

<https://github.com/PacktPublishing/Android-Programming-with-Kotlin-for-Beginners>

# Android com Kotlin

Aula 02 - Views

## Android Programming with Kotlin for Beginners

Build Android apps starting from zero programming experience  
with the new Kotlin programming language

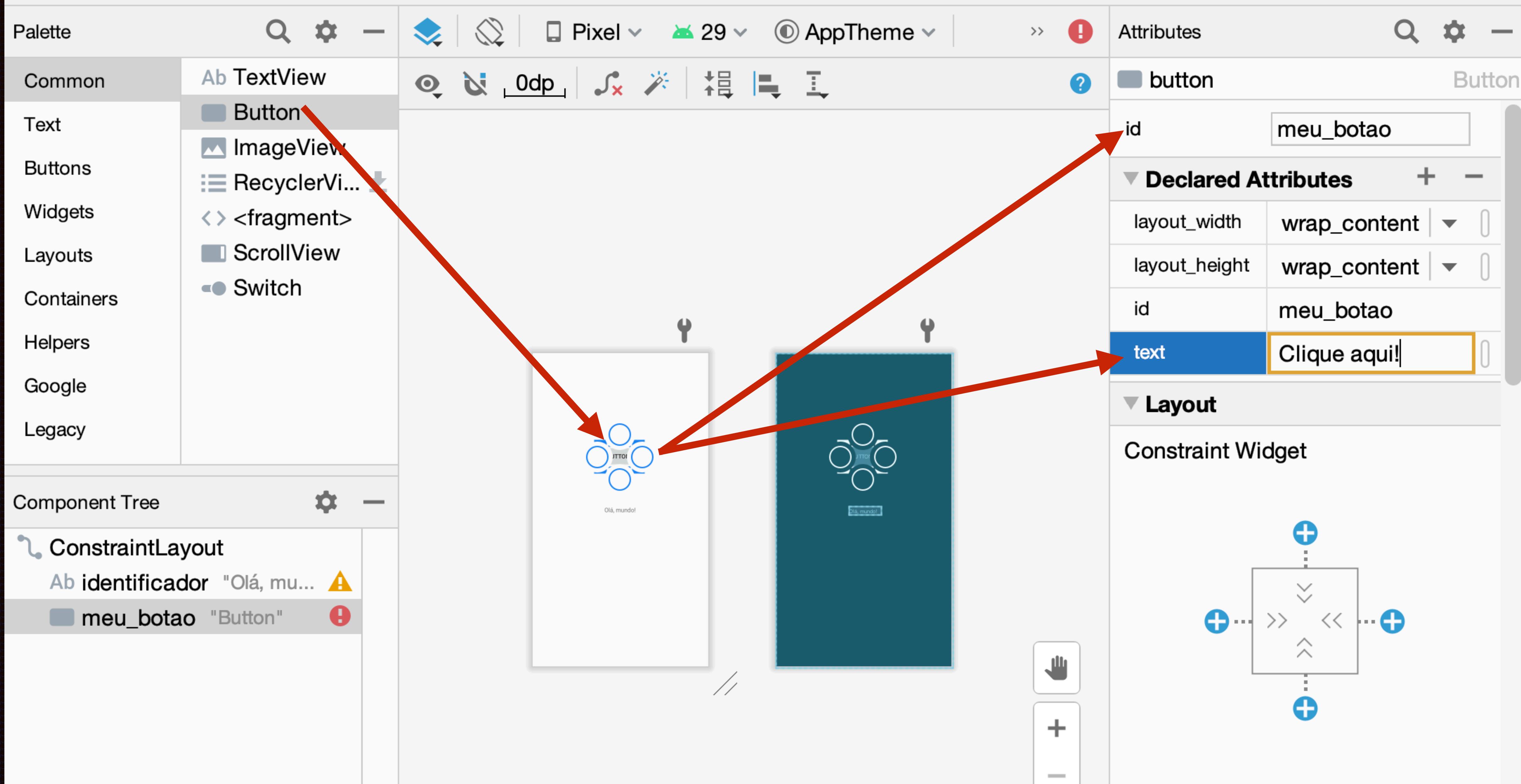


John Horton

# Aula 02: Views

- Como criar um botão em nosso App (Button)
- TextView
- ImageView
- Os Diferentes tipos de views

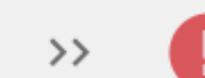
# COMO ADICIONAR UM BOTÃO AO APP?



Palette



Pixel 29 AppTheme



Attributes



Common

Ab TextView

Button

ImageView

RecyclerVi... ▾

&lt;&gt; &lt;fragment&gt;

ScrollView

Switch

Text

Buttons

Widgets

Layouts

Containers

Helpers

Google

Legacy

Component Tree

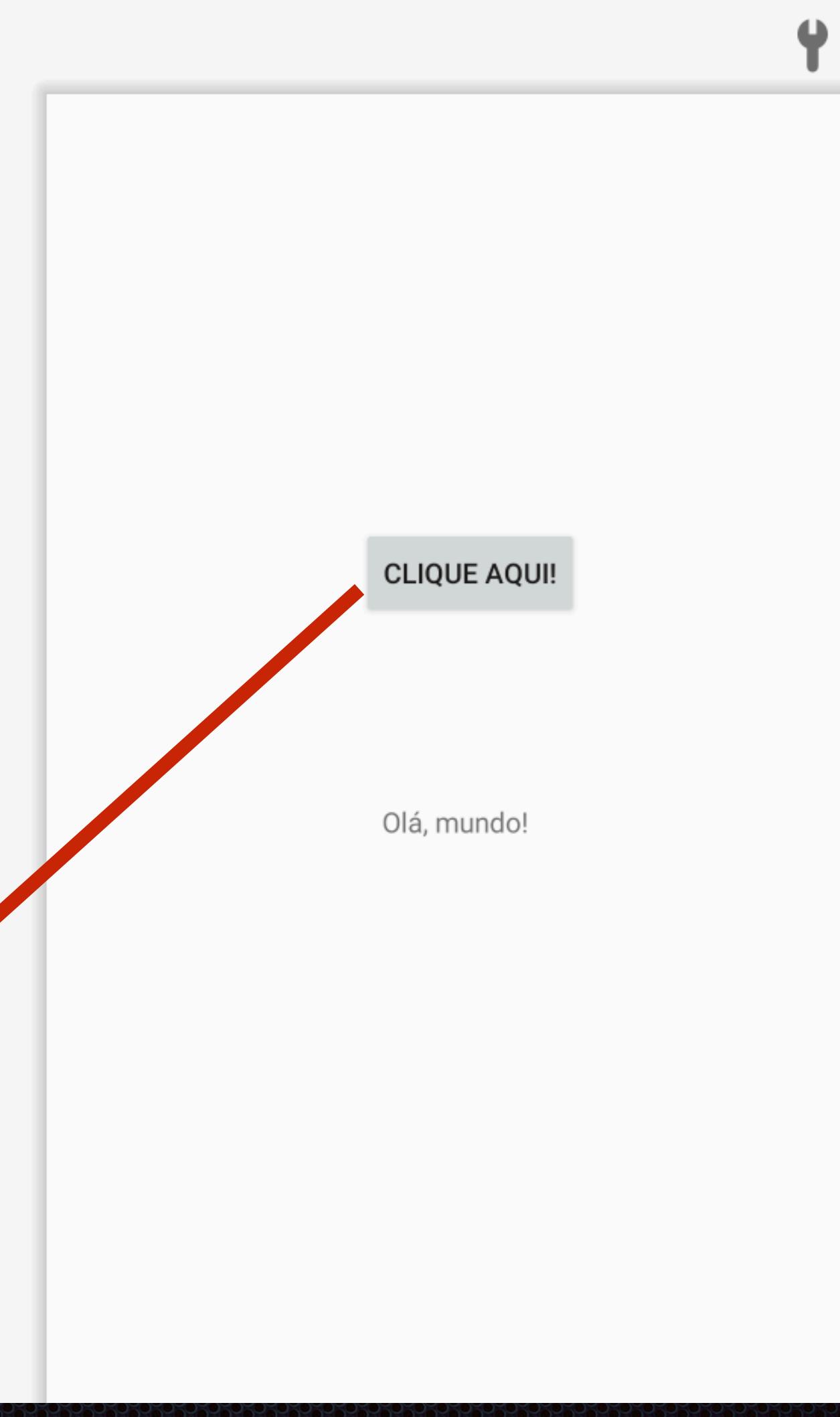


ConstraintLayout

Ab identificador "Olá, mu...



meu\_botao "Clique aq...



Attributes



&lt;unnamed&gt;

ConstraintLayout

id



## Declared Attributes

layout_width	match_parent
layout_height	match_parent
context	.MainActivity

## Layout

layout_width	match_parent
layout_height	match_parent
visibility	
context	

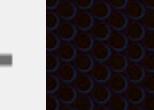
## Transforms



Palette



Attributes



Common

Ab TextView

Button

ImageView

RecyclerVi...  
↓

&lt;fragment&gt;

ScrollView

Switch

Text

Buttons

Widgets

Layouts

Containers

Helpers

Google

Legacy

Component Tree

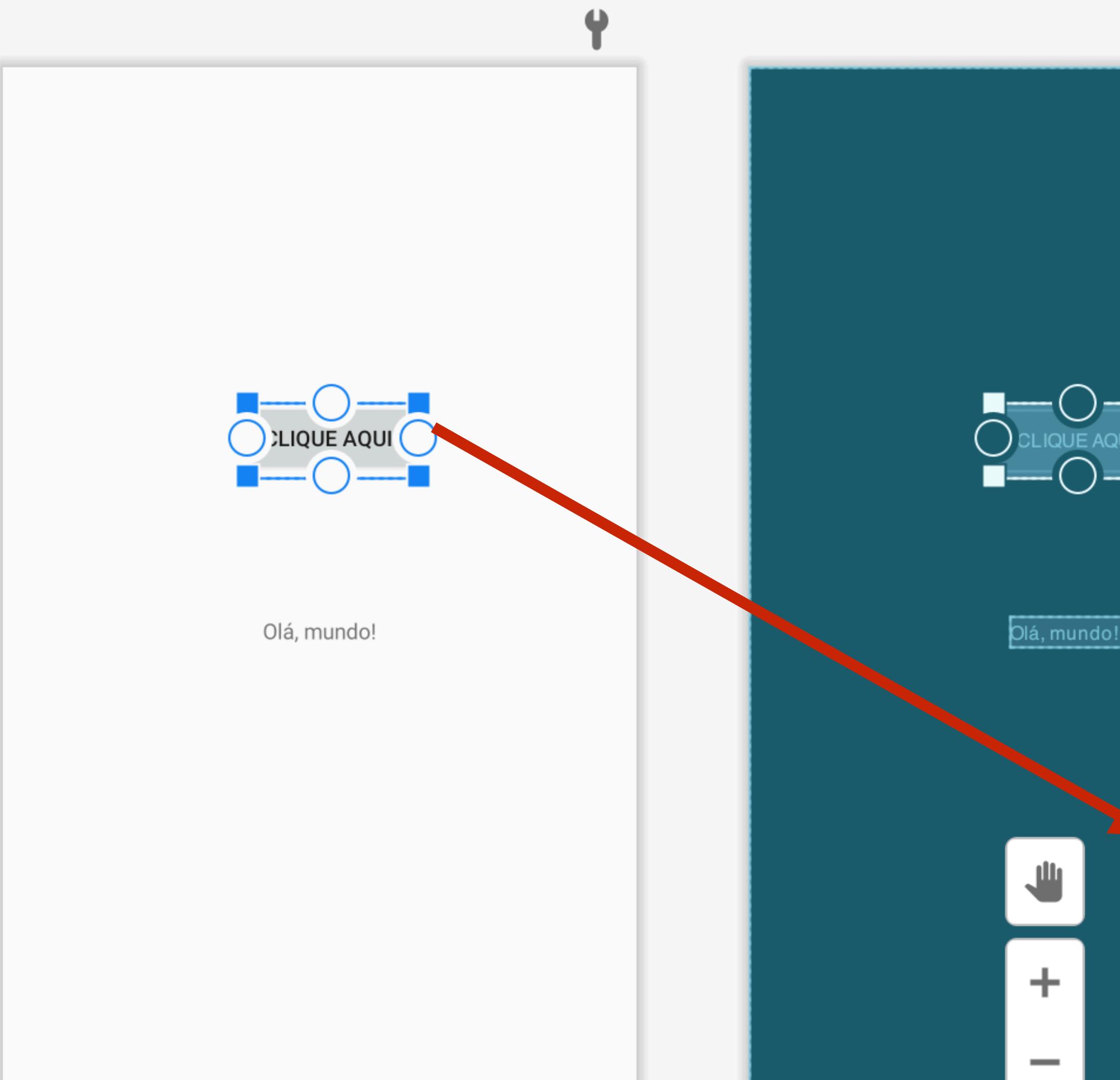


ConstraintLayout

Ab identificador "Olá, mu...



meu\_botao "Clique aq...



meu\_botao

Button

z

rotation

rotationX

rotationY

scaleX

scaleY

translationX

translationY

alpha

## ▼ Common Attributes

style

onClick

background

text

text

@android:style

fui\_clicado

@android:drawa

Clique aqui!

MainActivity.kt X activity\_main.xml X

Code Split Design

```
9 <TextView  
10    android:id="@+id/identificador"  
11    android:layout_width="wrap_content"  
12    android:layout_height="wrap_content"  
13    android:text="Olá, mundo!"  
14    app:layout_constraintBottom_toBottomOf="parent"  
15    app:layout_constraintLeft_toLeftOf="parent"  
16    app:layout_constraintRight_toRightOf="parent"  
17    app:layout_constraintTop_toTopOf="parent" />  
18  
19 <Button  
20    android:id="@+id/meu_botao"  
21    android:layout_width="wrap_content"  
22    android:layout_height="wrap_content"  
23    android:onClick="fui_clicado"  
24    android:text="Cliquei!"  
25    android:visibility="visible"  
26    tools:layout_editor_absoluteX="50dp"  
27    tools:layout_editor_absoluteY="50dp"  
28    tools:visibility="visible"/>
```

Corresponding method handler 'public void fui\_clicado(android.view.View)' not found

Provide feedback on this warning ↗ ↘ More actions... ↗ ↘

com.example.botao.MainActivity  
public final fun fui\_clicado(): Unit

PrimeiroApp.app

Como o método não foi criado ainda, precisa ser implementado na classe MainActivity (more actions).

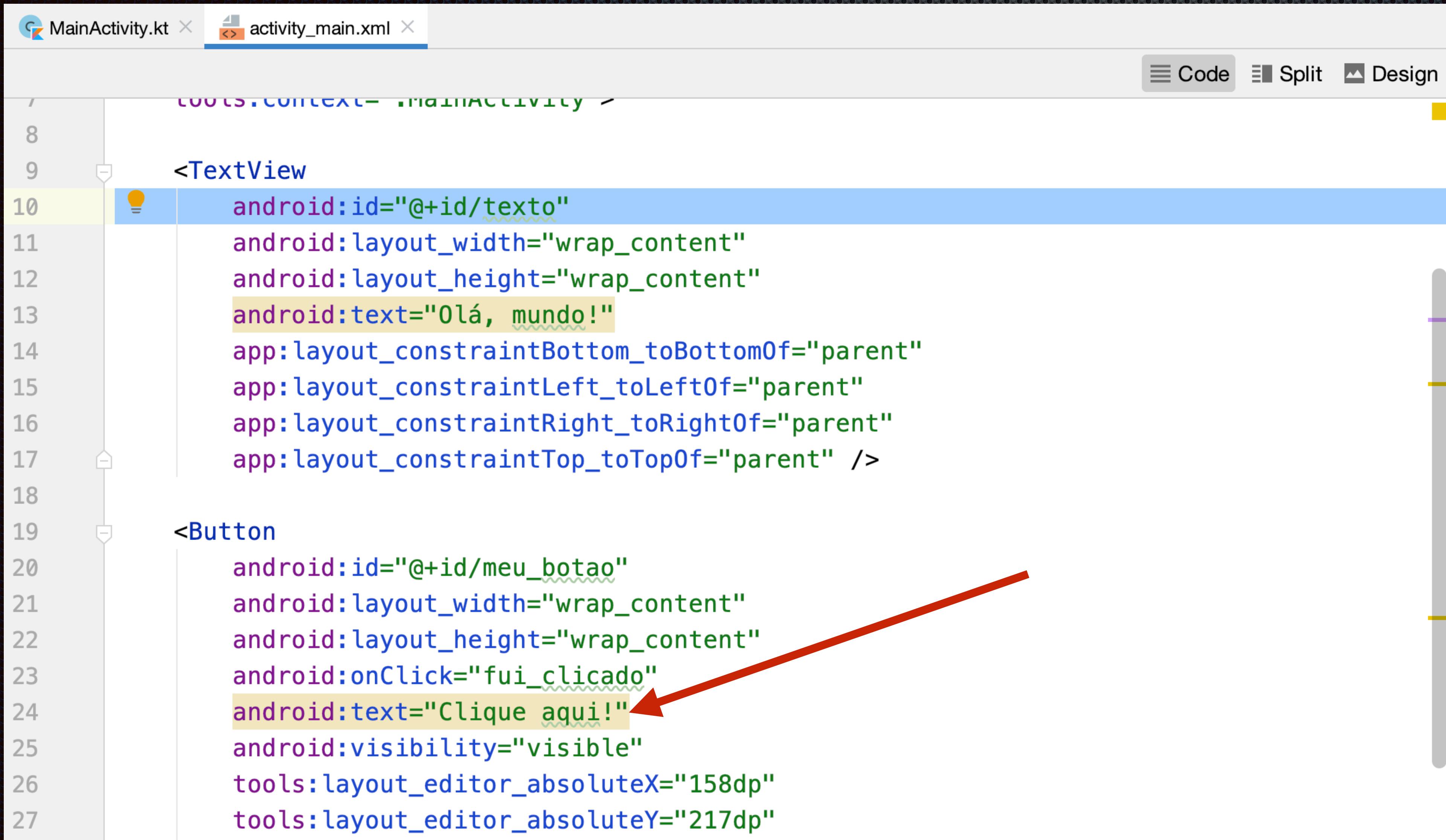
```
9      <TextView  
10         android:id="@+id/identificador"  
11         android:layout_width="wrap_content"  
12         android:layout_height="wrap_content"  
13         android:text="Olá, mundo!"  
14         app:layout_constraintBottom_toBottomOf="parent"  
15         app:layout_constraintLeft_toLeftOf="parent"  
16         app:layout_constraintRight_toRightOf="parent"  
17         app:layout_constraintTop_toTopOf="parent" />  
  
18  
19      <Button  
20         android:id="@+id/meu_botao"  
21         android:layout_width="wrap_content"  
22         android:layout_height="wrap_content"  
23         android:onClick="fui_clicado" ?  
24         android:text="Cli ? Provide feedback on this warning  
25         android:visibility="visible" X Suppress: Add tools:ignore="OnClick" attribute  
26         tools:layout_editor_absoluteX="50dp" ? Create 'fui_clicado(View)' in 'MainActivity'  
27         tools:layout_editor_absoluteY="50dp" ? Create onClick event handler ▶  
28         tools:visibility="visible" ? Override Resource in Other Configuration...
```

# Enquanto isto, em MainActivity....

The screenshot shows the Android Studio interface with the file `MainActivity.kt` open. The code defines a `MainActivity` class that extends `AppCompatActivity`. It overrides the `onCreate` method to set the content view to `R.layout.activity_main`. A new function `fui_clicado` is defined, which takes a `View` parameter and contains an empty body. A red arrow points from the cursor at the end of line 15 to the start of line 14, highlighting the function definition.

```
1 package com.example.botao
2
3 import ...
4
5
6
7 class MainActivity : AppCompatActivity() {
8
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12     }
13
14     fun fui_clicado(view: View) {
15     }
16 }
17
18 }
```

# Depois de criado o método, o erro deve desaparecer.



The screenshot shows the Android Studio interface with two tabs open: `MainActivity.kt` and `activity_main.xml`. The `activity_main.xml` tab is active, displaying the XML layout code for the main activity. The code defines a `TextView` and a `Button`. The `TextView` has its text set to "Olá, mundo!". The `Button` has its `onClick` attribute set to "fui\_clicado". A red arrow points from the text "fui\_clicado" to the `onClick` attribute of the `Button`.

```
1 <FrameLayout
2   android:layout_width="match_parent"
3   android:layout_height="match_parent">
4     android:background="#EAEAEA"
5
6     <TextView
7       android:id="@+id/texto"
8       android:layout_width="wrap_content"
9       android:layout_height="wrap_content"
10      android:text="Olá, mundo!"
11
12      app:layout_constraintBottom_toBottomOf="parent"
13      app:layout_constraintLeft_toLeftOf="parent"
14      app:layout_constraintRight_toRightOf="parent"
15      app:layout_constraintTop_toTopOf="parent" />
16
17
18     <Button
19       android:id="@+id/meu_botao"
20       android:layout_width="wrap_content"
21       android:layout_height="wrap_content"
22       android:onClick="fui_clicado"
23       android:text="Clique aqui!"
24       android:visibility="visible"
25
26       tools:layout_editor_absoluteX="158dp"
27       tools:layout_editor_absoluteY="217dp"
```

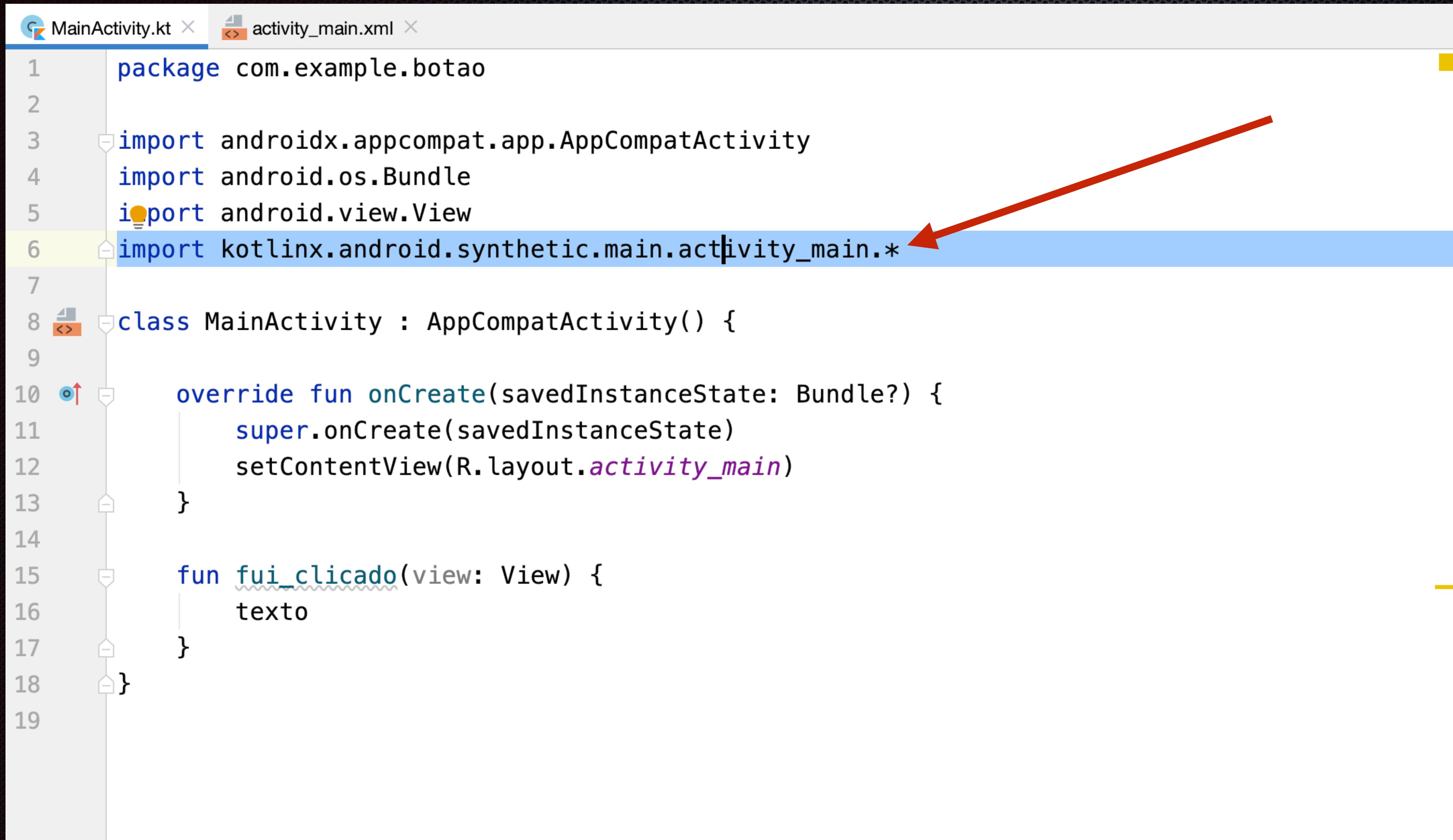
# Mas como podemos acessar um componente da view?

The screenshot shows the Android Studio interface with two tabs open: `MainActivity.kt` and `activity_main.xml`. The `MainActivity.kt` file contains the following code:

```
1 package com.example.botao
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.View
6
7 class MainActivity : AppCompatActivity() {
8
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12     }
13
14     fun fui_clicado(view: View) {
15         texto
16     }
17 }
18
```

A red arrow points from a tooltip to the `texto` variable in the `fui_clicado` function. The tooltip says "Unresolved reference: texto".

# Precisamos importá-lo primeiro...



The screenshot shows the Android Studio code editor with the file `MainActivity.kt` open. The code defines a `MainActivity` class that extends `AppCompatActivity`. It overrides the `onCreate` method to set the content view to `R.layout.activity_main`. It also contains a `fui_clicado` function that takes a `View` parameter and prints `texto`.

```
1 package com.example.botao
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.View
6 import kotlinx.android.synthetic.main.activity_main.*
7
8 class MainActivity : AppCompatActivity() {
9
10    override fun onCreate(savedInstanceState: Bundle?) {
11        super.onCreate(savedInstanceState)
12        setContentView(R.layout.activity_main)
13    }
14
15    fun fui_clicado(view: View) {
16        texto
17    }
18}
19
```

A red arrow points from the text "Precisamos importá-lo primeiro..." at the top of the slide to the `import kotlinx.android.synthetic.main.activity_main.*` line in the code editor.

# Agora sim.....



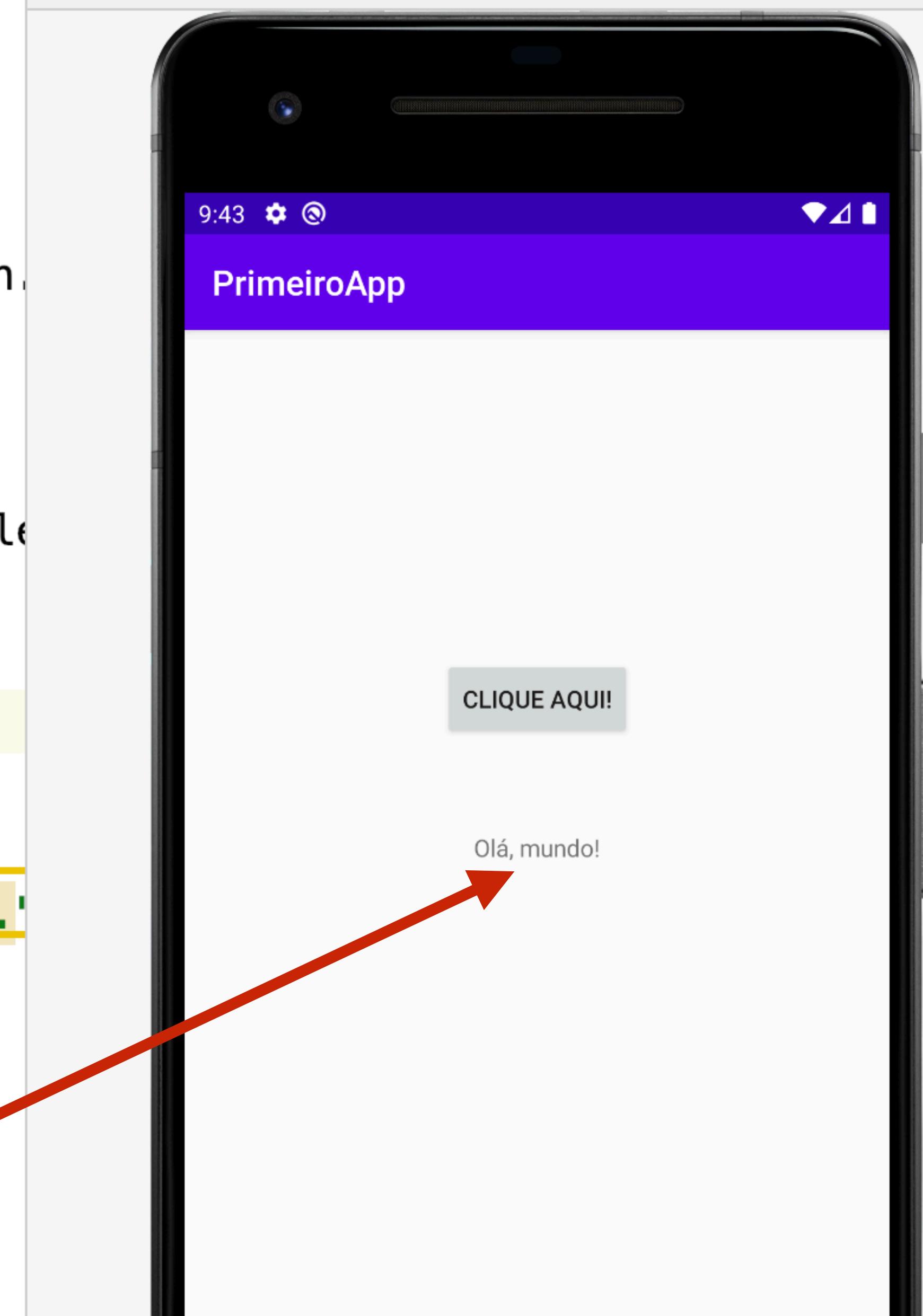
MainActivity.kt x activity\_main.xml x

```
1 package com.example.botao
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.View
6 import kotlinx.android.synthetic.main.activity_main.*
7
8 class MainActivity : AppCompatActivity() {
9
10    override fun onCreate(savedInstanceState: Bundle?) {
11        super.onCreate(savedInstanceState)
12        setContentView(R.layout.activity_main)
13    }
14
15    fun fui_clicado(view: View) {
16        texto.setText("Eu fui clicado, ora bolas...")
17    }
18
19 }
```

o método setText() define o texto do componente TextView

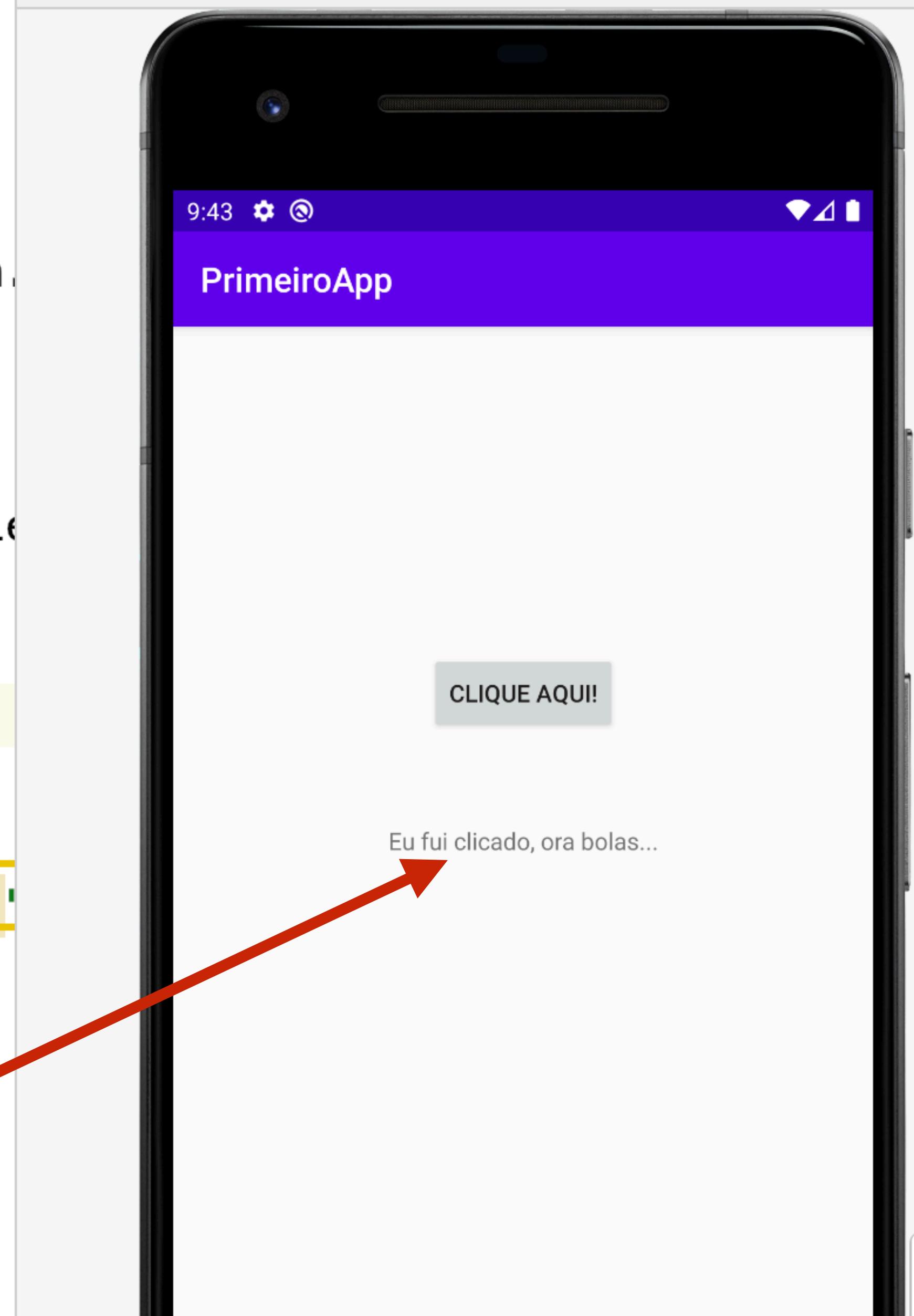
```
1 package com.example.botao
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.View
6 import kotlinx.android.synthetic.main.activity_main.*
7
8 class MainActivity : AppCompatActivity() {
9
10    override fun onCreate(savedInstanceState: Bundle) {
11        super.onCreate(savedInstanceState)
12        setContentView(R.layout.activity_main)
13    }
14
15    fun fui_clicado(view: View) {
16        texto.setText("Eu fui clicado, ora bolas...")
17    }
18
19}
```

Texto original



```
1 package com.example.botao
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.View
6 import kotlinx.android.synthetic.main.activity_main.*
7
8 class MainActivity : AppCompatActivity() {
9
10    override fun onCreate(savedInstanceState: Bundle) {
11        super.onCreate(savedInstanceState)
12        setContentView(R.layout.activity_main)
13    }
14
15    fun fui_clicado(view: View) {
16        texto.setText("Eu fui clicado, ora bolas...")
17    }
18
19}
```

Texto após o clique no botão



# Podemos usar o log...

The screenshot shows the Android Studio interface with the following components:

- Project Structure:** On the left, the project tree shows the `app` module with `AndroidManifest.xml`, `activity_main.xml`, and several Java files under `com.example.botao`.
- MainActivity.kt:** The code editor displays the `MainActivity` class:

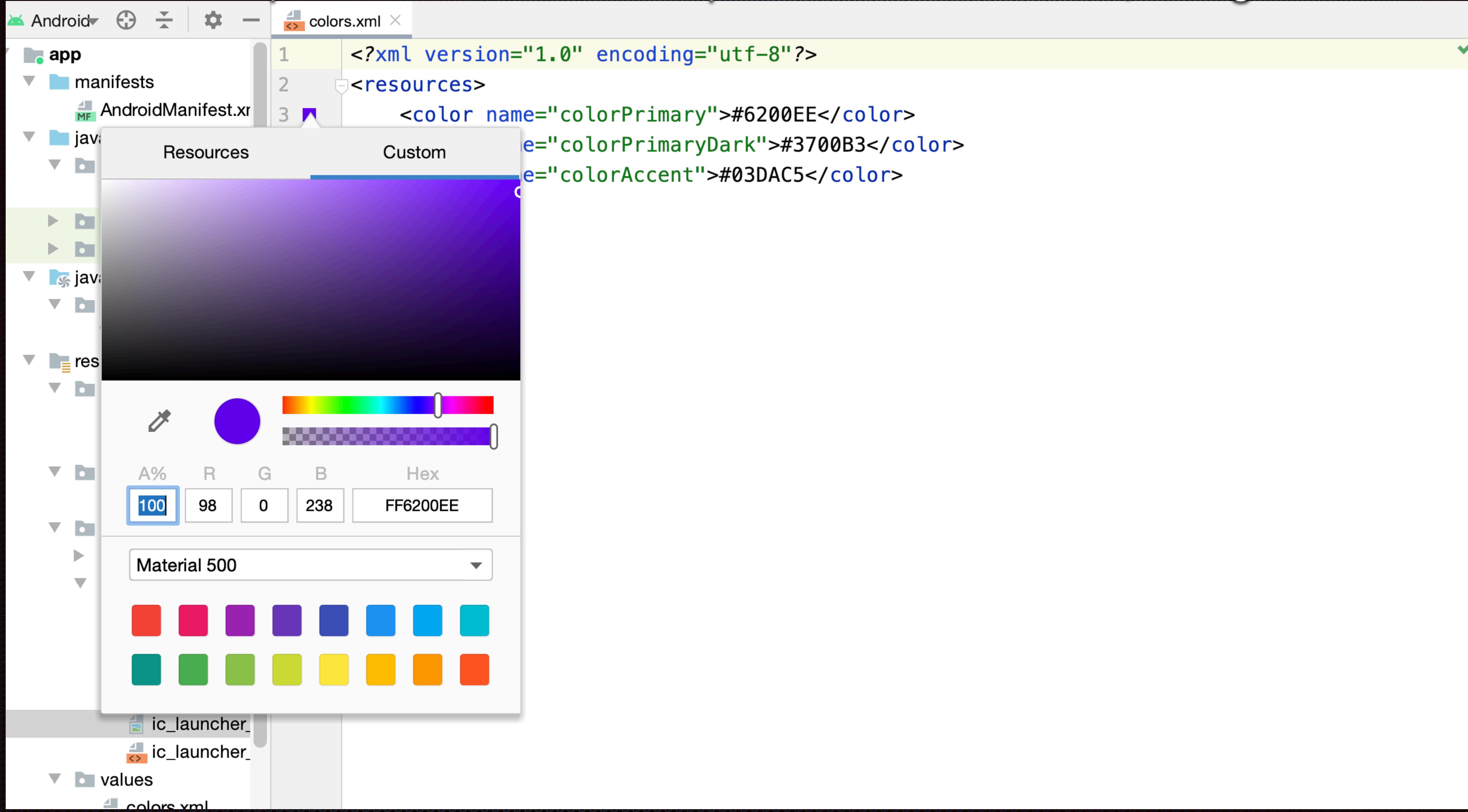
```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    fun fui_clicado(view: View) {
        Log.i(tag: "info", msg: "Botão clicado")
        texto.setText("Eu fui clicado, ora bolas...")
    }
}
```

A red arrow points from the `Log.i` call in the code to the corresponding log entry in the Logcat panel.
- Emulator:** A Pixel 2 API 29 device is running the app, showing a purple header with "PrimeiroApp" and a button labeled "CLIQUE AQUI!". Below it, a text view says "Eu fui clicado, ora bolas...".
- Logcat:** The bottom panel shows log entries:

```
2021-03-30 09:50:18.991 8711-8711/com.example.primeiroapp I/Choreographer: Skipped 33 frames! The application may
2021-03-30 09:50:19.011 8711-8741/com.example.primeiroapp D/EGL_emulation: eglMakeCurrent: 0xe1c1a120: ver 3 0 (tir
2021-03-30 09:50:22.874 8711-8711/com.example.primeiroapp I/info: Botão clicado
```

# Existem arquivos xml de recursos para definirmos cores, strings e estilos



# Existem arquivos xml de recursos para definirmos cores, strings e estilos

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right.

**Project Structure:**

- java**:
  - com.example.bota
    - MainActivity
  - com.example.bota
  - com.example.bota
- java (generated)**:
  - com.example.bota
    - BuildConfig
- res**:
  - drawable**: ic\_launcher\_background, ic\_launcher\_foreground
  - layout**: activity\_main.xml
  - mipmap**: ic\_launcher (6), ic\_launcher\_round
    - ic\_launcher\_1, ic\_launcher\_2, ic\_launcher\_3, ic\_launcher\_4, ic\_launcher\_5, ic\_launcher\_6
  - values**: colors.xml, strings.xml, styles.xml

**Code Editor (strings.xml):**

```
<resources>
    <string name="app_name">Botao</string>
</resources>
```

The code editor shows the XML resource file with a single string entry. The string value "Botao" is highlighted with a yellow background and a blue border, indicating it is selected or being edited. The editor interface includes tabs for "strings.xml", "Open editor", and "Hide notification".

# Para acessar estes recursos, devemos usar o @



The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `AndroidManifest.xml` file. A red arrow points from the text "Atributo app\_name definido no arquivo strings.xml" at the bottom to the `android:label="@string/app_name"` line in the XML code.

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        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>
```

Atributo app\_name definido no arquivo strings.xml

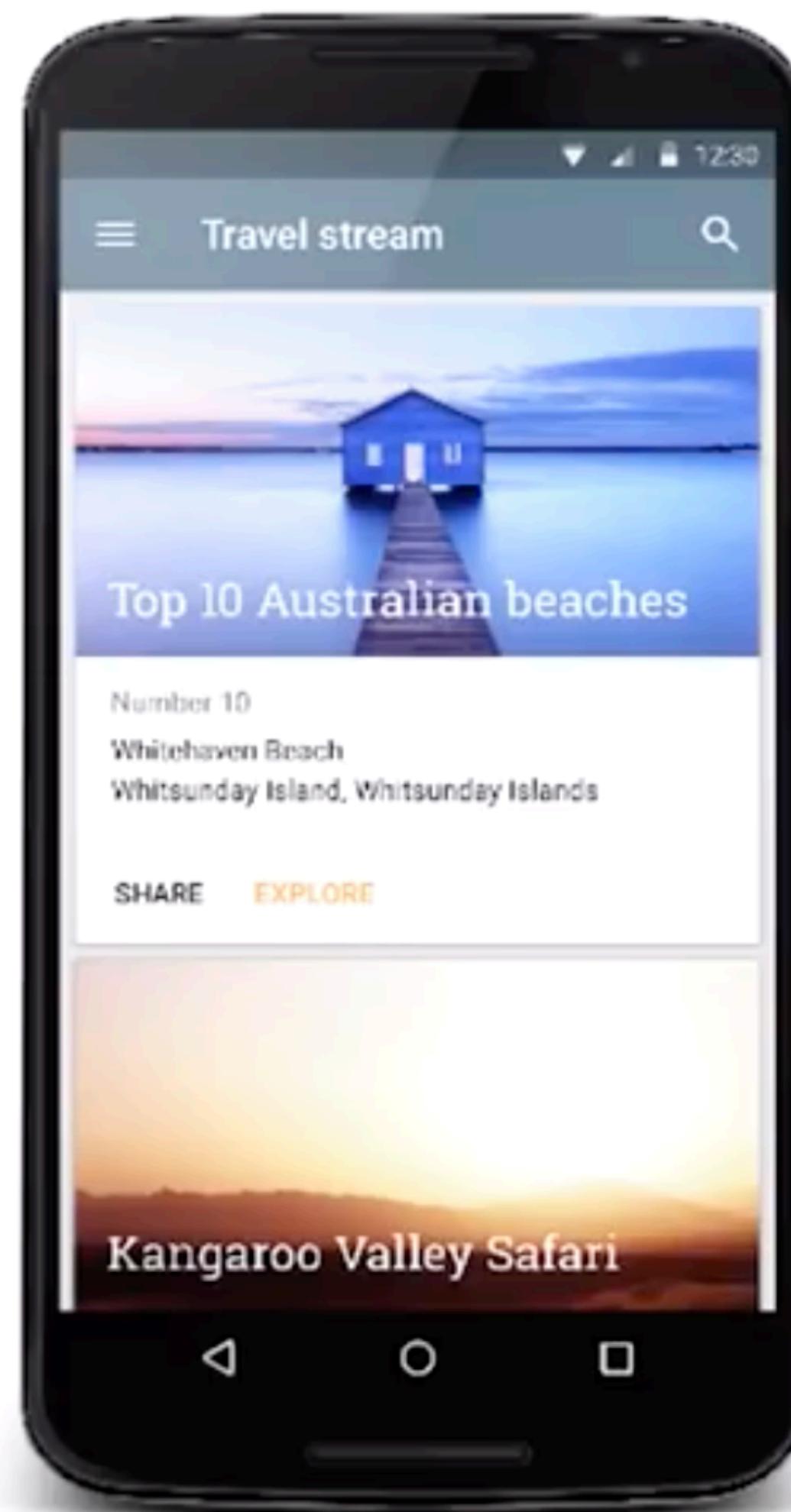
A classe BuildConfig tem as constantes com informações de construção do App e não devem ser alteradas diretamente.

The screenshot shows the Android Studio interface with the BuildConfig.java file open in the editor. The code defines a public final class BuildConfig with static final fields for DEBUG, APPLICATION\_ID, BUILD\_TYPE, FLAVOR, VERSION\_CODE, and VERSION\_NAME. The DEBUG field is set to true. The APPLICATION\_ID is com.example.botao. The BUILD\_TYPE is debug. The FLAVOR is empty. The VERSION\_CODE is 1. The VERSION\_NAME is 1.0. A tooltip above the code states: "Generated source files should not be modified". The file path is ~/Desktop/Android/Aula 01 - Android Studio/1.PrimeiroApp/app/src/main/res/values/colors.xml. The Java code is as follows:

```
1  /...
4  package com.example.botao;
5
6  public final class BuildConfig {
7      public static final boolean DEBUG = Boolean.parseBoolean("true");
8      public static final String APPLICATION_ID = "com.example.botao";
9      public static final String BUILD_TYPE = "debug";
10     public static final String FLAVOR = "";
11     public static final int VERSION_CODE = 1;
12     public static final String VERSION_NAME = "1.0";
13 }
14
```

# TextView

# VIEWS



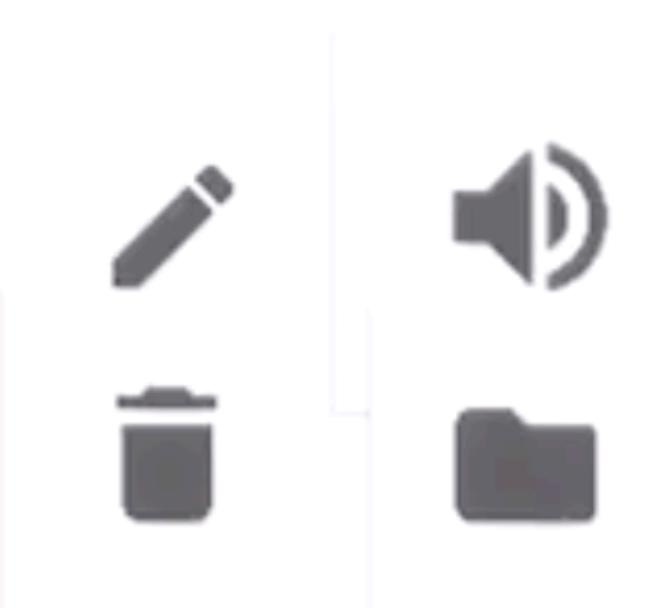
Text View

Morning Pizza

I'll be in your neighborhood doing errands  
this weekend if you want to meet up for  
brunch. Lemme know.

Show notifications

ImageView



Button



BUTTON

BUTTON



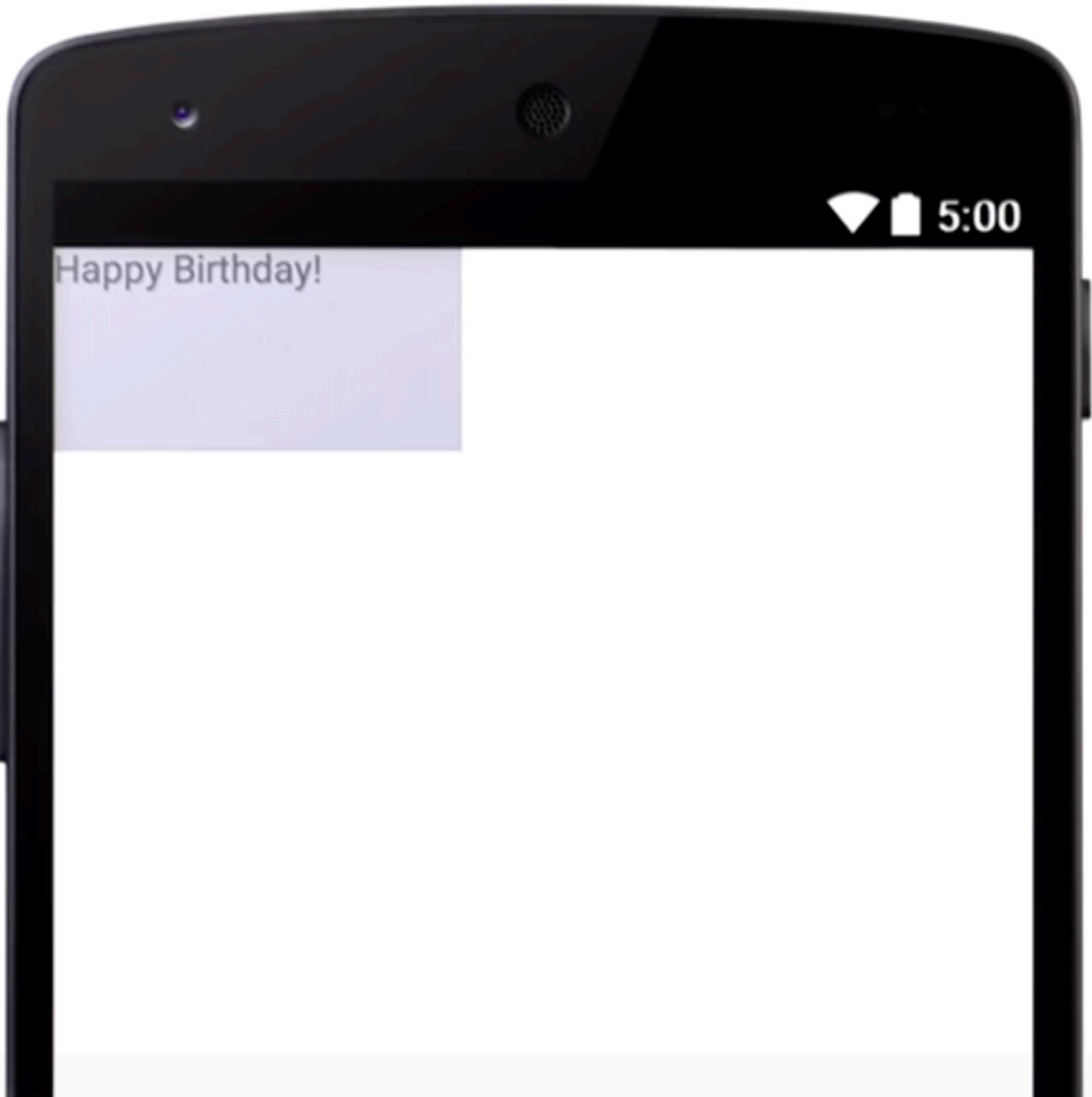
# VIEWS

Identificando os componentes:

A	<input type="radio"/> TextView	<input checked="" type="radio"/> ImageView	<input type="radio"/> Button
B	<input checked="" type="radio"/> TextView	<input type="radio"/> ImageView	<input type="radio"/> Button
C	<input checked="" type="radio"/> TextView	<input type="radio"/> ImageView	<input type="radio"/> Button
D	<input checked="" type="radio"/> TextView	<input type="radio"/> ImageView	<input type="radio"/> Button
E	<input type="radio"/> TextView	<input type="radio"/> ImageView	<input checked="" type="radio"/> Button

# TEXTVIEW

```
<TextView  
    android:text="Happy Birthday!"  
    android:background="@android:color/darker_gray"  
    android:layout_width="150dp"  
    android:layout_height="75dp" />
```



# XML SYNTAX

```
<TextView  
    android:text="Happy Birthday!"  
    android:background="@android:color/darker_gray"  
    android:layout_width="150dp"  
    android:layout_height="75dp"/>
```

Cada componente pode definir uma série de atributos (características).

## Attributes:

- width
- height
- background
- text

## Attributes:

- black hair
- brown eyes



# XML SYNTAX

```
<TextView  
    android:text="Happy Birthday!"  
    android:background="@android:color/darker_gray"  
    android:layout_width="150dp"  
    android:layout_height="75dp" />
```

↑  
Attribute  
Name

↑  
Attribute  
Value

Cada atributo deve ter o par: Nome e Valor

# XML SYNTAX

```
<TextView  
    android:text="Happy Birthday!"  
    android:background="@android:color/darker_gray"  
    android:layout_width="150dp"  
    android:layout_height="75dp" />
```



Attribute  
Name



Attribute  
Value in  
"Quotations"

O valor do atributo deve estar sempre entre aspas duplas.

## ML Layout Code

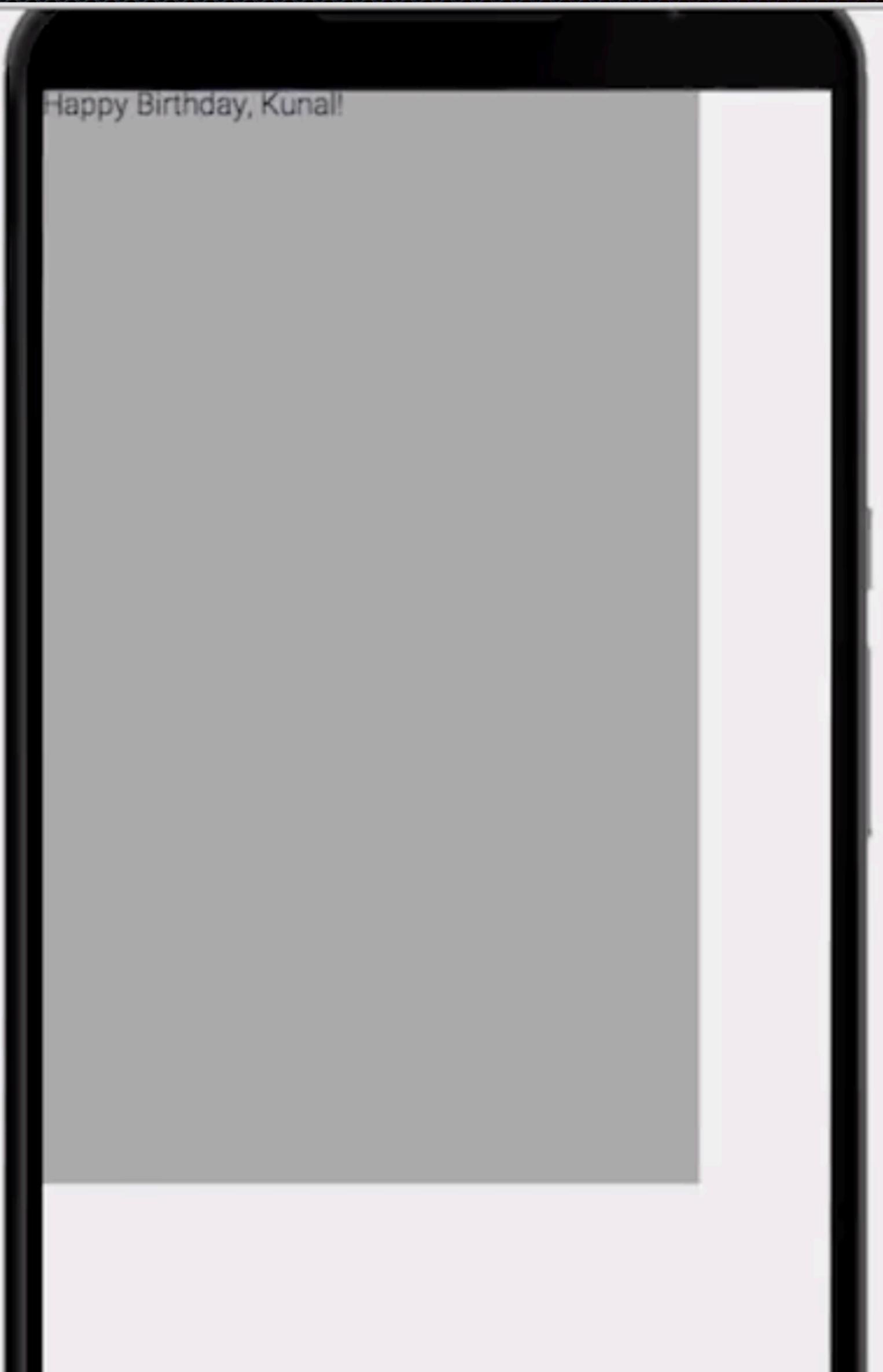
```
1 <TextView  
2     android:text="Happy Birthday!"  
3     android:background="@android:color/darker_gray"  
4     android:layout_width="150dp"  
5     android:layout_height="75dp" />
```



Visualizando no emulador...

## XML Layout Code

```
1 <TextView  
2     android:text="Happy Birthday, Kunal!"  
3     android:background="@android:color/darker_gray"  
4     android:layout_width="300dp"  
5     android:layout_height="500dp" />
```



Aumentando a largura e o comprimento

## XML Layout Code

UNDO    REDO

*Code has been saved*

RESET CODE

```
1 <TextView  
2     android:text="Excited for the gift you'll surprise me with."  
3     android:background="@android:color/darker_gray"  
4     android:layout_width="wrap_content"  
5     android:layout_height="wrap_content"  
6     android:textSize="45sp" />
```

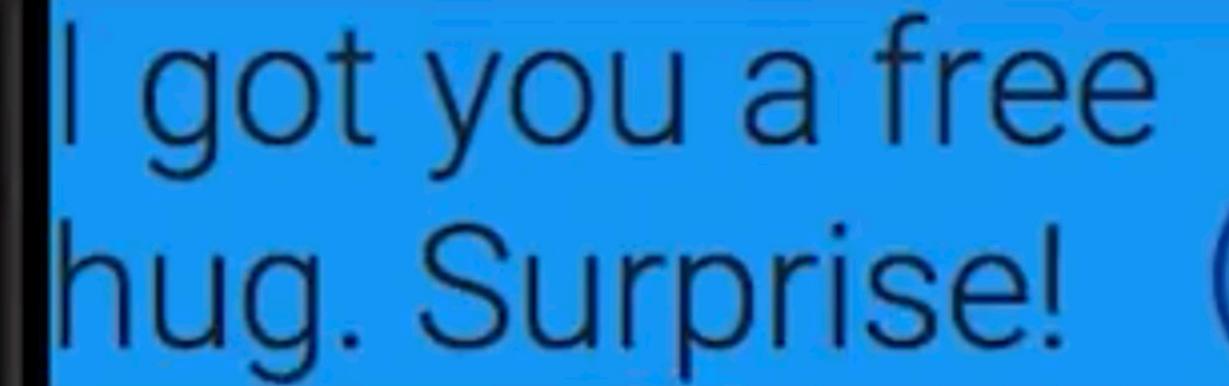


Modificando o texto e deixando o tamanho do TextView se ajustar conforme o tamanho do texto (wrap)

## XML Layout Code

```
1 <TextView  
2     android:text="I got you a free hug. Surprise!"  
3     android:background="#2196F3"  
4     android:layout_width="wrap_content"  
5     android:layout_height="wrap_content"  
6     android:textSize="45sp" />
```

□ <http://www.color-hex.com/color-wheel/>

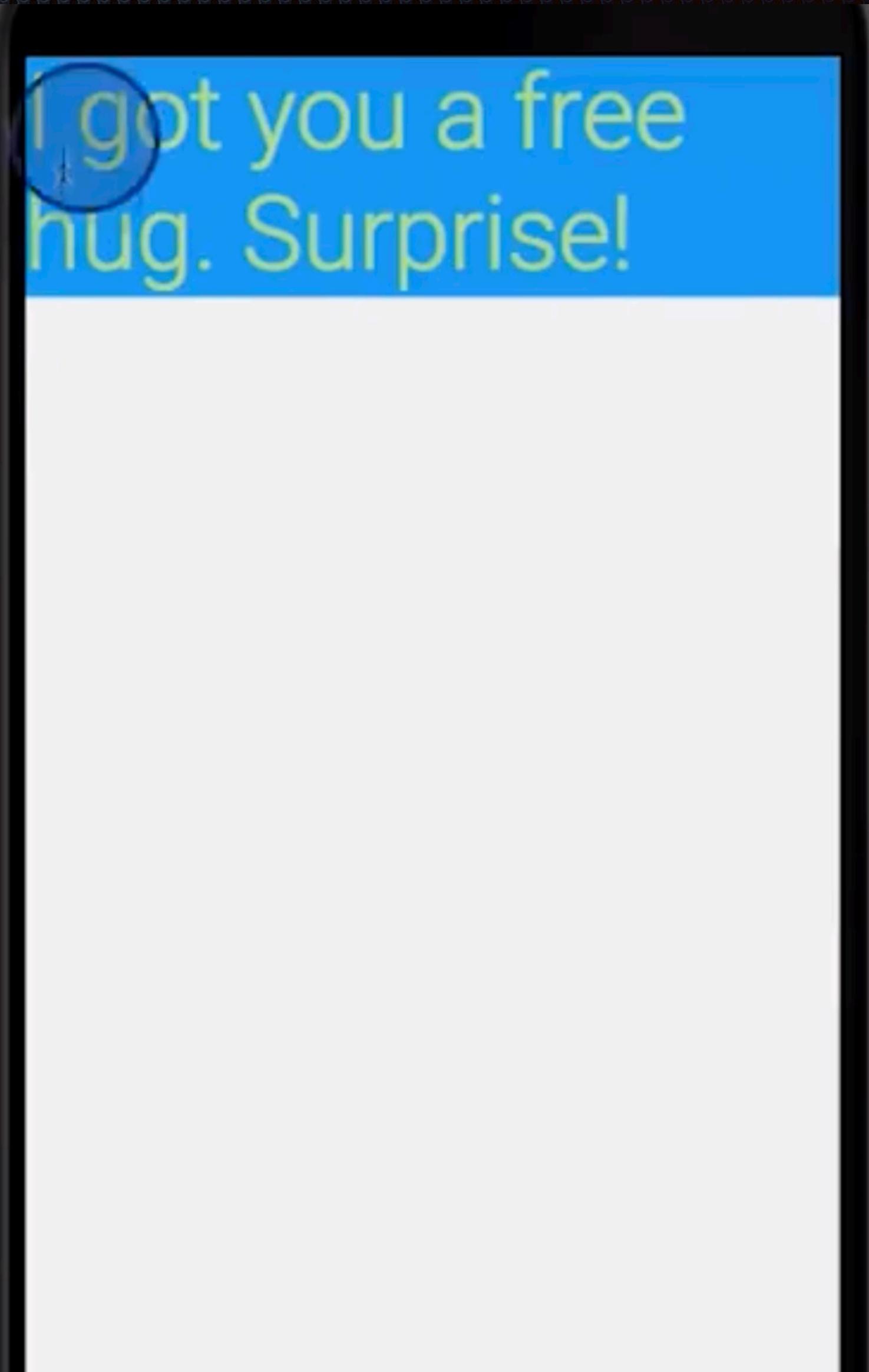


I got you a free  
hug. Surprise!

Modificando a cor de fundo (background)

## XML Layout Code

```
1 <TextView  
2     android:text="I got you a free hug. Surprise!"  
3     android:background="#2196F3"  
4     android:textColor="#AED581"  
5     android:layout_width="wrap_content"  
6     android:layout_height="wrap_content"  
7     android:textSize="45sp" />
```

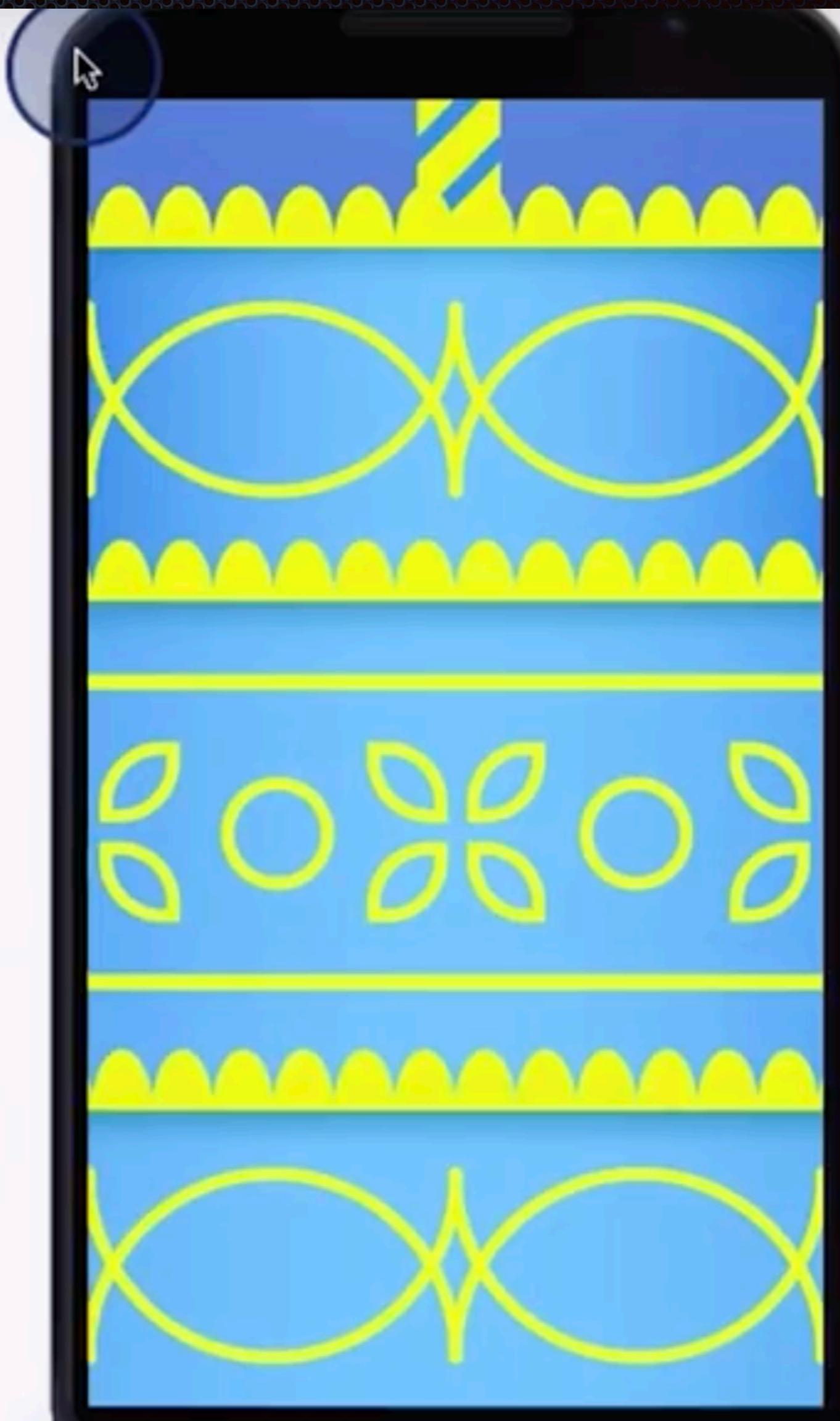


Mudando a cor do texto (textColor).

# ImageView

## XML Layout Code

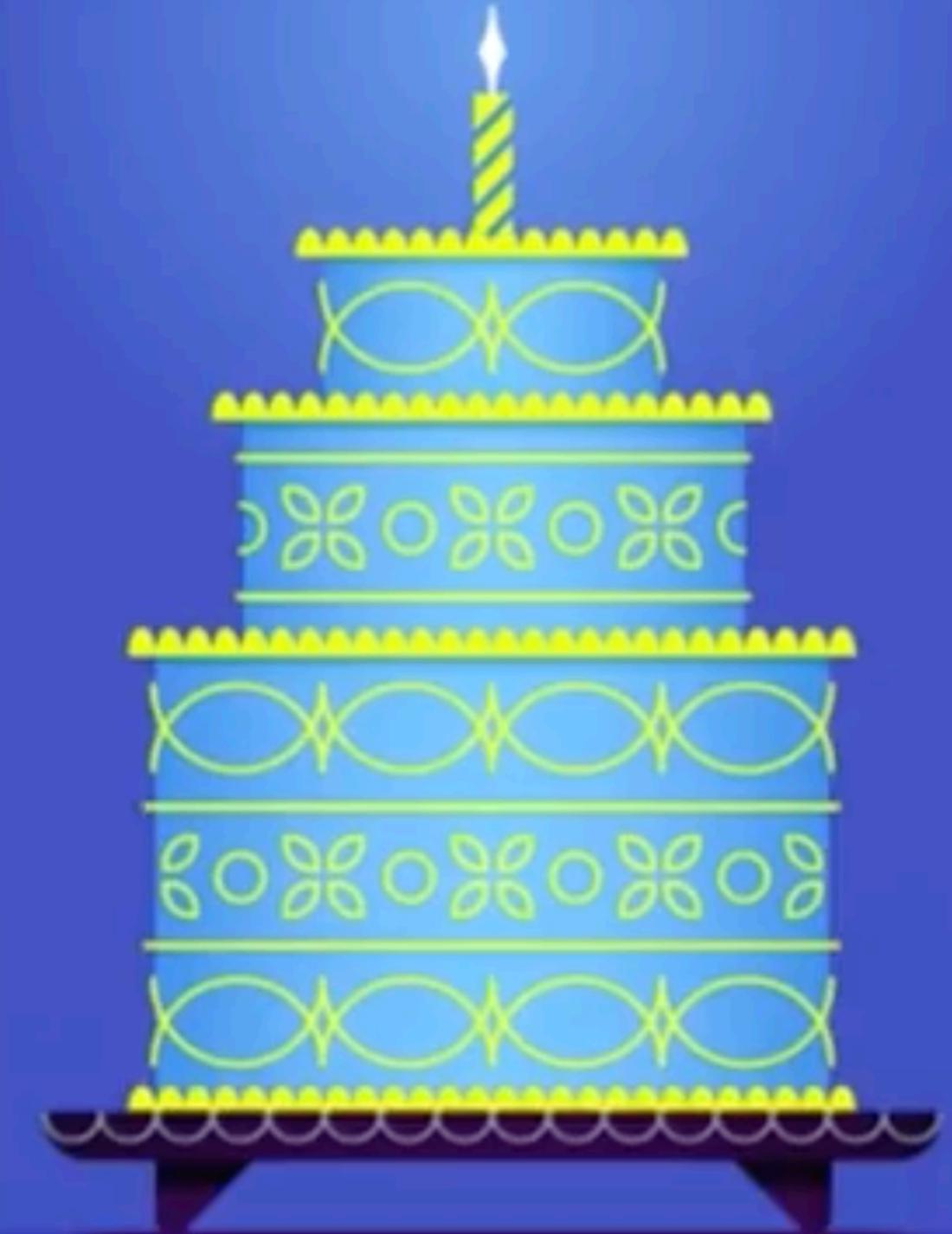
```
1 <ImageView  
2     android:src="@drawable/cake"  
3     android:layout_width="wrap_content"  
4     android:layout_height="wrap_content"  
5     android:scaleType="center" />
```



Colocando uma imagem em um ImageView.

## XML Layout Code

```
1 <ImageView  
2     android:src="@drawable/cake"  
3     android:layout_width="wrap_content"  
4     android:layout_height="wrap_content"  
5     android:scaleType="centerCrop" />
```



Centralizando a imagem.

# Diferentes Tipos de Views

## TextView

Displays text

```
<TextView  
    android:id="@+id/title_text_view"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/my_photos"  
    android:textAppearance="?android:textAppearanceLarge"  
    android:textColor="#4689C8"  
    android:textStyle="bold" />
```

## My Photos

## ImageView

Displays Image

```
<ImageView  
    android:id="@+id/photo_image_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:scaleType="centerCrop"  
    android:src="@drawable/beach" />
```



## Button

Button with text label

```
<Button  
    android:id="@+id/next_button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/next" />
```

NEXT

## EditText

Text field that you can type into

## Spinner

Click on it to show a list  
of dropdown options

```
<EditText  
    android:id="@+id/album_description_view"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="@string/album_description"  
    android:inputType="textMultiLine" />
```

## Album Description

```
<Spinner  
    android:id="@+id/sort_by_spinner"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

Create SpinnerAdapter in Java code to populate the  
options. [See more](#)

Beach ▾

Beach

BBQ

Family dinner

Party

# CheckBox

Checkbox with text label

# RadioButton

Radio button (where you can select one out of a group of radio buttons)

## <CheckBox

```
    android:id="@+id/notify_me_checkbox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/notify_me"
    android:textAppearance="?android:textAppearanceMedium" />
```

Notify me

## <RadioGroup

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <RadioButton
        android:id="@+id/yes_radio_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/yes"
        android:textAppearance="?android:textAppearanceMedium" />
```

Yes

```
    <RadioButton
        android:id="@+id/no_radio_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/no"
        android:textAppearance="?android:textAppearanceMedium" />
```

No

```
</RadioGroup>
```

# RatingBar

Star rating

<RatingBar

```
    android:id="@+id/rating_bar"
    style="?android:attr/ratingBarStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numStars="5"
    android:rating="2.5"
    android:stepSize="0.5" />
```



# Switch

On / off switch that you can drag right or left (or just tap to toggle the state)

<Switch

```
    android:id="@+id/backup_photos_switch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/auto_backup_photos"
    android:textAppearance="?android:textAppearanceSmall" />
```

Auto backup photos 

# SeekBar

Displays progress and allows you to drag the handle anywhere in the bar (i.e. for music or video player)

<SeekBar

```
    android:id="@+id/seek_bar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:max="100"
    android:progress="20" />
```



# SearchView

A search field that you can type a query into

## <SearchView

```
    android:id="@+id/search_viewr"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:iconifiedByDefault="false"
    android:queryHint="@string/search_photos" />
```

Search Photos



# ProgressBar

Loading spinner

## <ProgressBar

```
    android:id="@+id/loading_spinner"
    style="@style/Widget.AppCompat.ProgressBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

# ProgressBar

Horizontal loading indicator

## <ProgressBar

```
    android:id="@+id/progress_bar"
    style="@style/Widget.AppCompat.ProgressBar.Horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:indeterminate="false"
    android:max="100"
    android:progress="40" />
```



# Links

- <https://labs.udacity.com/android-visualizer/#/android/>
- <https://labs.udacity.com/android-visualizer/#/android/equal-space-children>
- <https://labs.udacity.com/android-visualizer/#/android/match-parent>