



# Componentes Android Studio

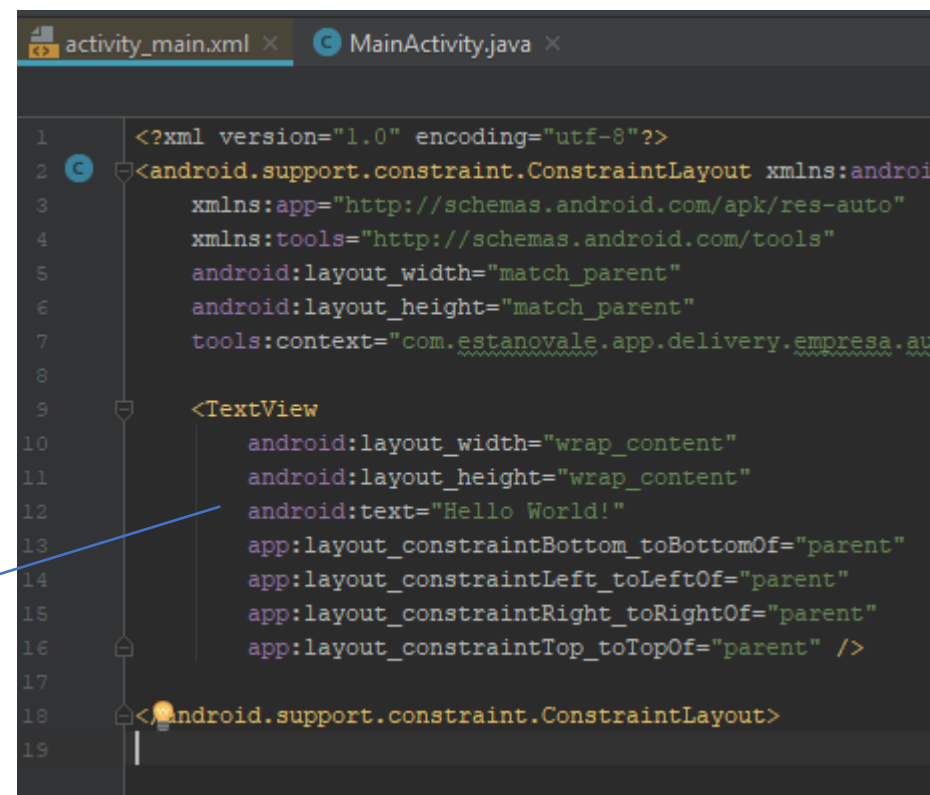
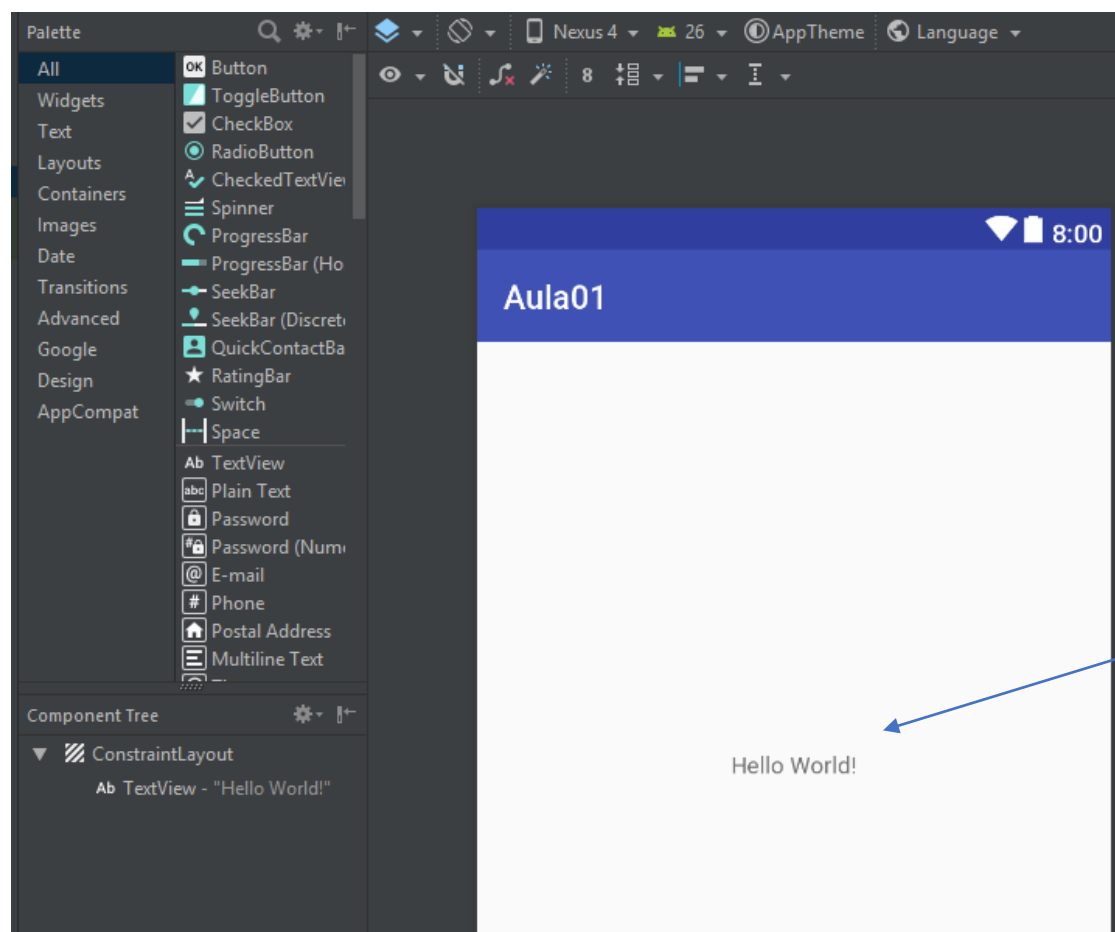
Prof. Me. Hélio  
Esperidião.

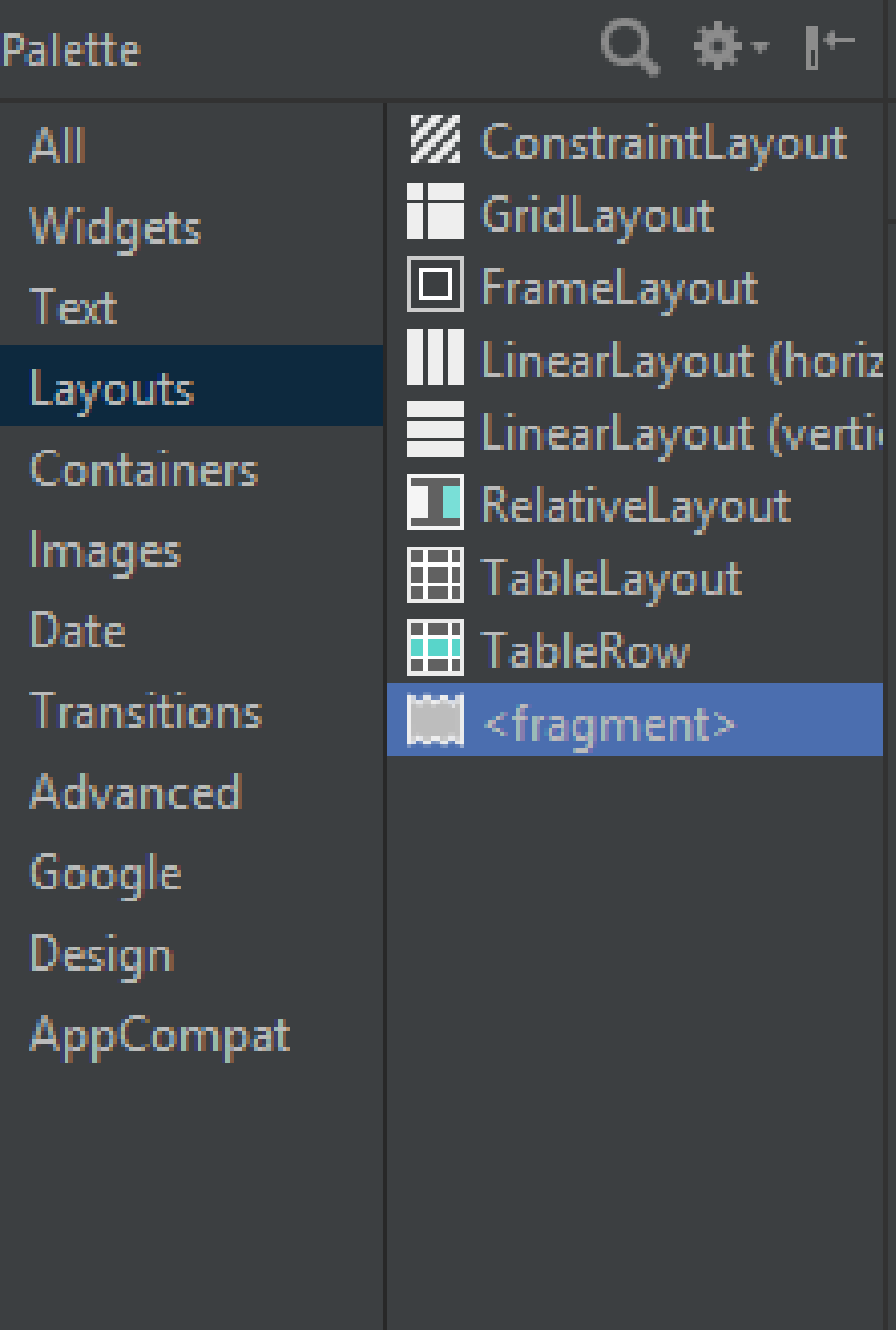
# Arquivo xml

```
activity_main.xml x MainActivity.java x
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.estanovale.app.delivery.empresaa.aula01.MainActivity">
8
9     <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="Hello World!"
13         app:layout_constraintBottom_toBottomOf="parent"
14         app:layout_constraintLeft_toLeftOf="parent"
15         app:layout_constraintRight_toRightOf="parent"
16         app:layout_constraintTop_toTopOf="parent" />
17
18 </android.support.constraint.ConstraintLayout>
19
```

- O arquivo xml do activity\_main é responsável pelo layout do aplicativo, nele são definidas as posições de todos os componentes visuais.

# Editor Gráfico





# Paleta de componentes visuais

- Os layouts determinam como os componentes visuais serão distribuídos na janela.
- O Layout `LinearLayout` é o mais simples pois permite que os componentes sejam posicionados um abaixo do outro ou lado a lado.

# Estrutura básica do xml

```
<?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.estanovale.app.delivery.empresaa.aula01.MainActivity">
    |
</android.support.constraint.ConstraintLayout>
```

Programe aqui seus componentes visuais.

# Programme o LinearLayout

```
<?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.estanovale.app.delivery_empresa.aula01.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
    >

    </LinearLayout>

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

Seu tamanho é definido de acordo  
Com o tamanho da tela

Sua organização é na vertical, ou seja,  
Os componentes ficarão um abaixo  
Do outro.

# XML e .kt

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
>

    <TextView
        android:id="@+id/lblOlamundo"
        android:text="Ola mundo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
    />

</LinearLayout>
```

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

Todos os arquivos xml a princípio devem ser associados a um arquivo kt

# Label, caixa de texto e botão

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  >

  <TextView
    android:id="@+id/lblOlamundo"
    android:text="Ola mundo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
  />

  <EditText
    android:id="@+id/txtNome"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPersonName" />

  <Button
    android:id="@+id/btnConfirmar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Confirmar" />

</LinearLayout>
```

Id do componente

Texto do componente

Largura e altura



# Toast

---

Funciona como uma Caixa de mensagem .

Desaparece depois de alguns instantes.

```
Toast.makeText(this@MainActivity, "Ola Mundo",  
Toast.LENGTH_SHORT).show()
```

# Button

```
<Button
```

```
    android:id="@+id/button"
```

```
    android:layout_width="match_parent"
```

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

```
    android:text="Button" />
```



BUTTON



# Instância do botão e evento click

---

```
val btn_click_me = findViewById(R.id.button) as Button

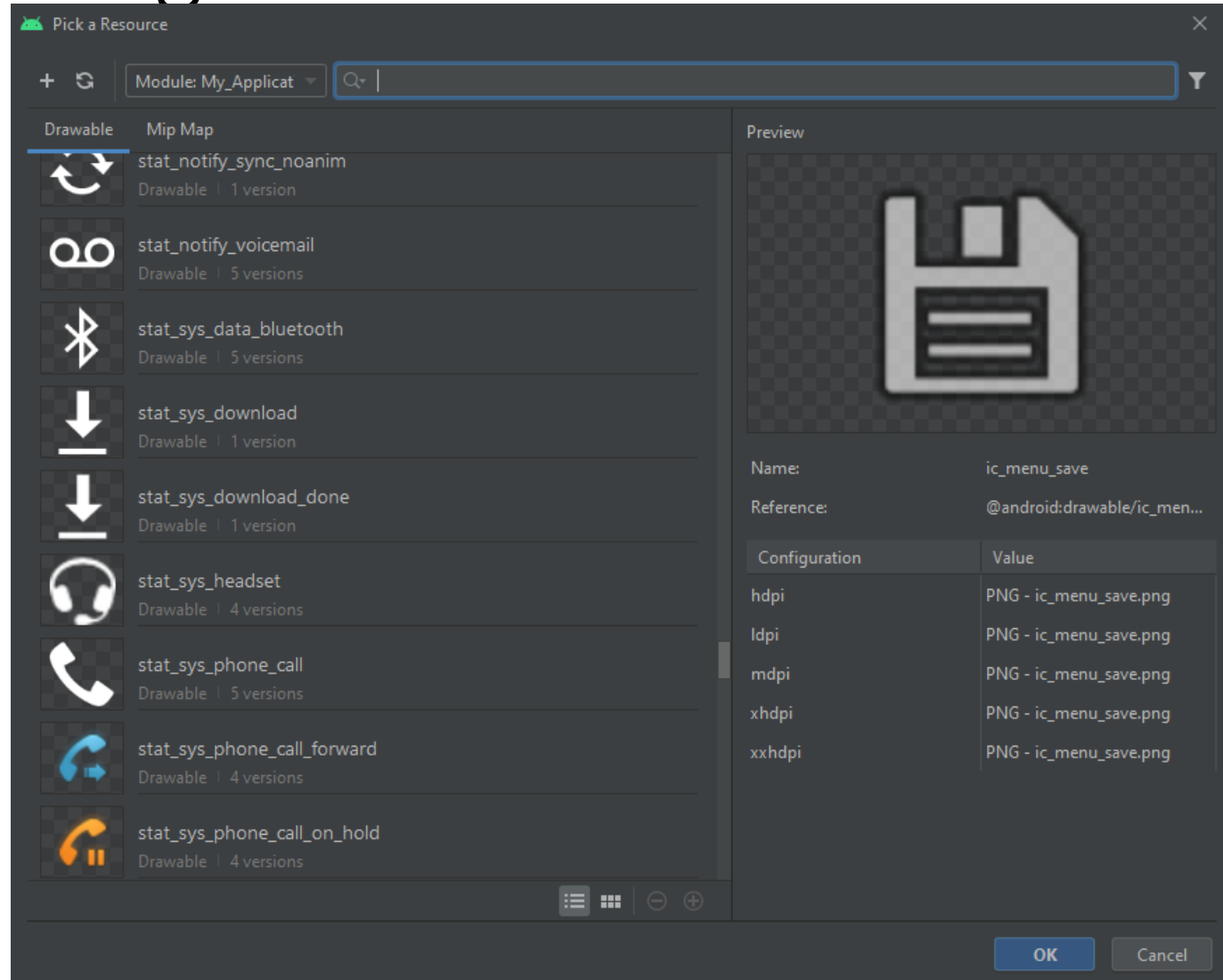
btn_click_me.setOnClickListener {
    Toast.makeText(this@MainActivity, "Ola Mundo",
        Toast.LENGTH_SHORT).show()
}
```

# ImageButton

```
<ImageButton  
    android:id="@+id/imageButton"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:srcCompat="@android:drawable/ic_menu_save" />
```



# Escolha a imagem



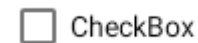
# ImageButton

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val btn_click_me = findViewById(R.id.imageButton) as ImageButton  
        btn_click_me.setOnClickListener {  
            Toast.makeText(this@MainActivity, "Ola Mundo", Toast.LENGTH_SHORT).show()  
        }  
    }  
}
```

# CheckBox

```
<CheckBox
```

```
    android:id="@+id/checkBox"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="CheckBox" />
```



# checkBox (CheckedChangeListener )

```
var checkBox: CheckBox = findViewById(R.id.checkBox)
checkBox.setOnCheckedChangeListener { buttonView, isChecked ->
    if (isChecked) {
        Toast.makeText(this@MainActivity, "Marcado", Toast.LENGTH_SHORT).show()
    }
    else{
        Toast.makeText(this@MainActivity, "desmarcado", Toast.LENGTH_SHORT).show()
    }
}
```



# checkBox (OnClickListener )

```
var checkBox: CheckBox = findViewById(R.id.checkBox)
checkBox.setOnClickListener {
    var temp: String = checkBox.isChecked.toString()
    Toast.makeText(this@MainActivity, temp, Toast.LENGTH_SHORT).show()
}
```

# RadioButton

```
<RadioButton
```

```
    android:id="@+id/radioButton"
```

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

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

```
    android:text="RadioButton" />
```



RadioButton

# RadioGroup

```
<RadioGroup
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <RadioButton
        android:id="@+id/radioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="RadioButton" />
    <RadioButton
        android:id="@+id/radioButton2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="RadioButton" />
</RadioGroup>
```

RadioButton

RadioButton

# RadioButton - Programando

```
var radioButton: RadioButton = findViewById(R.id.radioButton)
val btn_click_me = findViewById(R.id.imageButton) as ImageButton
btn_click_me.setOnClickListener {
    var temp: String = radioButton.isChecked.toString()
    Toast.makeText(this@MainActivity, temp, Toast.LENGTH_SHORT).show()
}
```

# ToggleButton

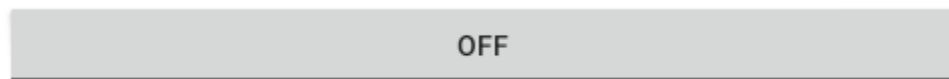
```
<ToggleButton
```

```
    android:id="@+id/toggleButton"
```

```
    android:layout_width="match_parent"
```

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

```
    android:text="ToggleButton" />
```



# ToggleButton – Recuperando status

```
var toggleButton: ToggleButton = findViewById(R.id.toggleButton)
val btn_click_me = findViewById(R.id.button) as Button

btn_click_me.setOnClickListener {
    var temp: String =toggleButton.isChecked.toString()
    Toast.makeText(this@MainActivity, temp, Toast.LENGTH_SHORT).show()
}
```

# Switch

```
<Switch
```

```
    android:id="@+id/switch1"
```

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

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

```
    android:text="Switch" />
```



# Switch

```
var switch: Switch = findViewById(R.id.switch1)
val btn_click_me = findViewById(R.id.button) as Button

btn_click_me.setOnClickListener {
    var temp: String = switch.isChecked.toString()
    Toast.makeText(this@MainActivity, temp, Toast.LENGTH_SHORT).show()
}
```



# Switch: Evento

```
var switch: Switch = findViewById(R.id.switch1)
val btn_click_me = findViewById(R.id.button) as Button

switch.setOnClickListener {
    var temp: String = switch.isChecked.toString()
    Toast.makeText(this@MainActivity, temp, Toast.LENGTH_SHORT).show()
}
```

# SeekBar

```
<SeekBar
```

```
    android:id="@+id/seekBar"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content" />
```



# SeekBar : Evento

```
var seekBar: SeekBar = findViewById(R.id.seekBar)
val btn_click_me = findViewById(R.id.button) as Button

seekBar.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener {

    override fun onProgressChanged(seekBar: SeekBar, i: Int, b: Boolean) {
        // Display the current progress of SeekBar
        Toast.makeText(this@MainActivity, i.toString(), Toast.LENGTH_SHORT).show()
    }

    override fun onStartTrackingTouch(p0: SeekBar?) {

    }

    override fun onStopTrackingTouch(p0: SeekBar?) {

    }

})
```

# SeekBar Discrete

```
<SeekBar
```

```
    android:id="@+id/seekBar2"
```

```
    style="@style/Widget.AppCompat.SeekBar.Discrete"
```

```
    android:layout_width="match_parent"
```

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

```
    android:max="10"
```

```
    android:progress="3" />
```



# SeekBar: Evento

```
var seekBar: SeekBar = findViewById(R.id.seekBar2)
val btn_click_me = findViewById(R.id.button) as Button

seekBar.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener {

    override fun onProgressChanged(seekBar: SeekBar, i: Int, b: Boolean) {
        // Display the current progress of SeekBar
        Toast.makeText(this@MainActivity, i.toString(), Toast.LENGTH_SHORT).show()
    }

    override fun onStartTrackingTouch(p0: SeekBar?) {

    }

    override fun onStopTrackingTouch(p0: SeekBar?) {

    }

})
```

# RatingBar

- `<RatingBar`
- `android:id="@+id/ratingBar"`
- `android:layout_width="wrap_content"`
- `android:layout_height="wrap_content"`
- `android:numStars="6"`



# RatingBar: Programando

```
val btn_click_me = findViewById(R.id.button) as Button
var ratingBar: RatingBar = findViewById(R.id.ratingBar)
btn_click_me.setOnClickListener {
    Toast.makeText(this@MainActivity, ratingBar.rating.toString(), Toast.LENGTH_SHORT).show()
}
```

# RatingBar: Evento

```
var ratingBar: RatingBar = findViewById(R.id.ratingBar)
ratingBar.setOnRatingBarChangeListener { ratingBar, fl, b ->
    Toast.makeText(this@MainActivity, ratingBar.rating.toString(), Toast.LENGTH_SHORT).show()
}
```



# ProgressBar

```
<ProgressBar  
    android:id="@+id/progressBar"  
    style="?android:attr/progressBarStyleHorizontal"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

# ProgressBar: Progredindo

```
val btn_click_me = findViewById(R.id.button) as Button
var progressBar: ProgressBar = findViewById(R.id.progressBar)
var progress:Int = 0;
btn_click_me.setOnClickListener {
    progress++
    progressBar.progress=progress;
}
```

# Message Box

```
val btn_click_me = findViewById(R.id.button) as Button
btn_click_me.setOnClickListener {
    messageBox("Ola Mundo")
}
```

```
fun messageBox(textoMensagem: String): Unit {
    Toast.makeText(this@MainActivity, textoMensagem, Toast.LENGTH_SHORT).show()
}
```

# Componentes Dinâmicos

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

        val btn_click_me = findViewById(R.id.button) as Button
        btn_click_me.setOnClickListener {
            criarBotaoProgramacao()
        }
    }

    fun criarBotaoProgramacao() {
        // creating the button
        val linearLayout = findViewById<View>(R.id.linearLayout) as LinearLayout
        val botaoDinamico = Button(this)
        // setting layout_width and layout_height using layout parameters
        botaoDinamico.layoutParams = LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.MATCH_PARENT,
            LinearLayout.LayoutParams.WRAP_CONTENT
        )
        botaoDinamico.text = "Ola Mundo"
        botaoDinamico.setBackgroundColor(Color.GREEN)
        // add Button to LinearLayout
        linearLayout.addView(botaoDinamico)
    }
}
```