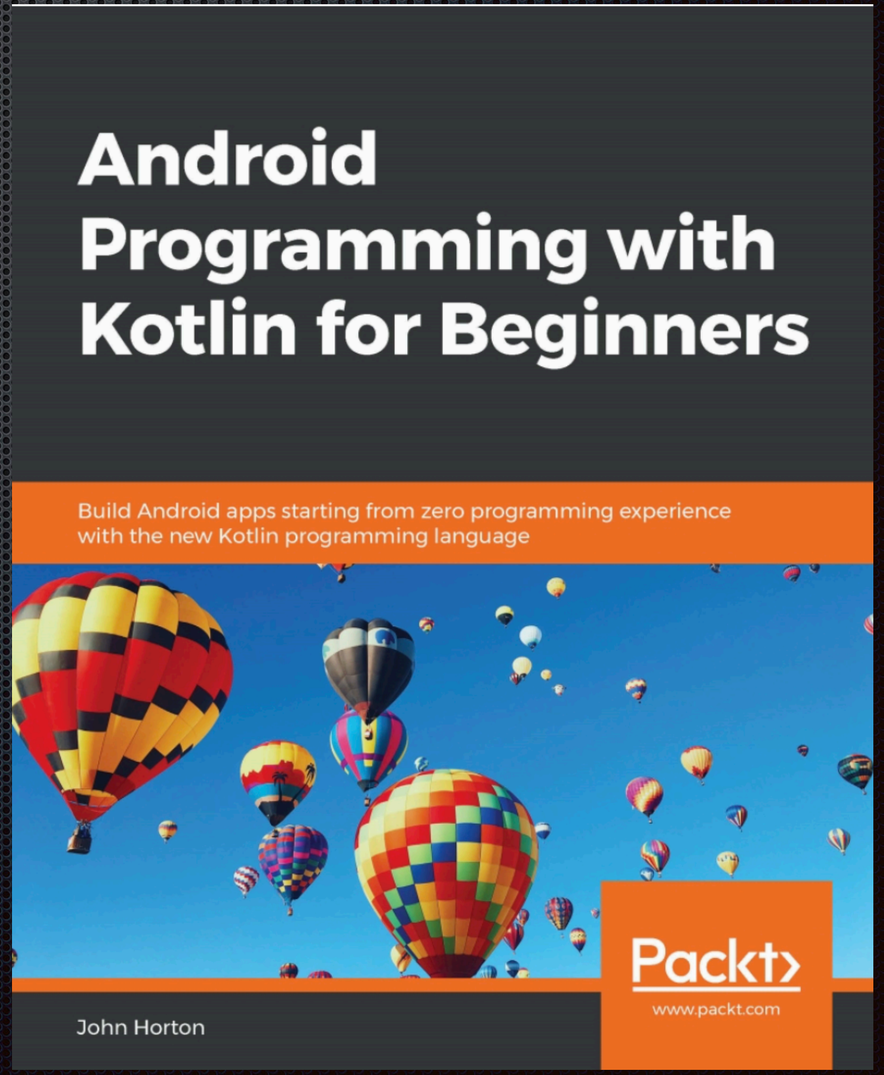
https://github.com/PacktPublishing/Android-Programming-with-Kotlin-for-Beginners

Android com Kotlin

Aula 04 - Layouts



Aula 04 - Constraint Layout e Table Layout

- Layout dentro de outro layout (LinearLayout)
- Interação entre Views (TableLayout e ConstraintLayout)
- CardView
- ScrollView

Layout dentro de outro Layout

Layouts

- São formas de agrupar componentes ou até mesmo outros layouts.
- Usamos o método setContentView() para definir qual estamos usando.

MainActivity.kt

```
1  package android.com.layouts
2
3  import androidx.appcompat.app.AppCompatActivity
4  import android.os.Bundle
5
6  class MainActivity : AppCompatActivity() {
7      override fun onCreate(savedInstanceState: Bundle?) {
8          super.onCreate(savedInstanceState)
9          setContentView(R.layout.activity_main) //aqui de
10     }
11 }
```

Linear Layout em Modo Vertical

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout 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      android:orientation="vertical"
8      tools:context=".MainActivity">
9      <!--- aqui precisamos colocar a orientação vertical -->
```

Adicionamos aqui um TextView

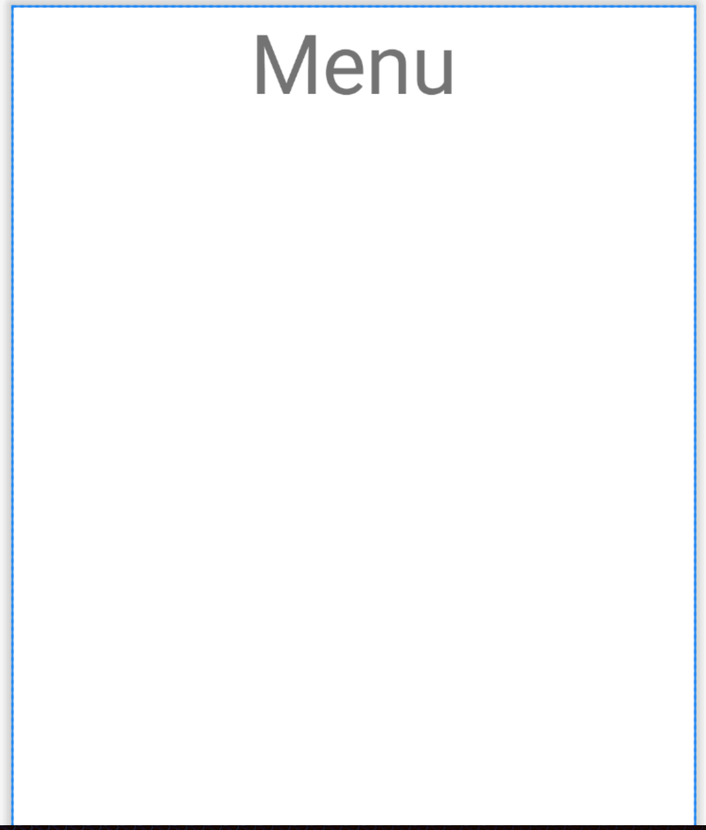
activity_main.xml

Code Split

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/rc
    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"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <!--- aqui precisamos colocar a orientação vertical -->

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="50sp"
        android:gravity="center_horizontal"
        android:text="Menu" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Palette

Pixel 29 Layouts 0dp



Adicionamos um Multi-line TextView

activity_main.xml

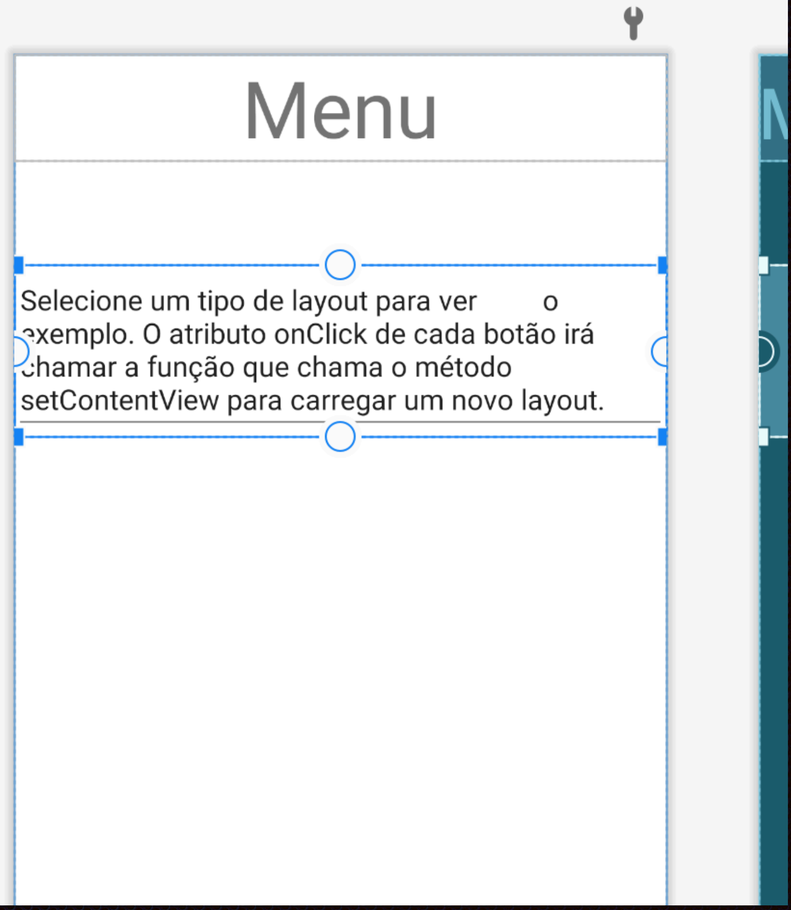
Code Split

```
11      <TextView
12          android:id="@+id/textView"
13          android:layout_width="match_parent"
14          android:layout_height="wrap_content"
15          android:textSize="50sp"
16          android:gravity="center_horizontal"
17          android:text="Menu" />
18
19      <EditText
20          android:id="@+id/editTextTextMultiLine"
21          android:layout_width="match_parent"
22          android:layout_height="wrap_content"
23          android:ems="10"
24          android:gravity="start|top"
25          android:inputType="textMultiLine"
26          android:text="Selecione um tipo de layout para ver
27          o exemplo. O atributo onClick de cada botão irá chamar
28          tools:layout_editor_absoluteX="20dp"
29          tools:layout_editor_absoluteY="133dp" />
30
31  </androidx.constraintlayout.widget.ConstraintLayout>
```

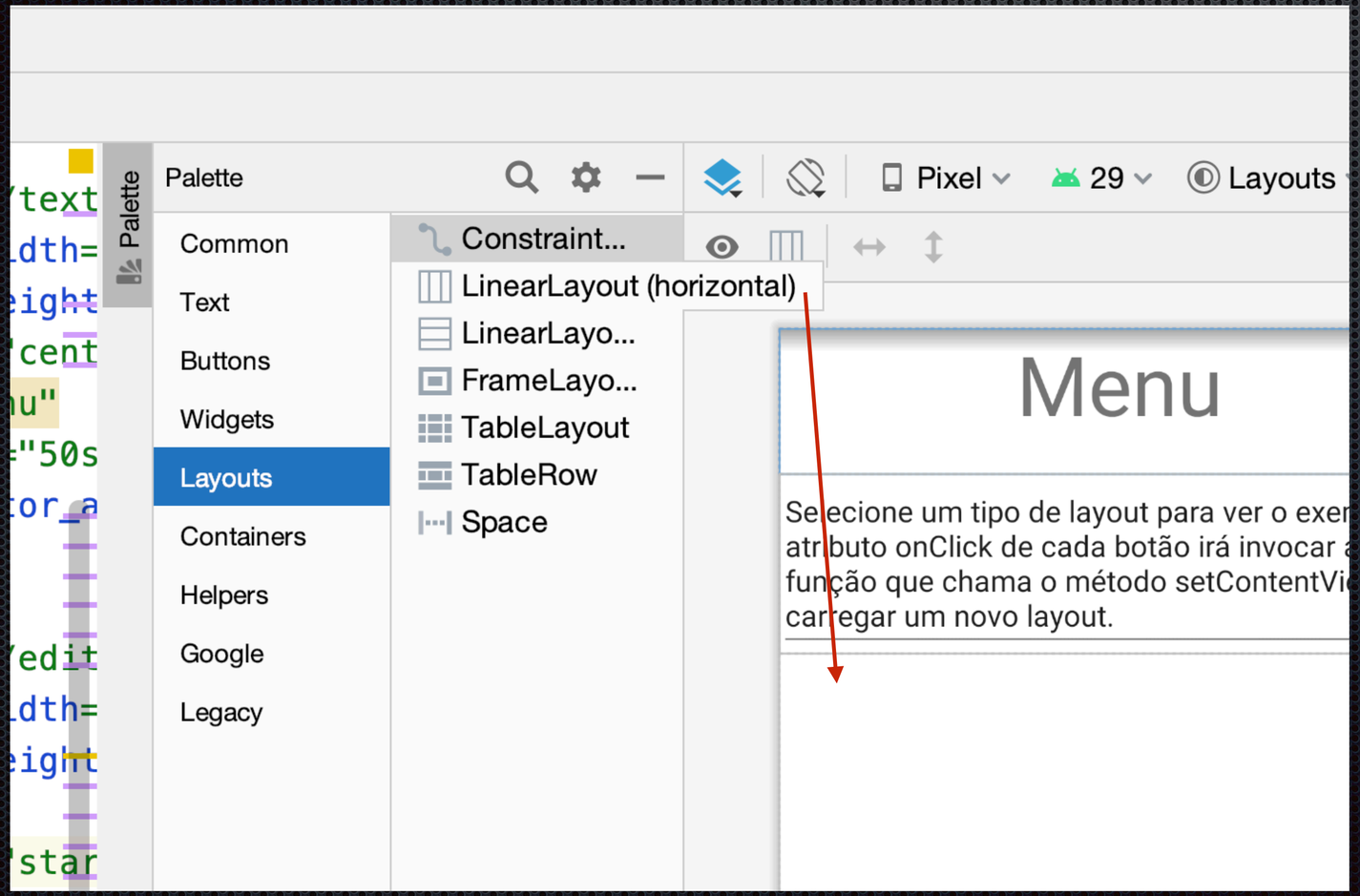
define quantos caracteres por linha

Palette

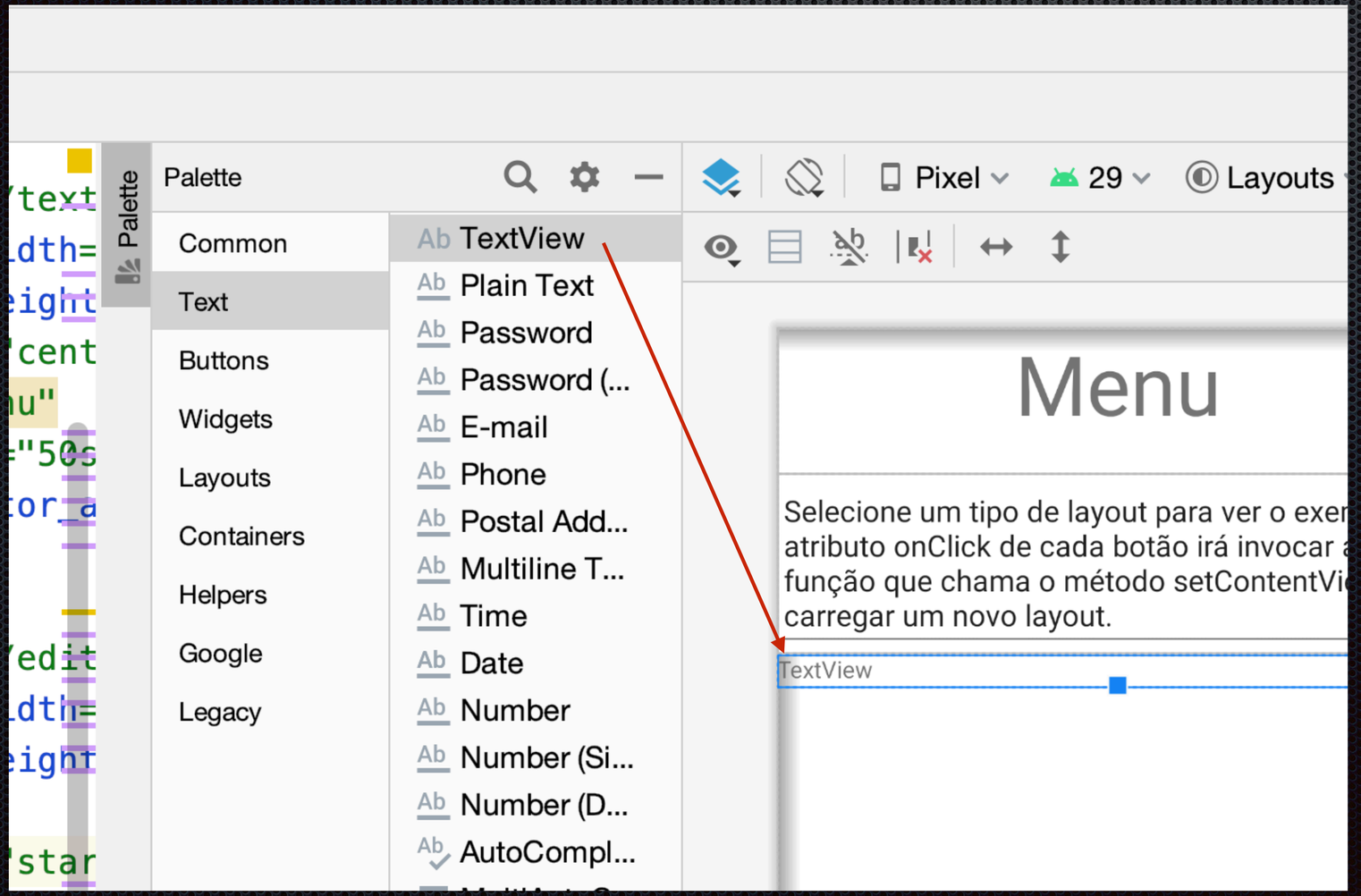
Pixel 29 Layouts 0dp



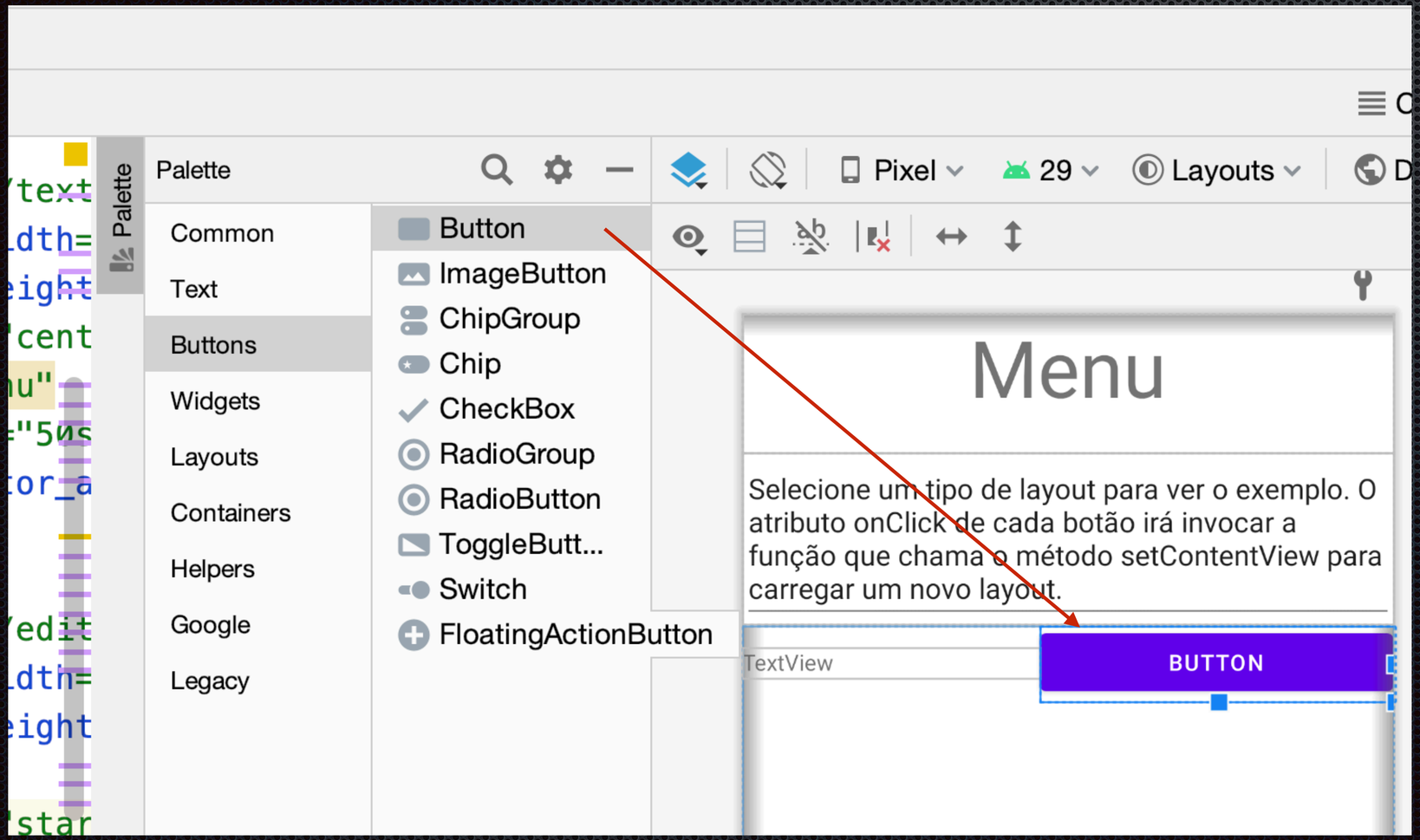
Vamos adicionar um LinearLayout abaixo do Multi-line TextView



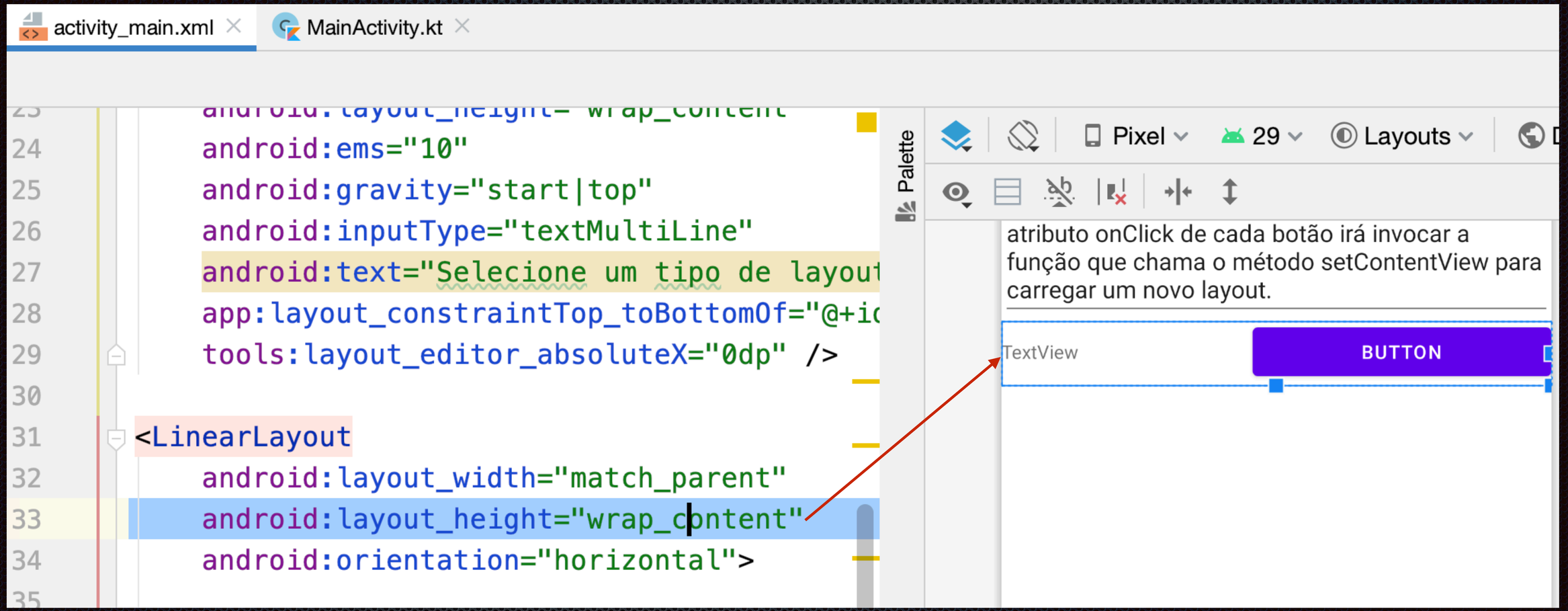
E adicionamos um TextView neste novo LinearLayout



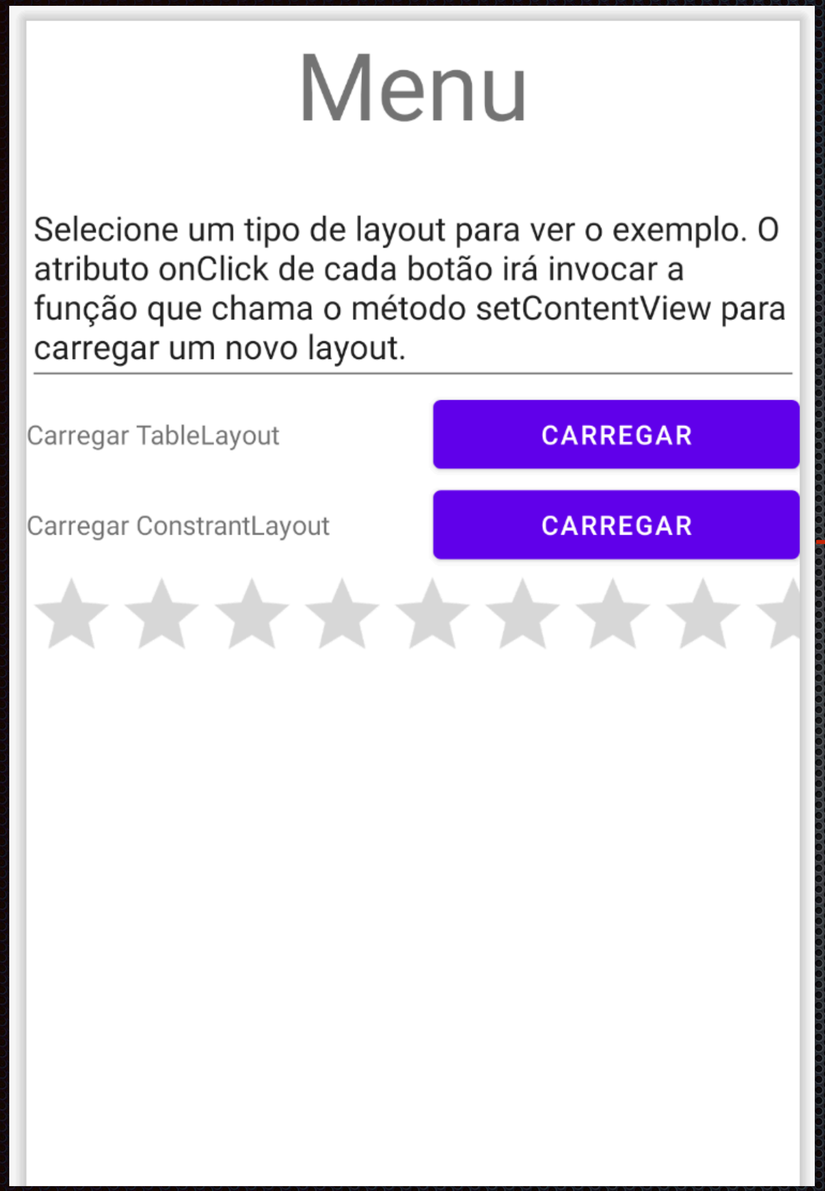
E um Button ao lado do TextView



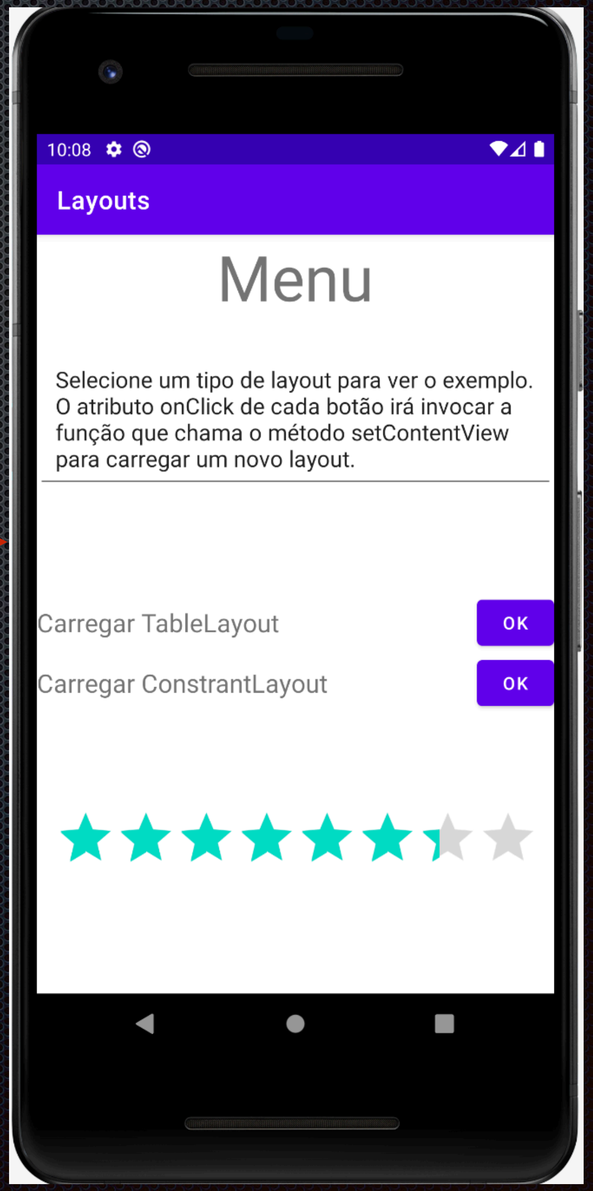
Alteramos aqui a propriedade da altura para terminar o layout ao final do tamanho dos componentes.



Vamos criar um novo LinearLayout abaixo do anterior e um RatingBar

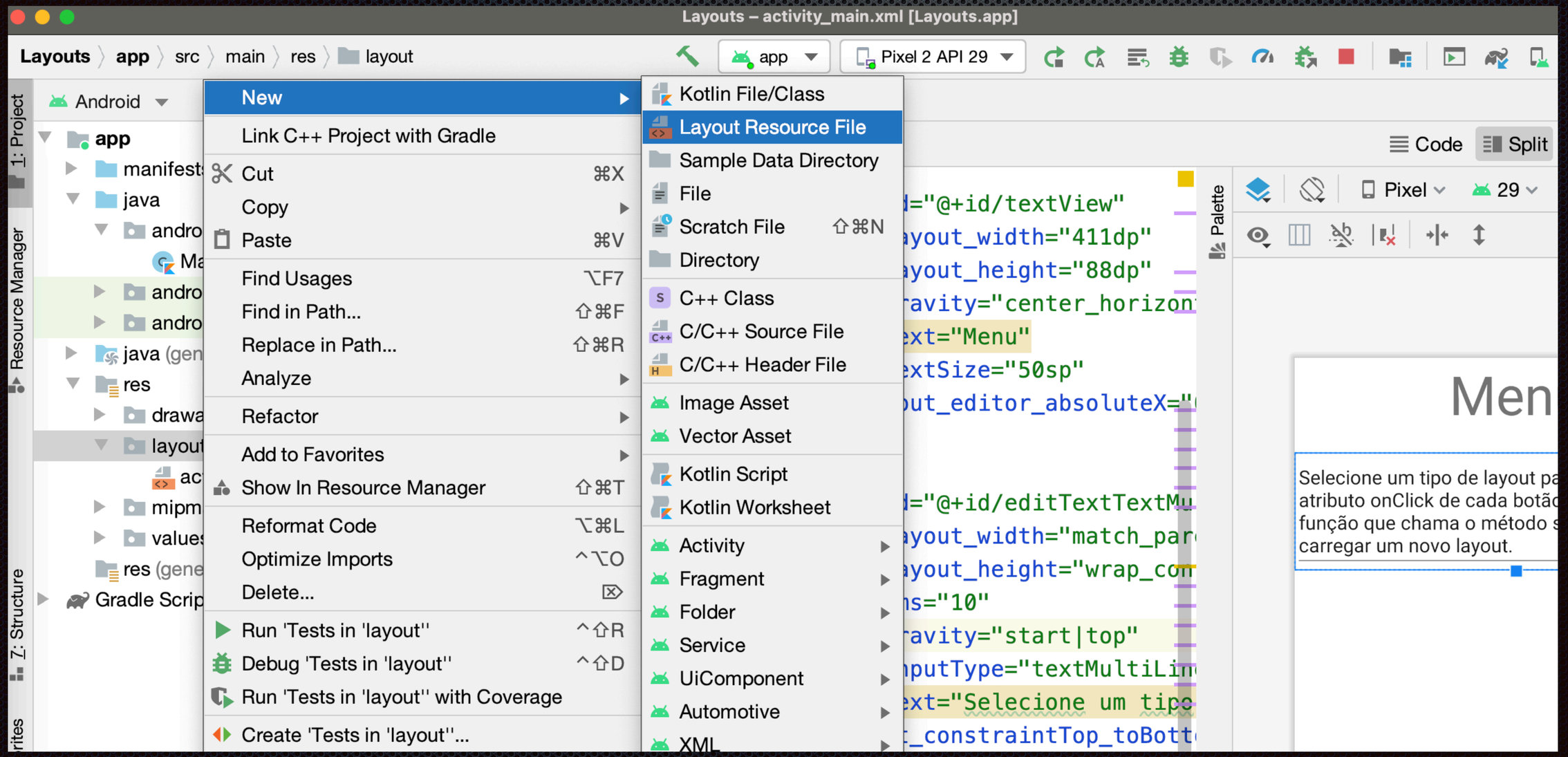


Alterando algumas propriedades...

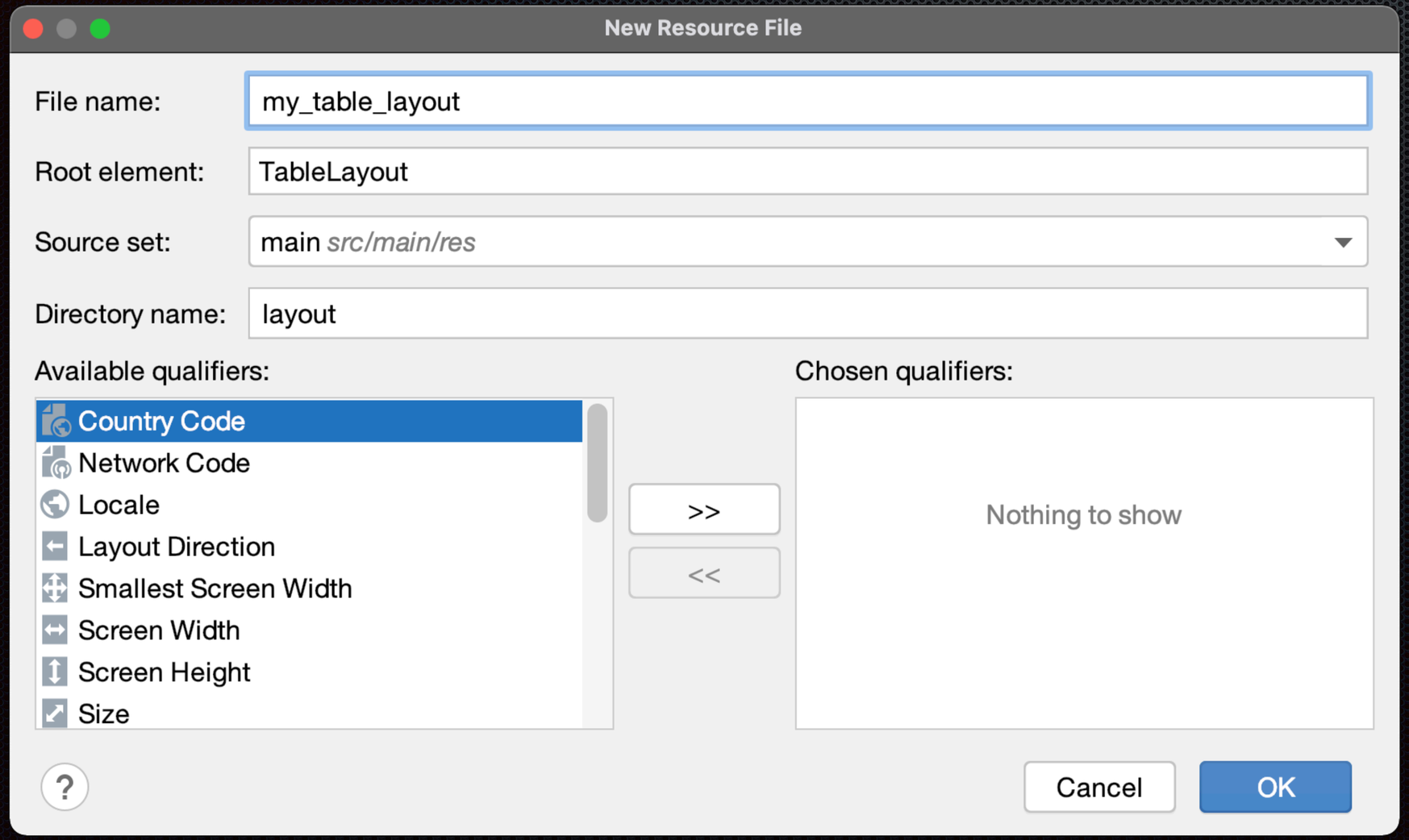


Interação entre Views

Vamos criar um novo arquivo de layout



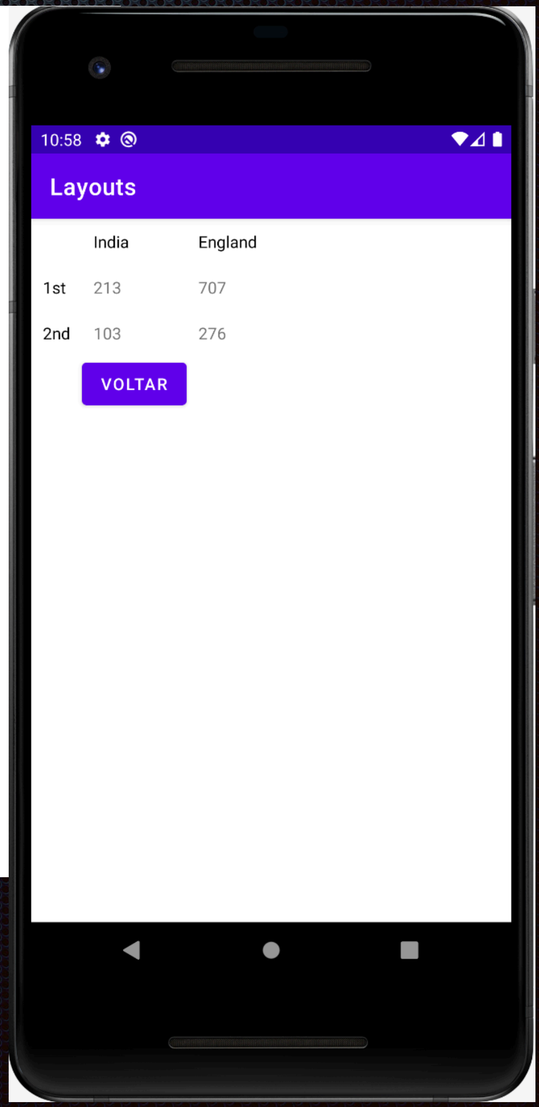
Criaremos um TableLayout



Criaremos no TableLayout uma tabela de dados

```
<TableLayout xmlns:android="http://schemas.android.com/
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TableRow
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_column="1"
            android:padding="10sp"
            android:text="India"
            android:textColor="@android:color/black" />

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_column="2"
            android:padding="10sp"
            android:text="England"
```

```
            android:layout_height="wrap_content"
            android:layout_column="2"
            android:padding="10sp"
            android:text="276" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <Button
            android:id="@+id/button4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_column="1"
            android:onClick="loadLinearLayout"
            android:text="Voltar" />
    </TableRow>
</TableLayout>
```



E criar uma função no código para carregar o novo layout

MainActivity.kt

```
1   package android.com.layouts
2
3   import androidx.appcompat.app.AppCompatActivity
4   import android.os.Bundle
5   import android.view.View
6
7   class MainActivity : AppCompatActivity() {
8       override fun onCreate(savedInstanceState: Bundle?) {
9           super.onCreate(savedInstanceState)
10          setContentView(R.layout.activity_main) //aqui def
11      }
12
13      fun loadTableLayout(v: View) {
14          setContentView(R.layout.my_table_layout)
15      }
16  }
```

E faremos o mesmo procedimento para um ConstrantLayout que conterá um CalendarView

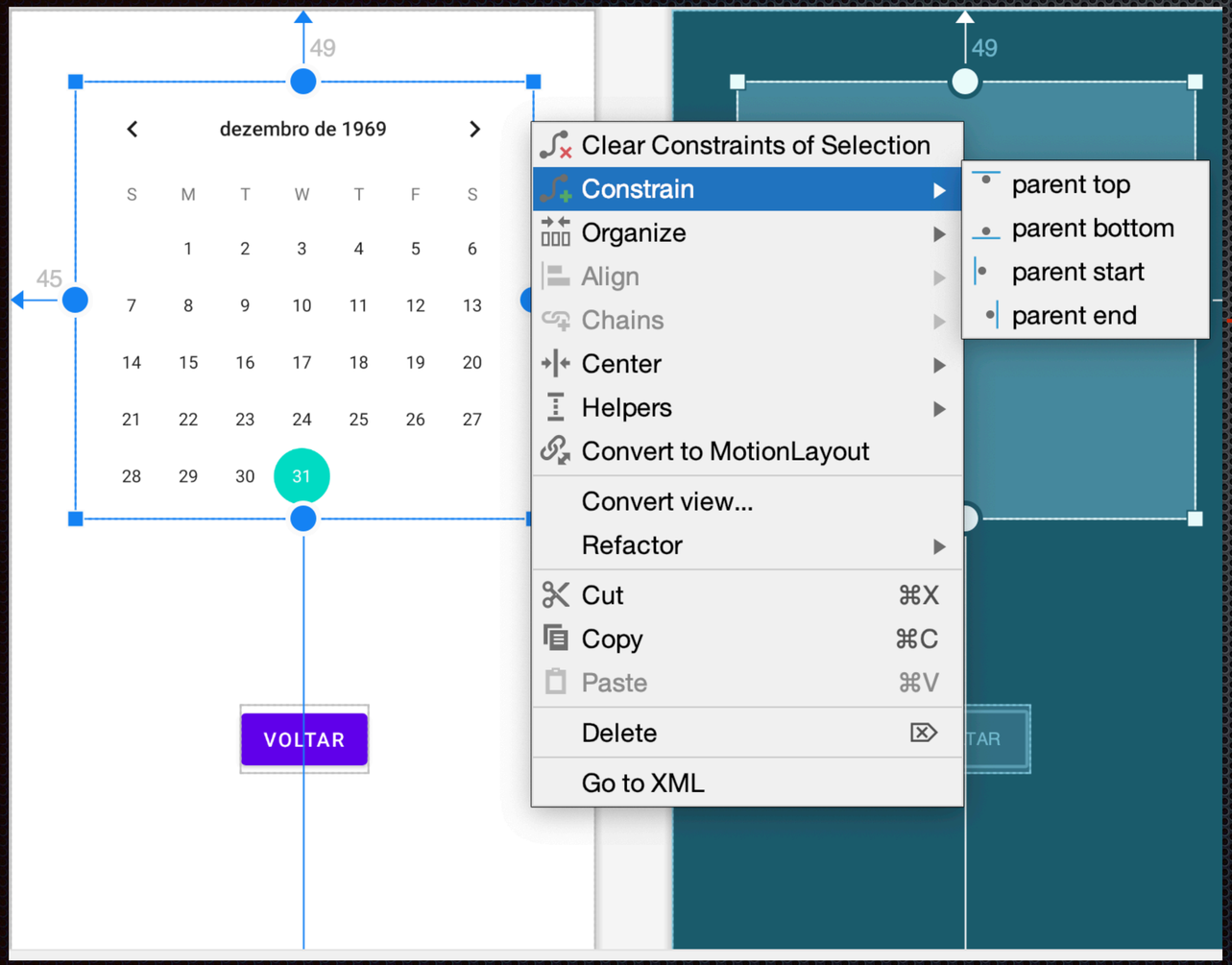
my_constrant_layout.xml | MainActivity.kt

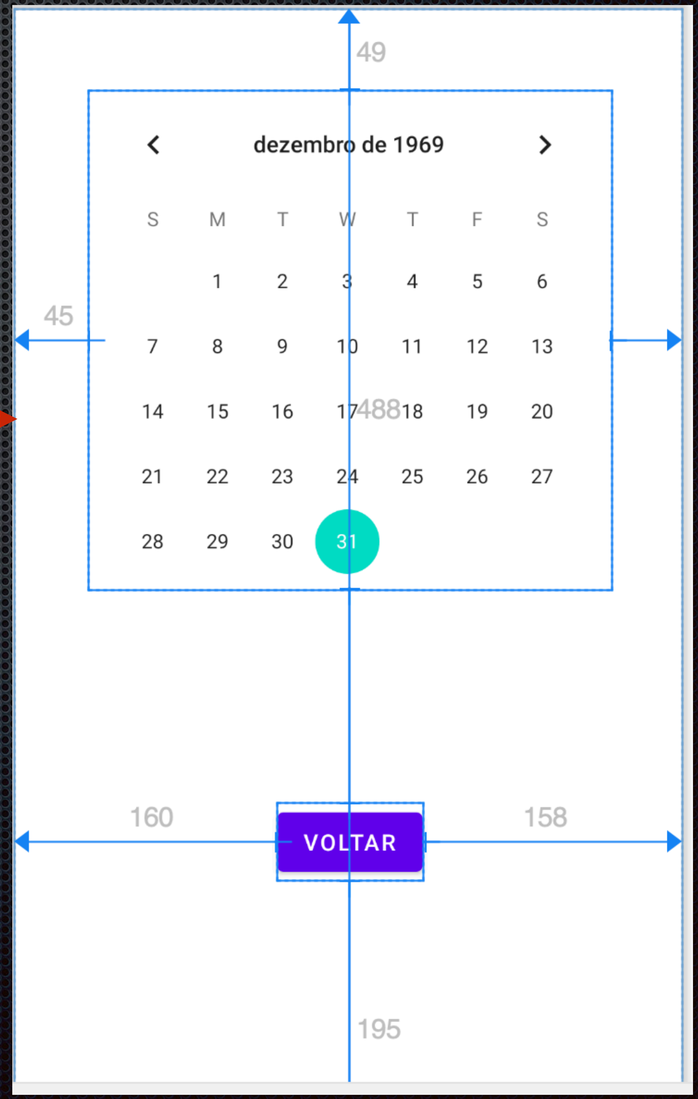
```kotlin
package android.com.layouts

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View

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

    fun loadTableLayout(v: View) {
        setContentView(R.layout.my_table_layout)
    }

    fun loadConstrantLayout(v: View) {
        setContentView(R.layout.my_constrant_layout)
    }
}
```

my_constrant_layout.xml | MainActivity.kt

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/
    android:layout_height="match_parent">

</androidx.constraintlayout.widget.ConstraintLayout>
```

Podemos usar restrições (constraints) para deixar fixos os componentes em proporção aos limites da tela.

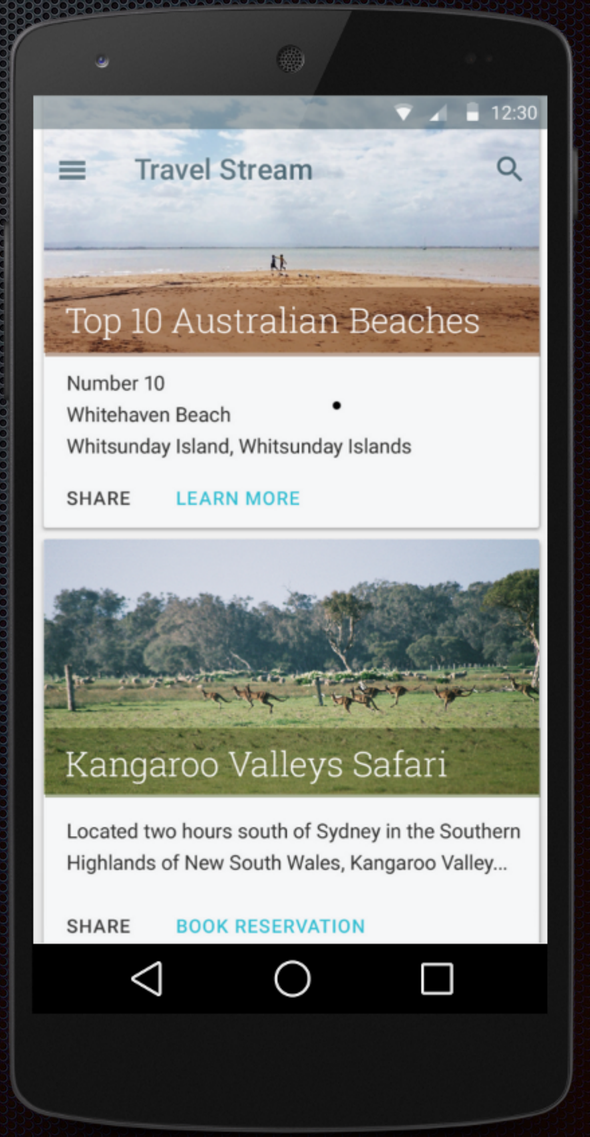




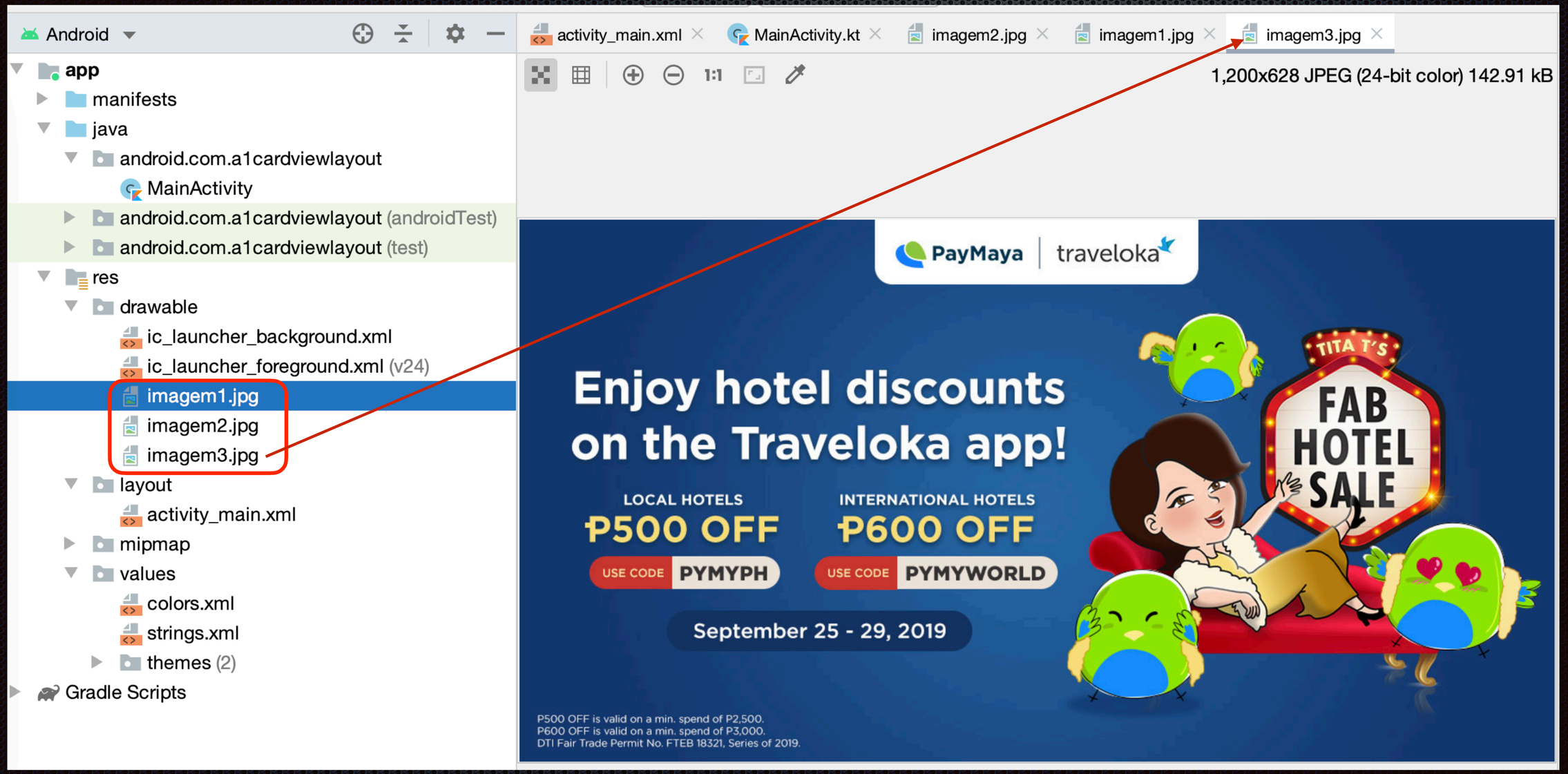
CardView

Layout Baseado em Cartões (Cards)

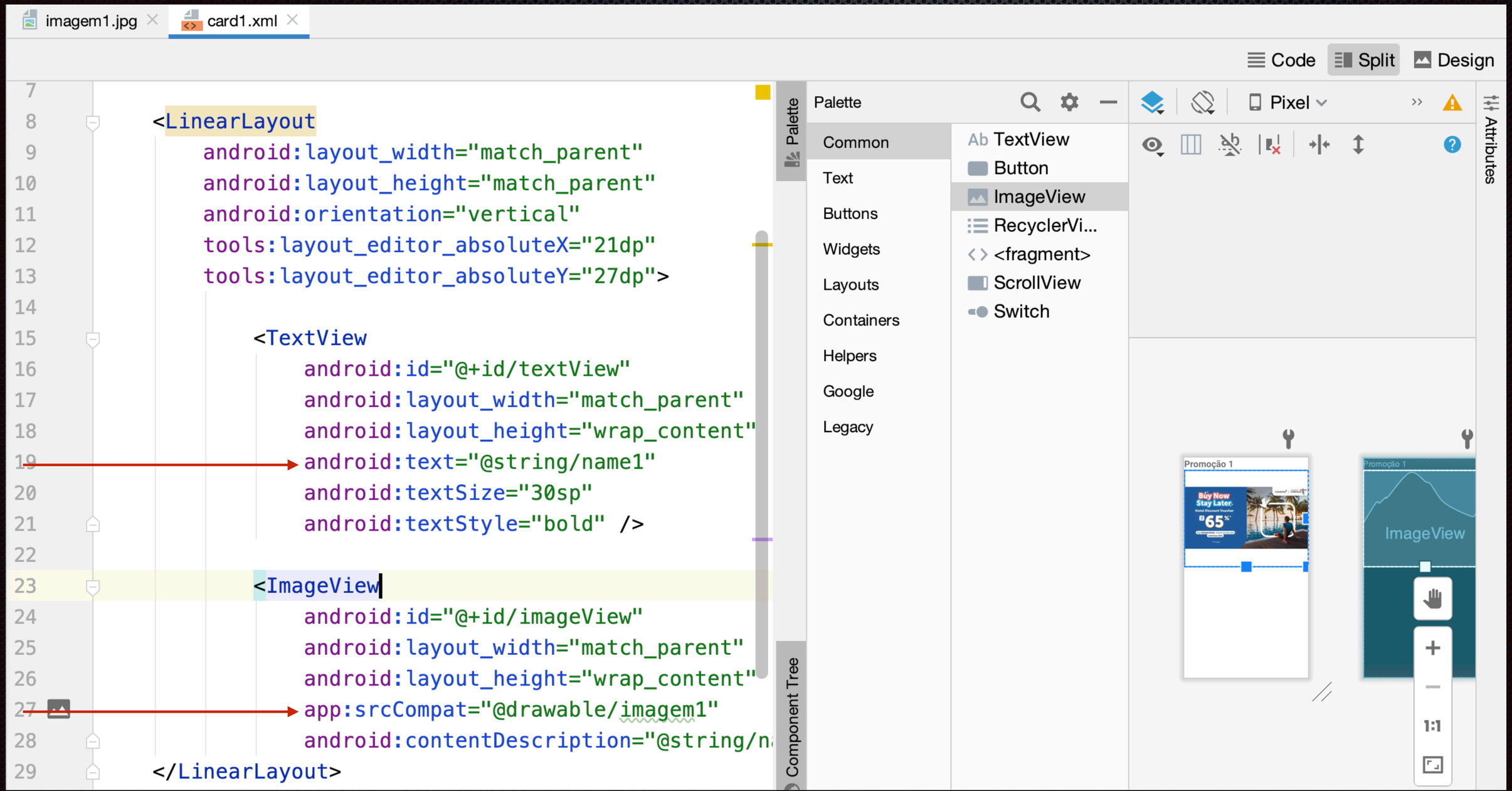
- Muitas vezes, os apps precisam exibir dados em contêineres com estilo semelhante.
- Esses contêineres são frequentemente usados em listas para armazenar as informações de cada item.
- O sistema fornece a API CardView como uma maneira fácil de mostrar informações dentro de cards que têm uma aparência consistente em toda a plataforma.



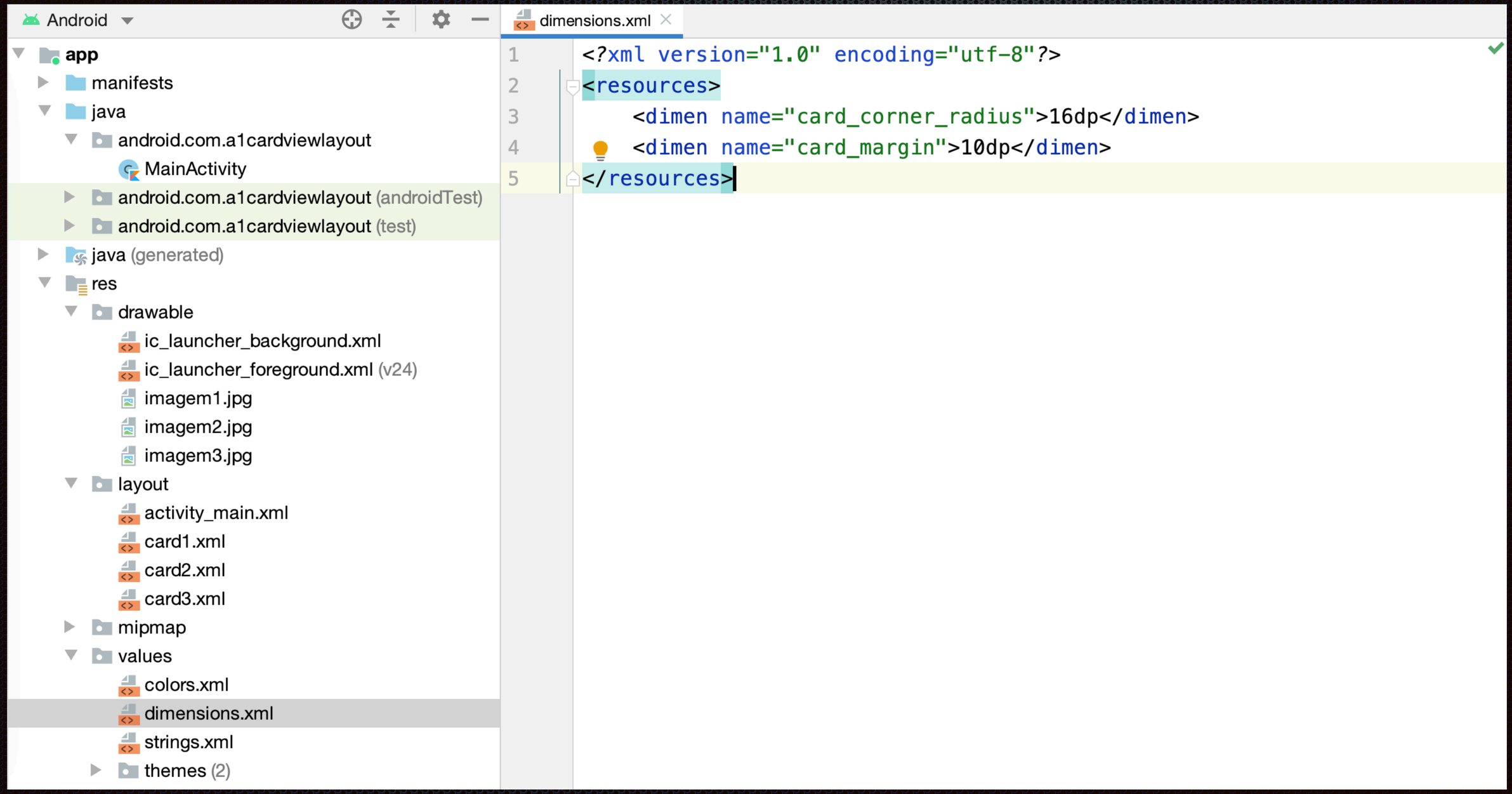
Precisamos adicionar algumas imagens à pasta drawable



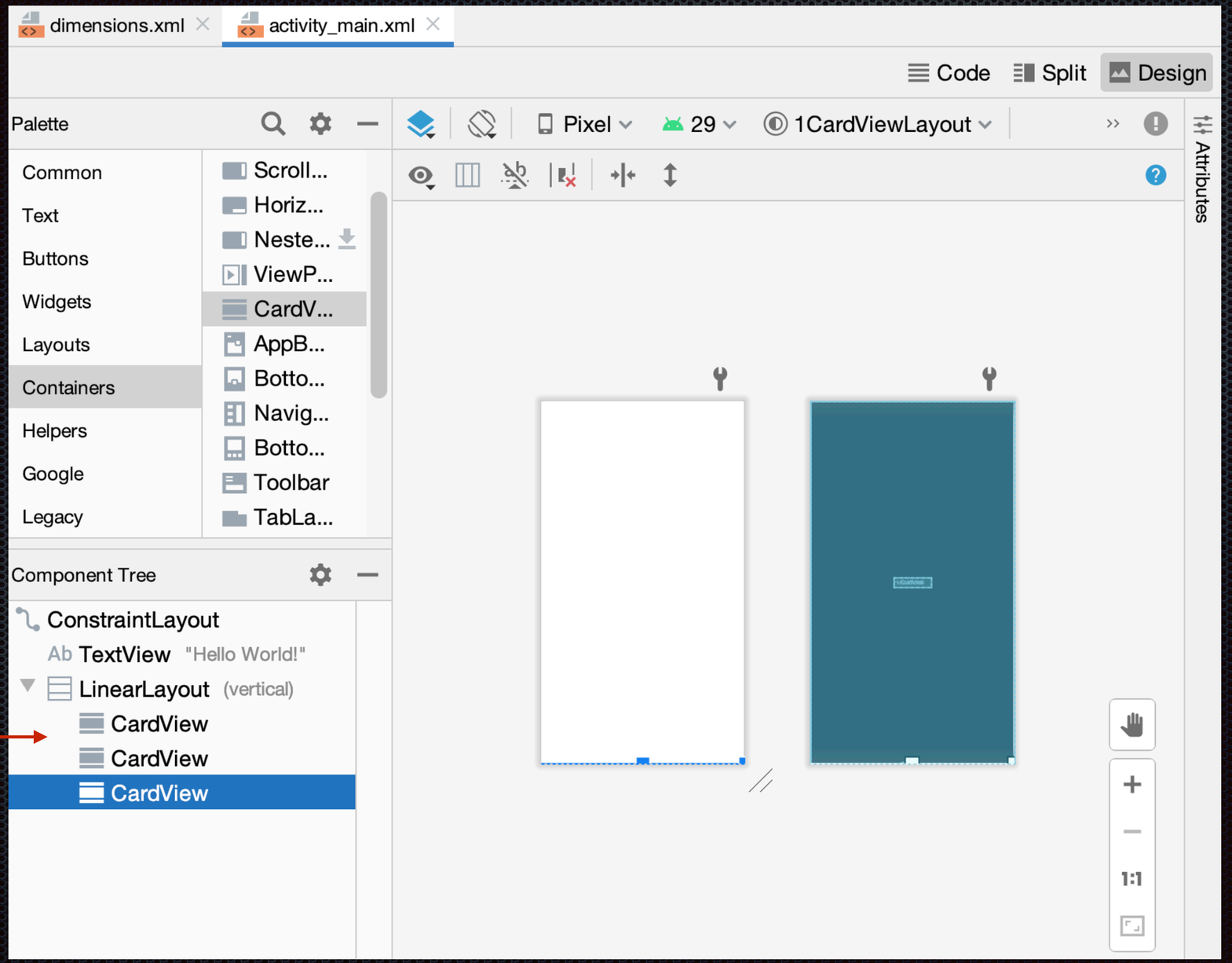
Criaremos um Arquivo de Layout para cada Imagem



Vamos um criar um arquivo de dimensões



E no arquivo principal, adicionar 3 CardViews em 1 LinearLayout



Usaremos o xml de dimensões para definir o raio e as margens

dimensions.xml | activity_main.xml

Code | Split | Design

```
9     <LinearLayout
10        android:layout_width="match_parent"
11        android:layout_height="match_parent"
12        android:orientation="vertical"
13        tools:layout_editor_absoluteX="150dp"
14        tools:layout_editor_absoluteY="217dp">
15
16        <androidx.cardview.widget.CardView
17            android:layout_width="match_parent"
18            android:layout_height="wrap_content"
19            android:layout_gravity="center"
20            android:layout_margin="@dimen/card_margin"
21            app:cardCornerRadius="@dimen/card_corner_radius"
22            app:cardElevation="2dp" />
23
24        <androidx.cardview.widget.CardView
25            android:layout_width="match_parent"
26            android:layout_height="wrap_content"
27            android:layout_gravity="center"
28            android:layout_margin="@dimen/card_margin"
29            app:cardCornerRadius="@dimen/card_corner_radius"
30            app:cardElevation="2dp" />
31
```

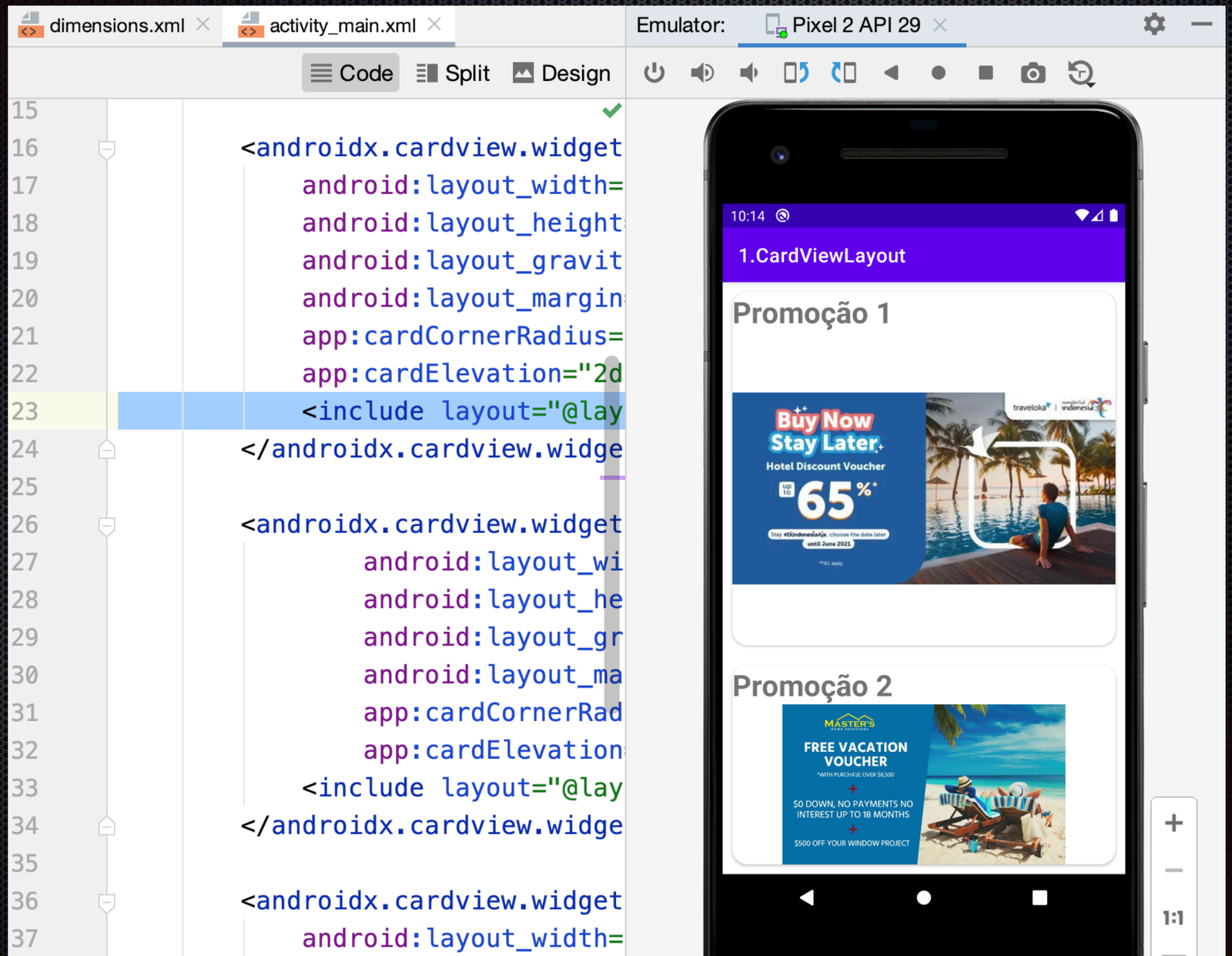
Adicionaremos os arquivos de layouts dentro dos Containers CardView

dimensions.xml | activity_main.xml

Code | Split | Design

```
15
16      <androidx.cardview.widget.CardView
17          android:layout_width="match_parent"
18          android:layout_height="wrap_content"
19          android:layout_gravity="center"
20          android:layout_margin="@dimen/card_margin"
21          app:cardCornerRadius="@dimen/card_corner_radius"
22          app:cardElevation="2dp">
23          <include layout="@layout/card1" />
24      </androidx.cardview.widget.CardView>
25
26      <androidx.cardview.widget.CardView
27              android:layout_width="match_parent"
28              android:layout_height="wrap_content"
29              android:layout_gravity="center"
30              android:layout_margin="@dimen/card_margin"
31              app:cardCornerRadius="@dimen/card_corner_radius"
32              app:cardElevation="2dp">
33          <include layout="@layout/card2" />
34      </androidx.cardview.widget.CardView>
35
36      <androidx.cardview.widget.CardView
37          android:layout_width="match_parent"
```

Agora temos o layout criado, mas não conseguimos rolar a tela para ver a promoção 3



ScrollView

Para isto, precisamos adicionar um ScrollView para ter a barra de rolagem

activity_main.xml

Code | Split | Design

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://
    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=".MainActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            tools:layout_editor_absoluteX="150dp"
            tools:layout_editor_absoluteY="217dp">

            <androidx.cardview.widget.CardView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:layout_margin="@dimen/card_margin"
```

Emulator: Pixel 2 API 29

Links

- https://www.jetbrains.com/help/idea/reformat-and-rearrange-code.html
- https://github.com/udacity/Court-Counter