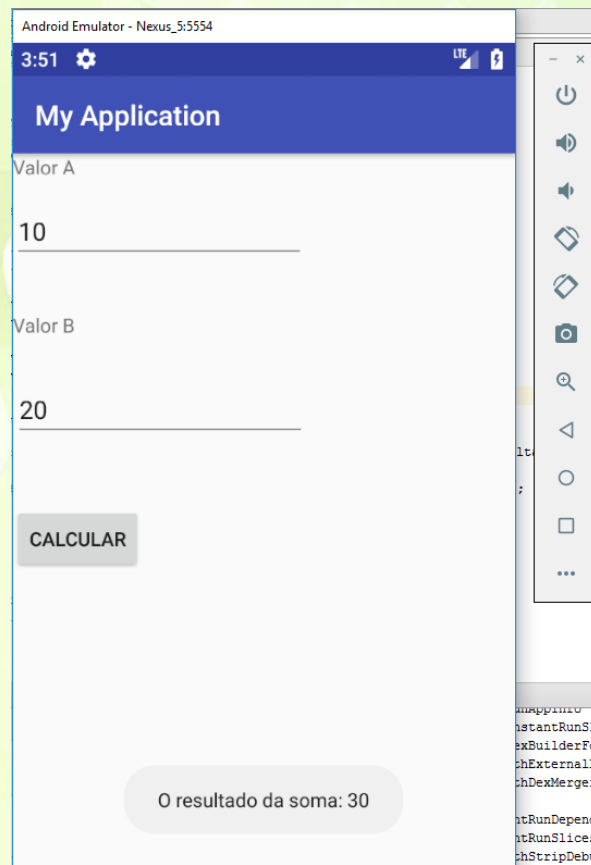
                int resultado = intValA + intValB;

                Toast.makeText(getApplicationContext(), text: "O resultado da soma: "+ resultado, Toast.LENGTH_SHORT).show();

                Log.i( tag: "Meu app", msg: "Finalizado o processamento do botão Calcular");
            }
        });
    }

    public String getTextFromElement(EditText campoTexto){
        return campoTexto.getText().toString();
    }
}
```

Resultado - Aplicação Somar



Referências

- SILVA, D. **Desenvolvimento para dispositivos móveis**. São Paulo: Pearson Education do Brasil, 2016.
- LEE, V., SCHNEIDER, H., SCHELL, R. **Aplicações móveis: arquitetura, projeto e desenvolvimento**. São Paulo: Pearson Education do Brasil, 2005.
- MEDNIEKS, Z; et al. **Programando o Android**. 2. ed. São Paulo: Novatec, 2012.
- DEITEL, P; DEITEL, H. **Java: como programar**. 8. ed. São Paulo: Pearson, 2010.
- DUARTE, W. **Delphi para Android e IOS Desenvolvendo Aplicações Móveis**. Rio de Janeiro: Brasport Livros e Multimídia, 2015.
- ABLESON, W. Frank; et al. **Android em ação**. 3. ed. Rio de Janeiro: Elsevier, 2012.
- LECHETA, R. R. **Google Android para Tablets. Aprenda a desenvolver aplicações para o Android**. São Paulo: Novatec, 2012.
- LECHETA, R. R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 2. ed. São Paulo: Novatec, 2010.
- ASCENCIO, A. F. G.; CAMPOS, E. A. V. de. **Fundamentos da programação de computadores: algoritmos, pascal, C/C ++ e Java**. 2. ed. São Paulo: Pearson, 2007