Programación en Android II
Layouts

Programación Multimedia y Dispositivos Móviles

2º Técnico en Desarrollo de Aplicaciones Multiplataforma

Antes de empezar

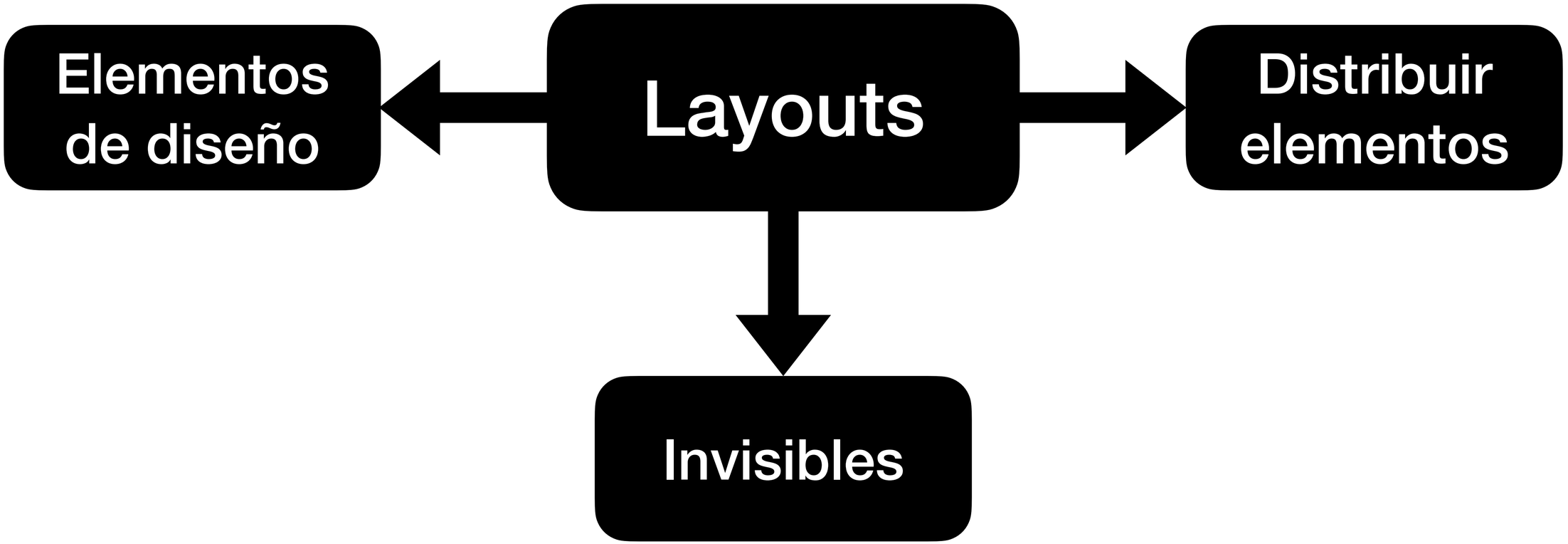
Antes de empezar

Antes de empezar

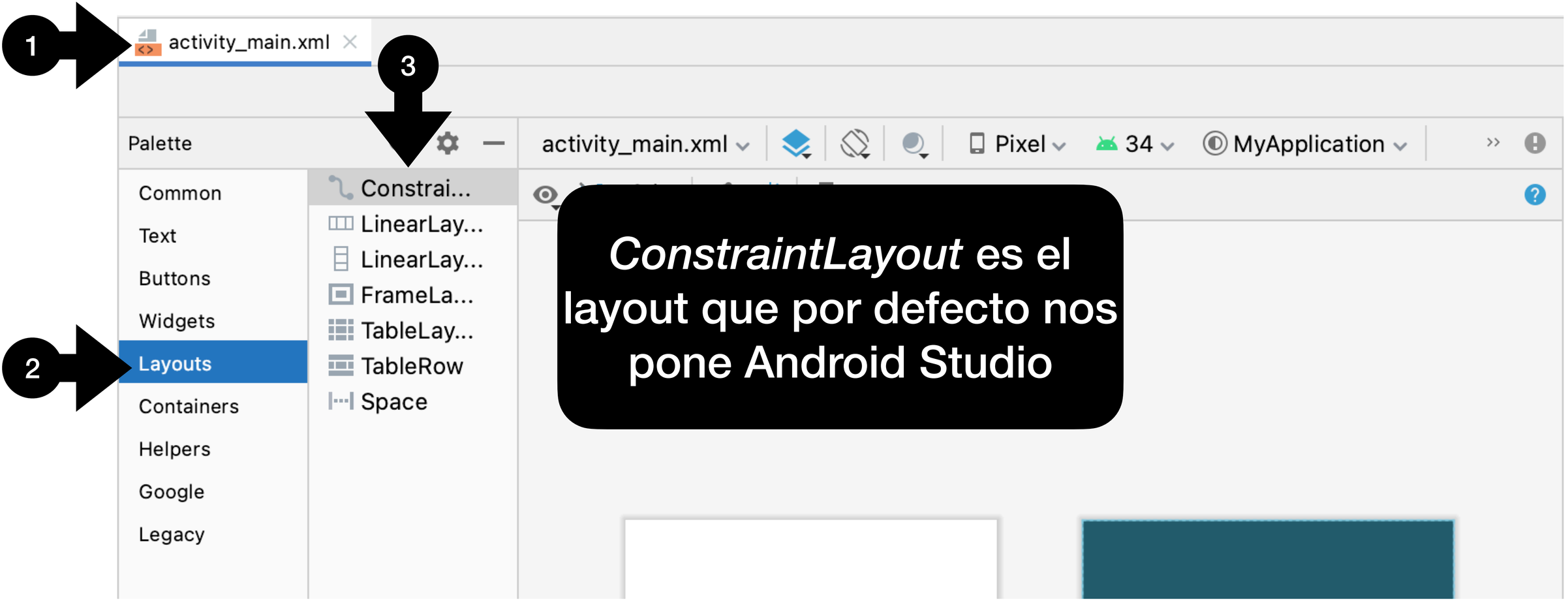
¿Qué vamos a ver?

1. Introducción
2. LinearLayout
3. TableLayout
4. FrameLayout
5. ConstraintLayout
6. Pestañas

1 Introducción



1 Introducción



Utilizaremos:

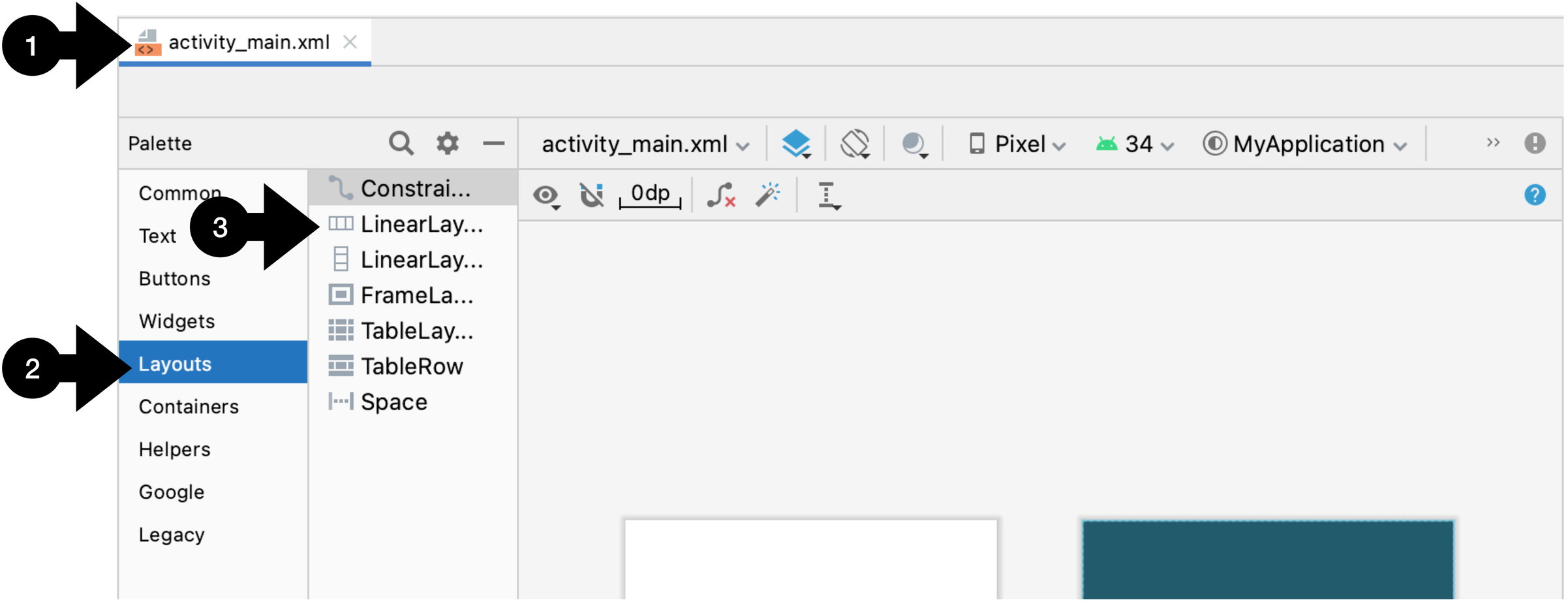
TableLayout

LinearLayout

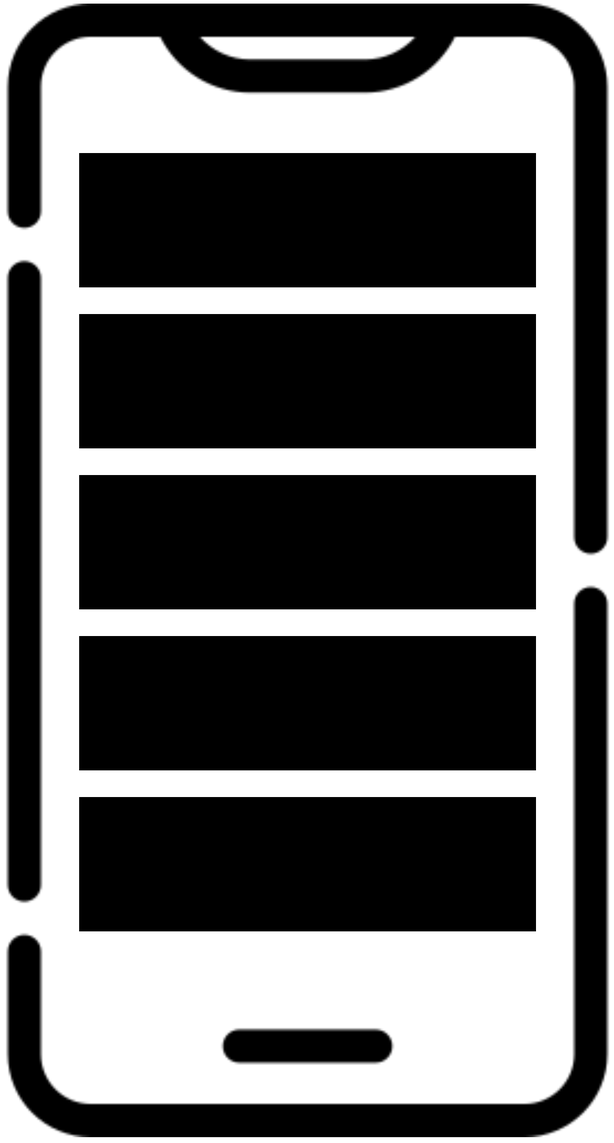
FrameLayout

ConstraintLayout

2 LinearLayout

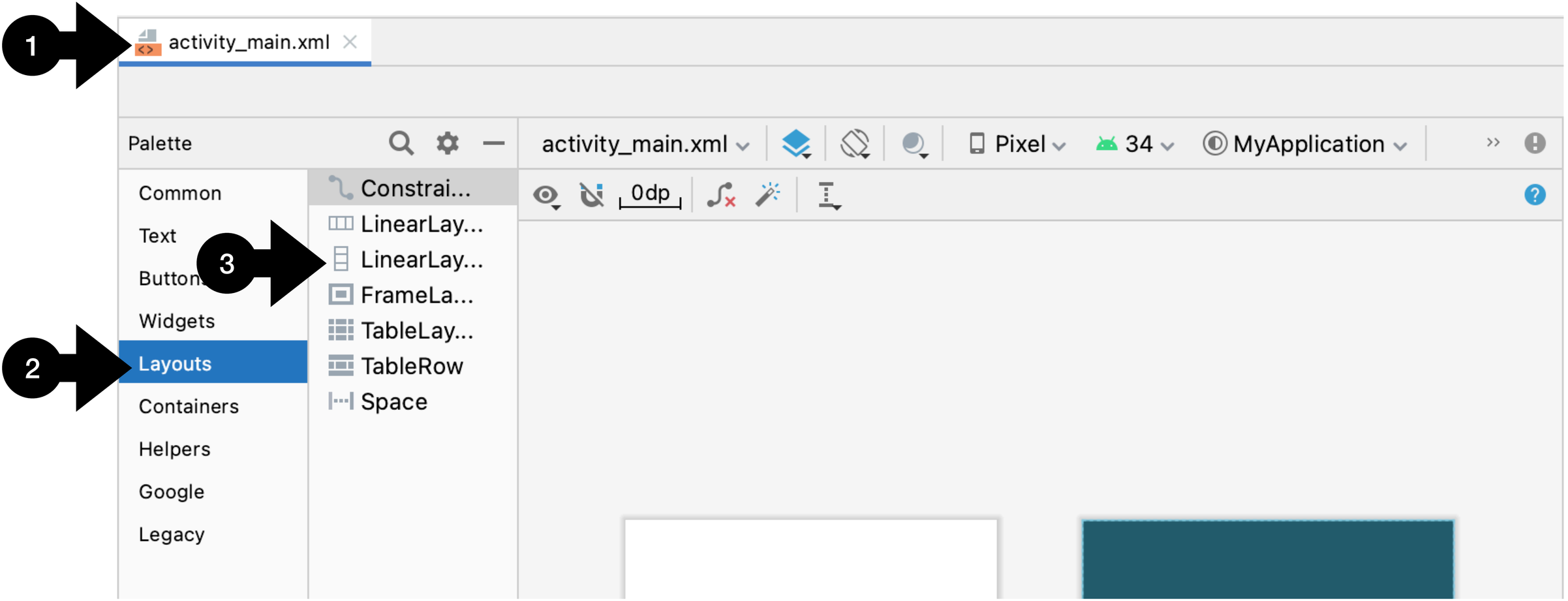


2 LinearLayout

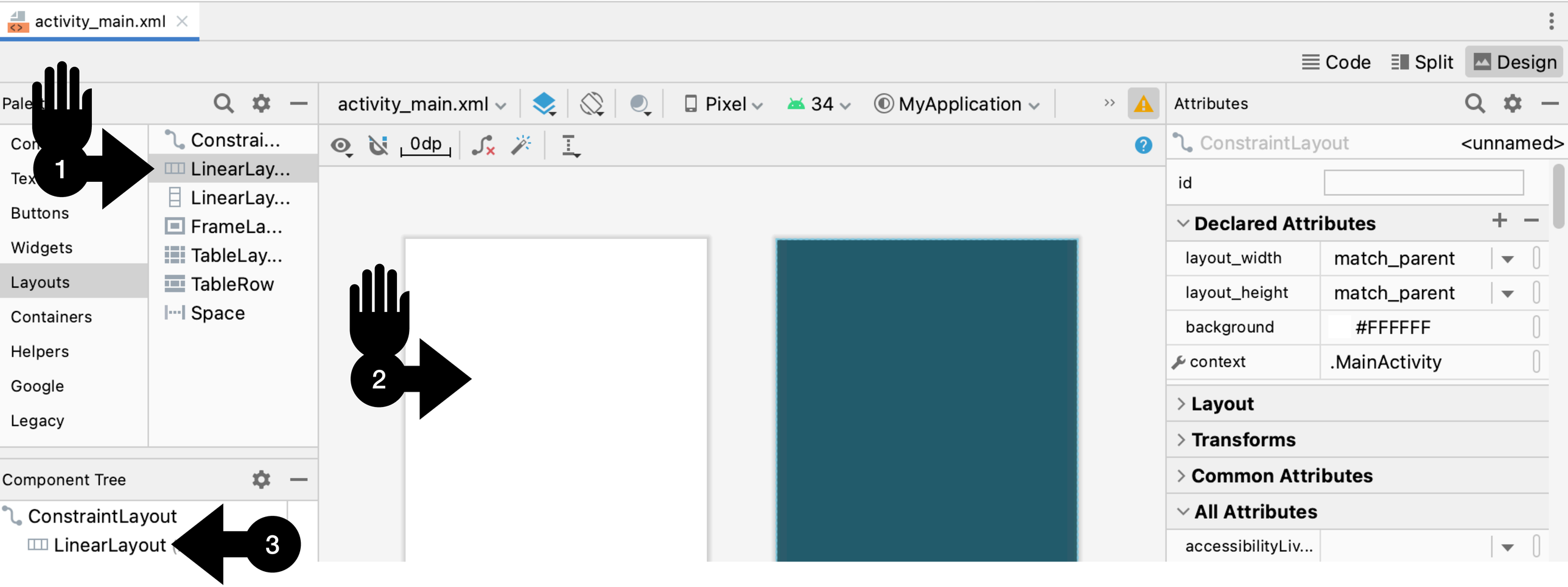


Divide los elementos de forma consecutiva

2 LinearLayout



2 LinearLayout



2 LinearLayout

activity_main.xml

1 Code Split Design

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    android:background="#FFFFFF"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

2

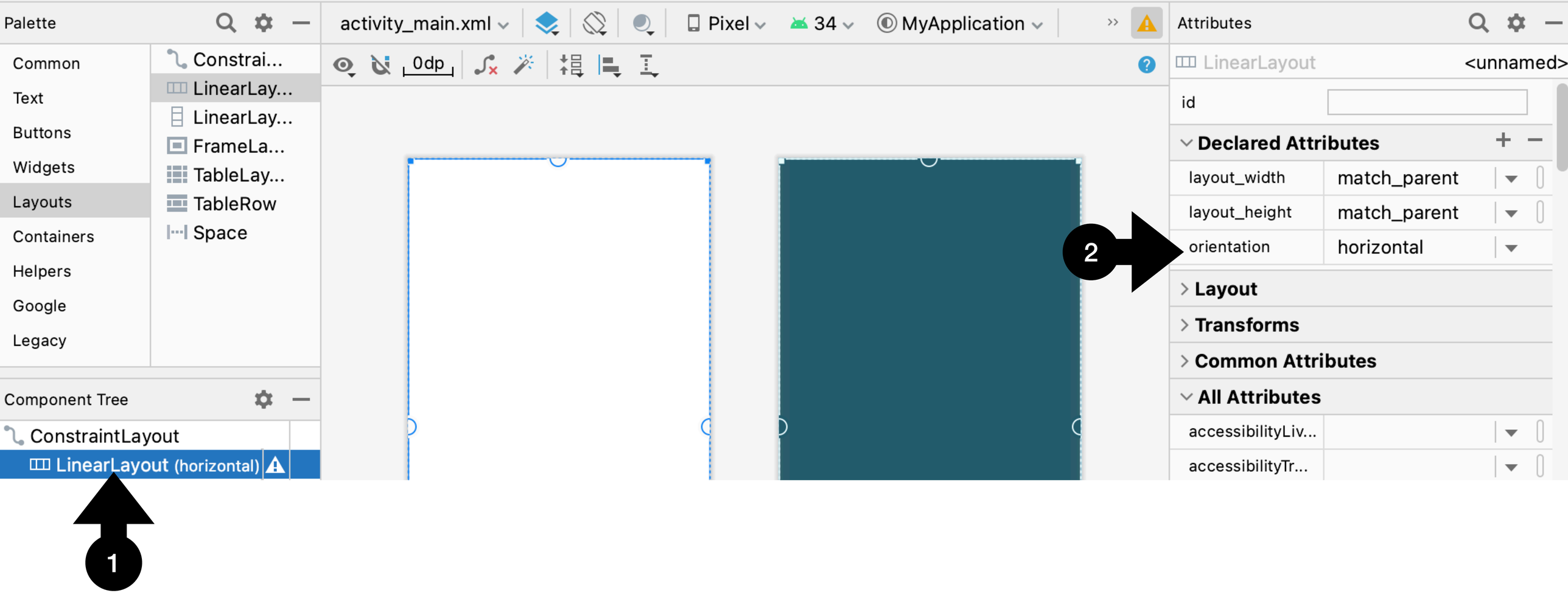
2 LinearLayout

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

android:orientation=“horizontal”

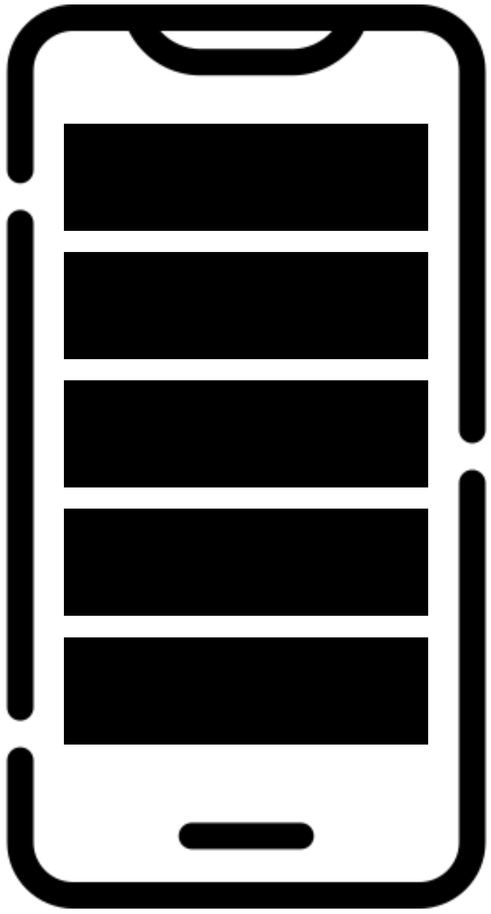
android:orientation=“vertical”

2 LinearLayout

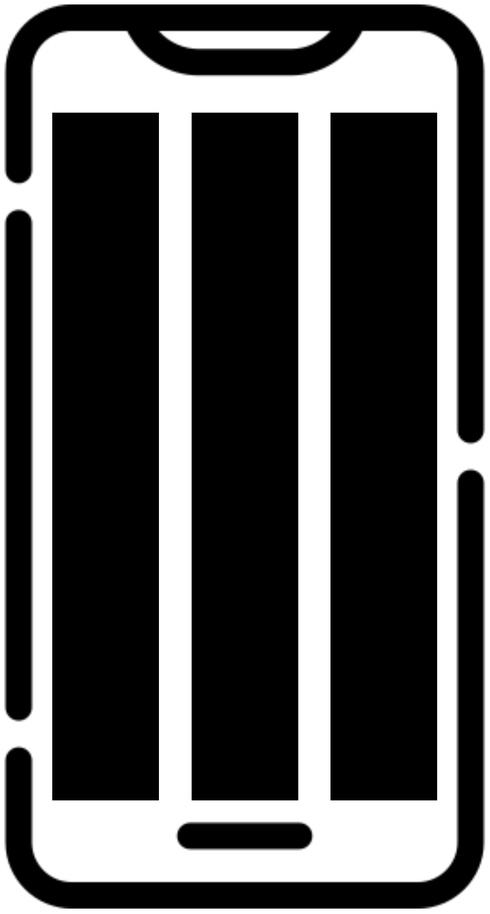


2 LinearLayout

Vertical

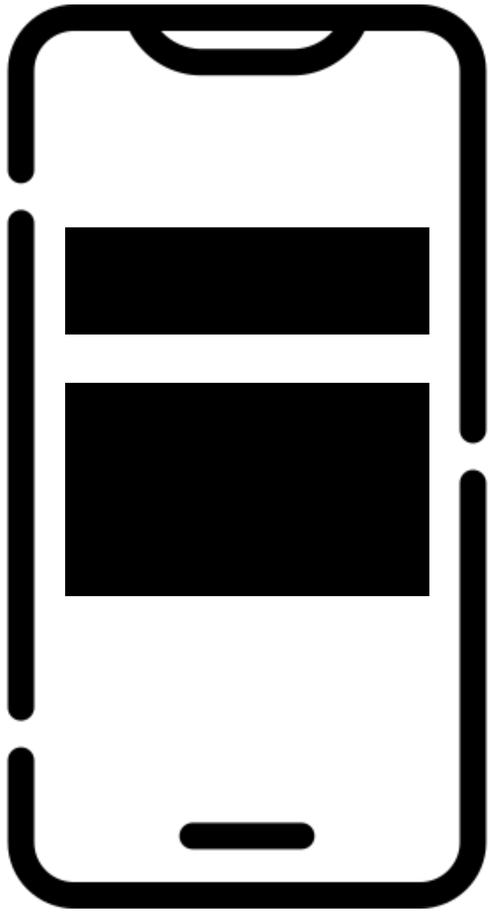


Horizontal



2 LinearLayout

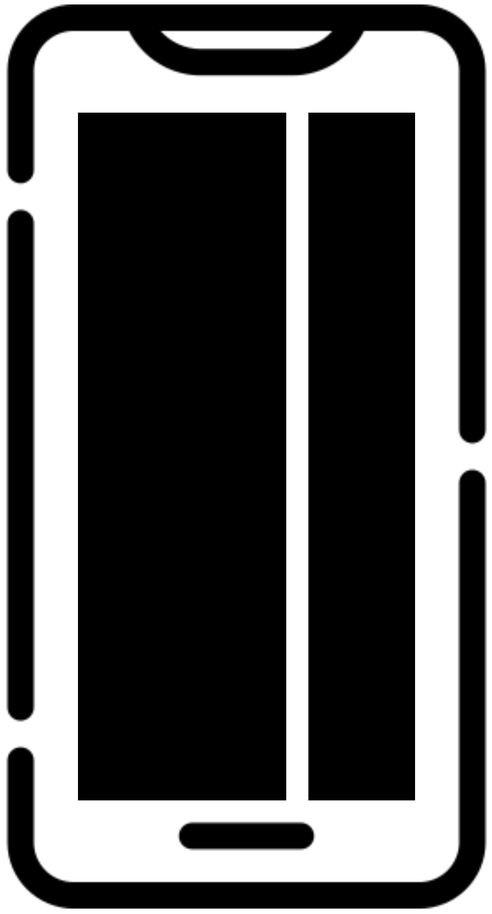
Vertical



Podemos definir que utilicen tamaños concretos proporcionales

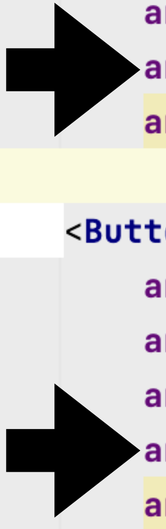
android:layout_weight

Horizontal



activity_main.xml

```
14
15      <Button
16          android:id="@+id/button"
17          android:layout_width="match_parent"
18          android:layout_height="wrap_content"
19          android:layout_weight="1"
20          android:text="Button" />
21
22      <Button
23          android:id="@+id/button2"
24          android:layout_width="match_parent"
25          android:layout_height="wrap_content"
26          android:layout_weight="1"
27          android:text="Button" />
28
29      <Button
30          android:id="@+id/button3"
31          android:layout_width="match_parent"
32          android:layout_height="wrap_content"
33          android:layout_weight="1"
34          android:text="Button" />
35  </LinearLayout>
36 </androidx.constraintlayout.widget.ConstraintLayout>
```



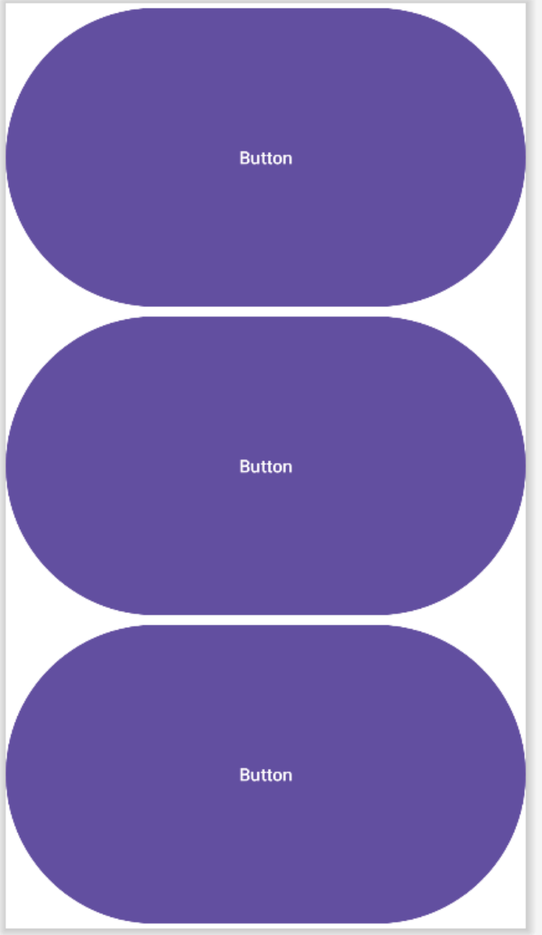
Palette

Component Tree

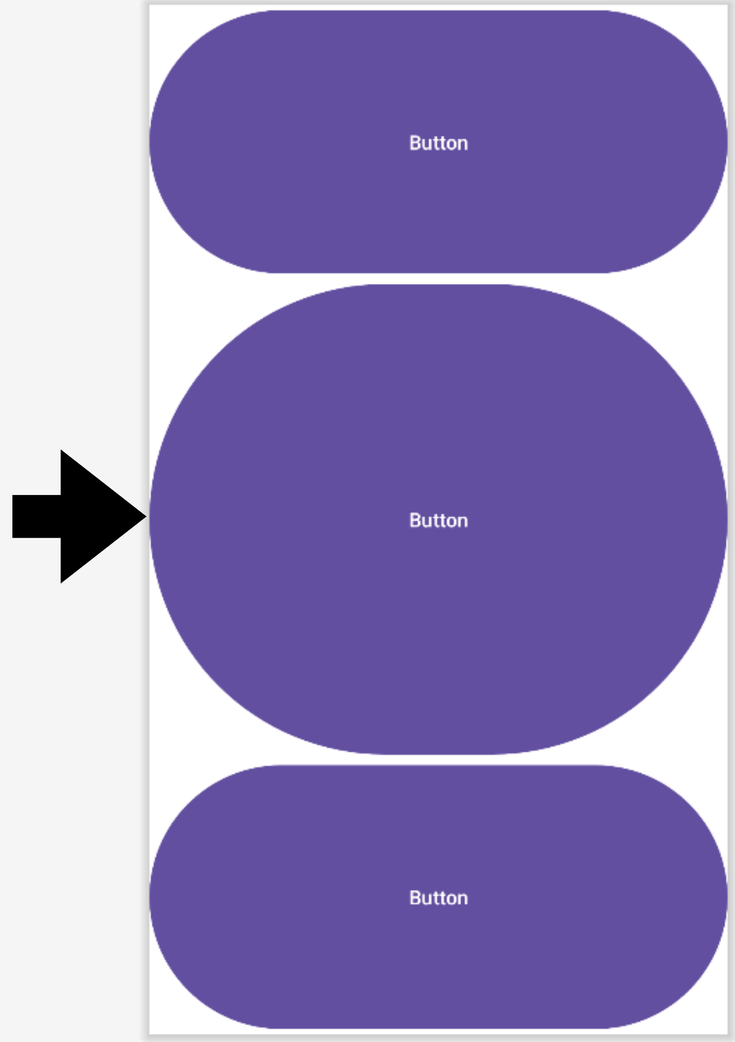
activity_main.xml

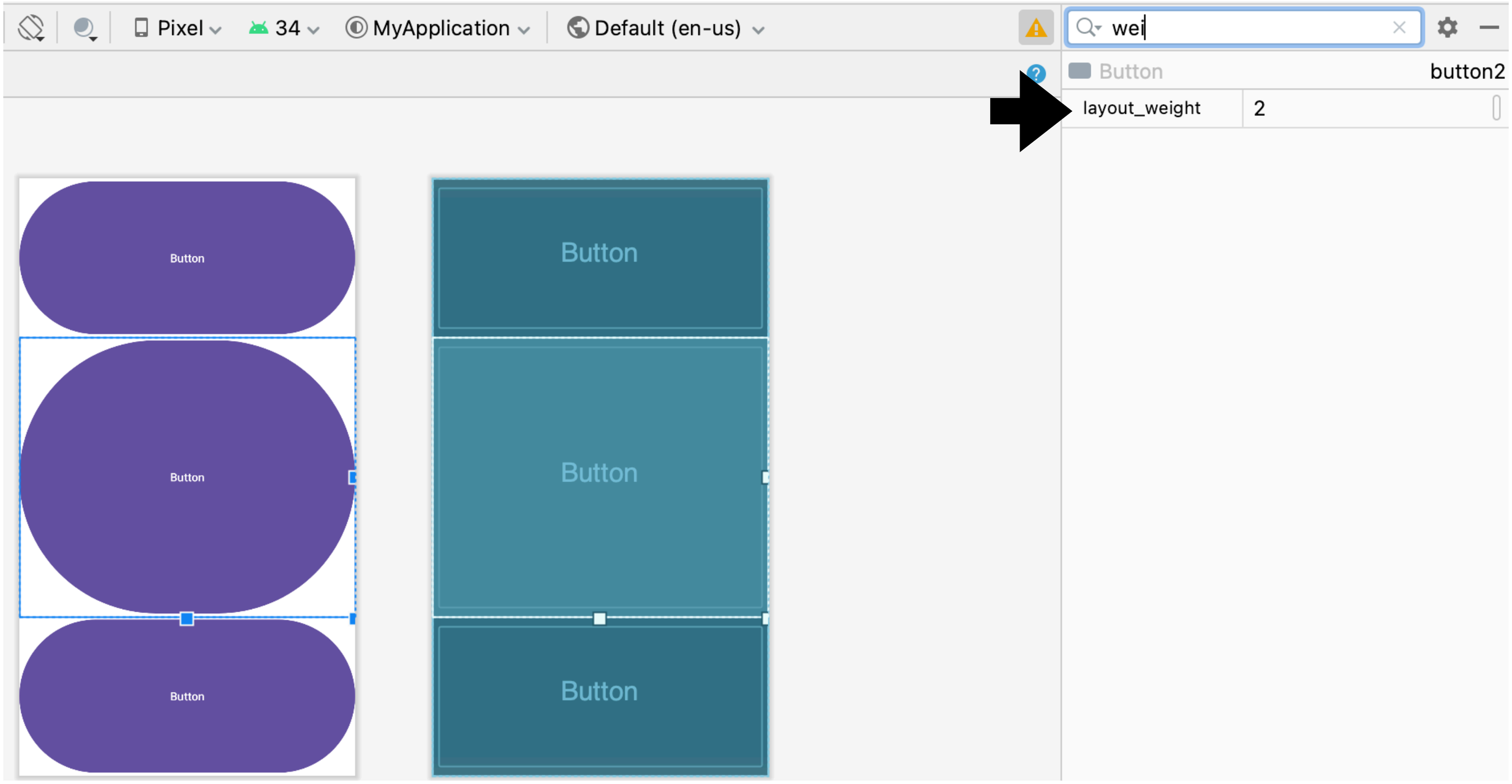
Pixel

0dp



```
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="Button" />

    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

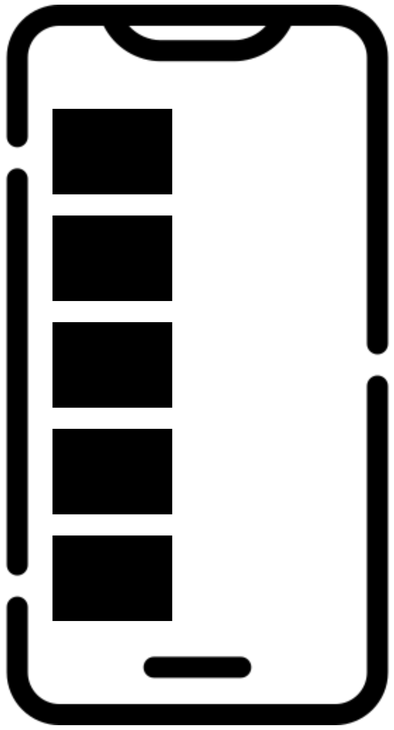


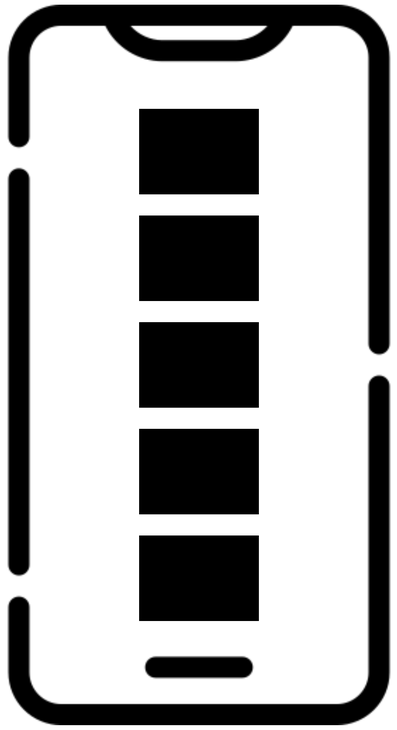

Pixel
34
MyApplication
Default (en-us)
wei
Button
button2
layout_weight
2
Button
Button
Button
Button
Button
Button

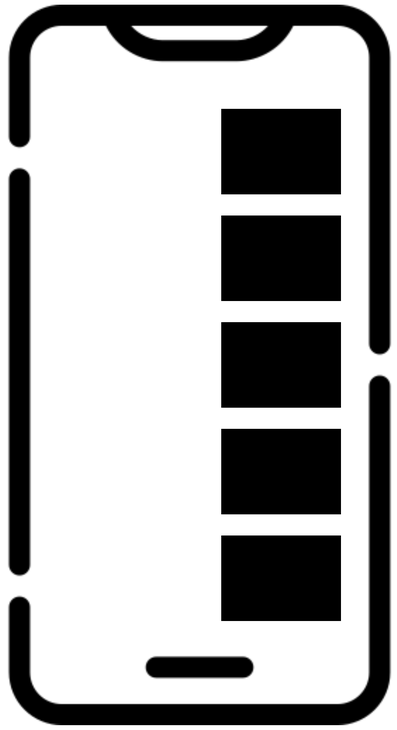
2 LinearLayout

Podemos elegir la alineación de los elementos

android:gravity





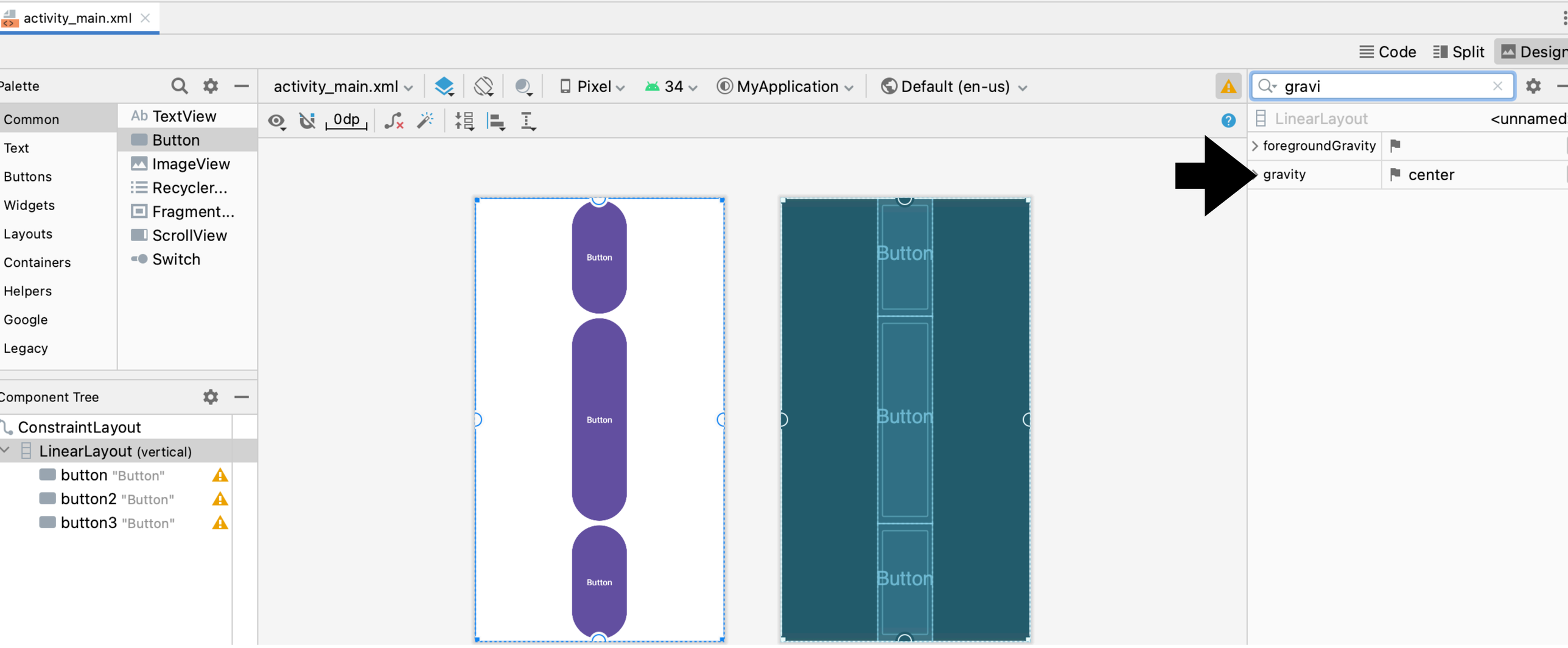


2 LinearLayout

activity_main.xml

```
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:background="#FFFFFF"
8     tools:context=".MainActivity">
9
10    <LinearLayout
11        android:layout_width="match_parent"
12        android:layout_height="match_parent"
13        android:orientation="vertical"
14        android:gravity="center">
```

4

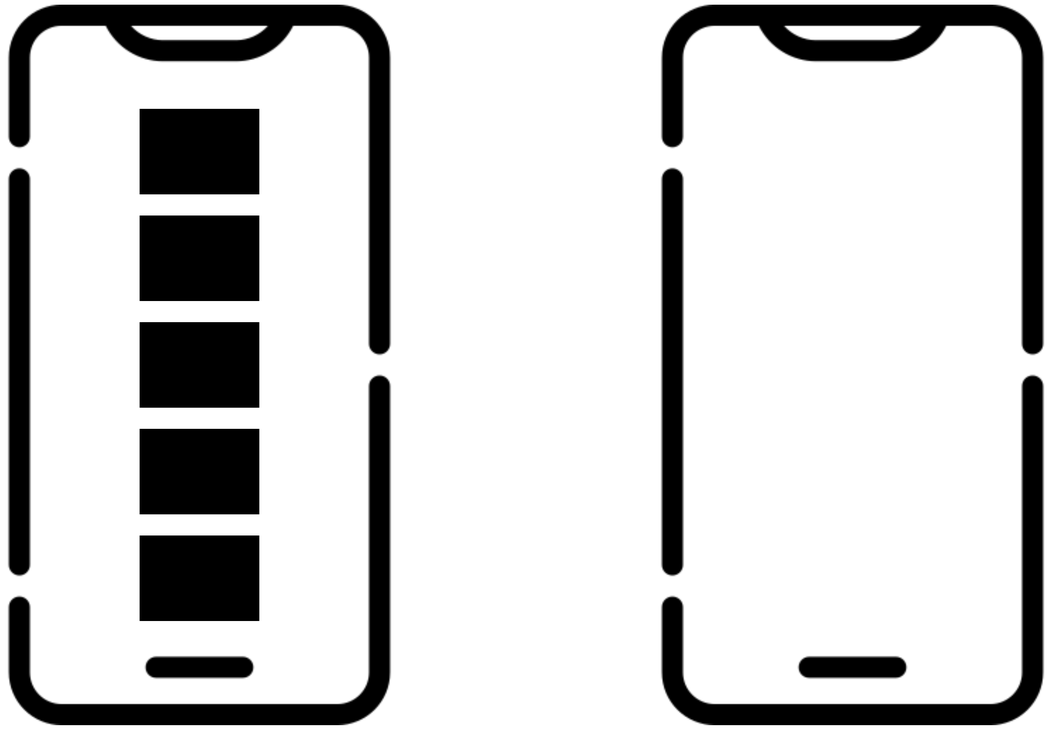

activity_main.xml
Code
Split
Design
Palette
activity_main.xml
Pixel
34
MyApplication
Default (en-us)
gravi
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
TextView
Button
ImageView
Recycler...
Fragment...
ScrollView
Switch
0dp
LinearLayout
<unnamed
foregroundGravity
gravity
center
Component Tree
ConstraintLayout
LinearLayout (vertical)
button "Button"
button2 "Button"
button3 "Button"
Button
Button
Button
Button
Button
Button

- **Center:** Centra los elementos dentro del Layout tanto de forma horizontal como vertical.
- **Top:** alinea arriba los elementos dentro del Layout.
- **Bottom:** Alinea abajo los elementos dentro del Layout.
- **Right:** Alinea a la derecha los elementos dentro del Layout.
- **Left:** Alinea a la izquierda los elementos dentro del Layout.
- **Center_horizontal:** Alinea al centro de forma horizontal los elementos dentro del Layout.
- **Center_vertical:** Alinea al centro de forma vertical los elementos dentro del Layout.

2 LinearLayout

Podemos ocultar y mostrar todos los elementos

Android:visibility

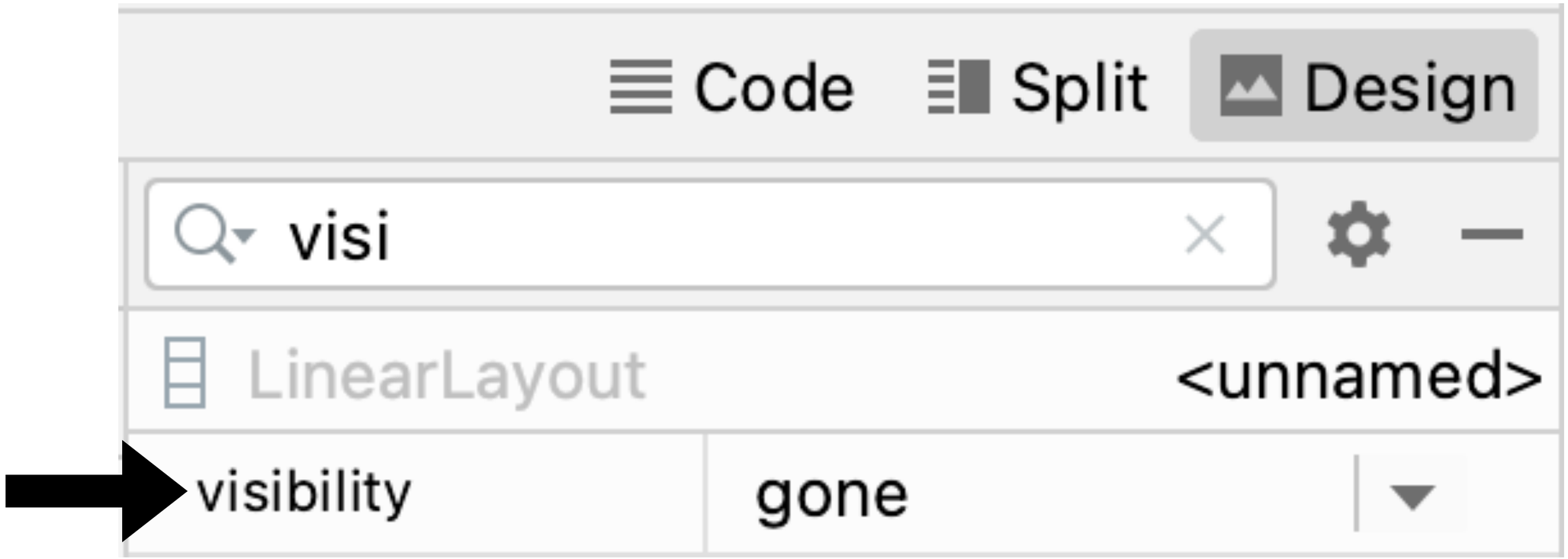


2 LinearLayout

activity_main.xml

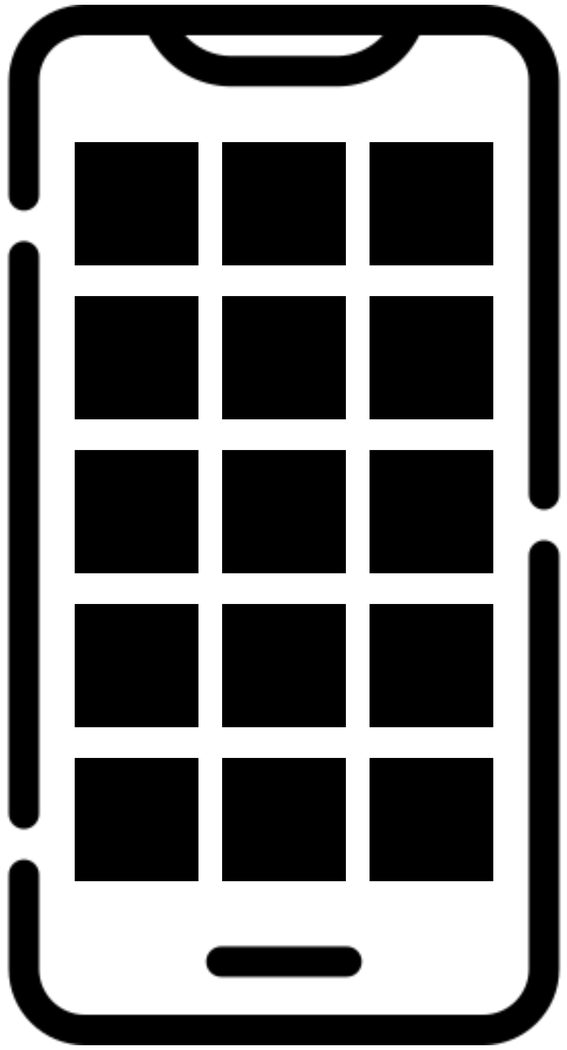
```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:androi
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical"
        android:visibility="gone">
```

2 LinearLayout



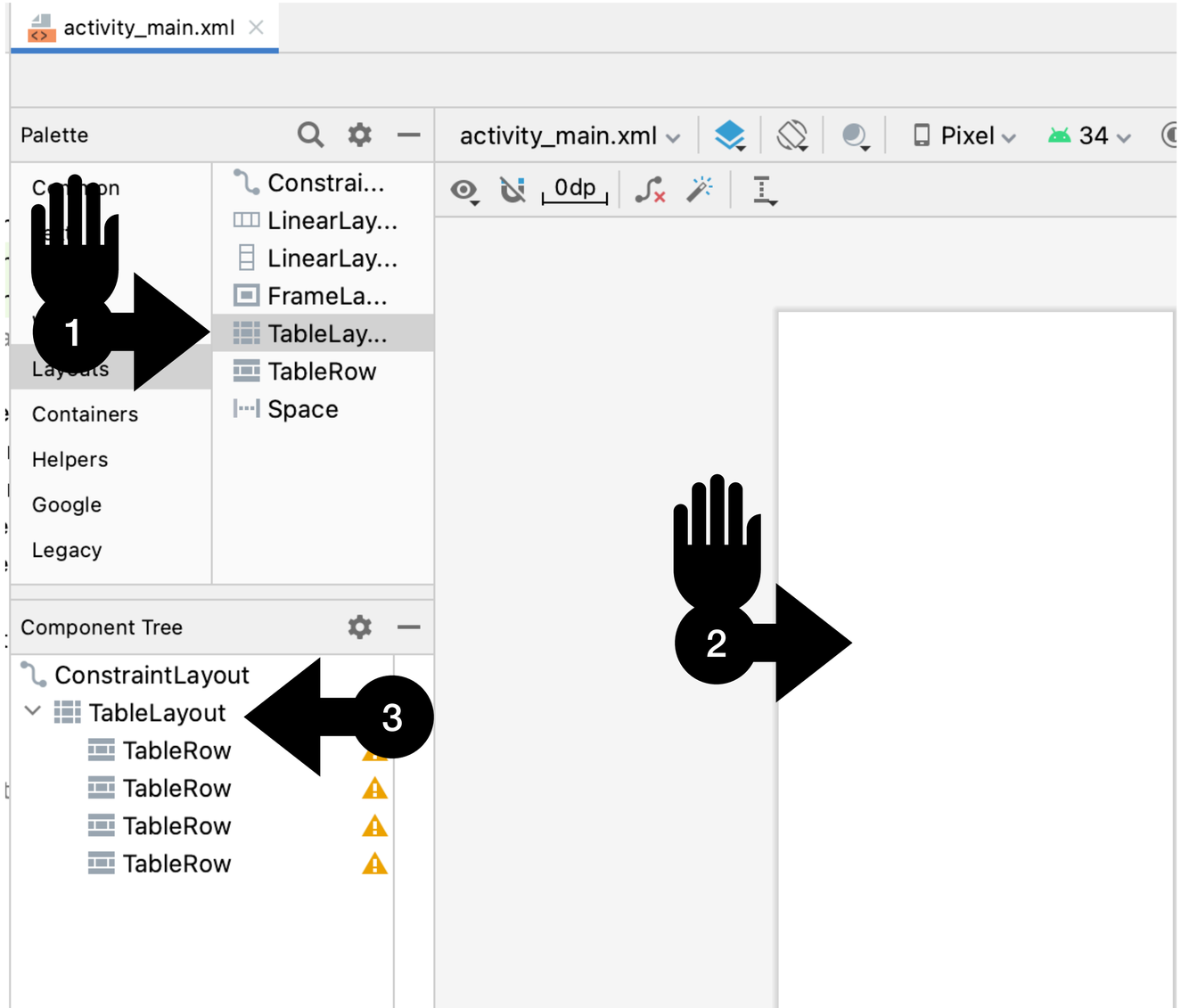
3 TableLayout

Diseño en forma de tabla



Podemos definir el número de filas y columnas

3 TableLayout



3 TableLayout

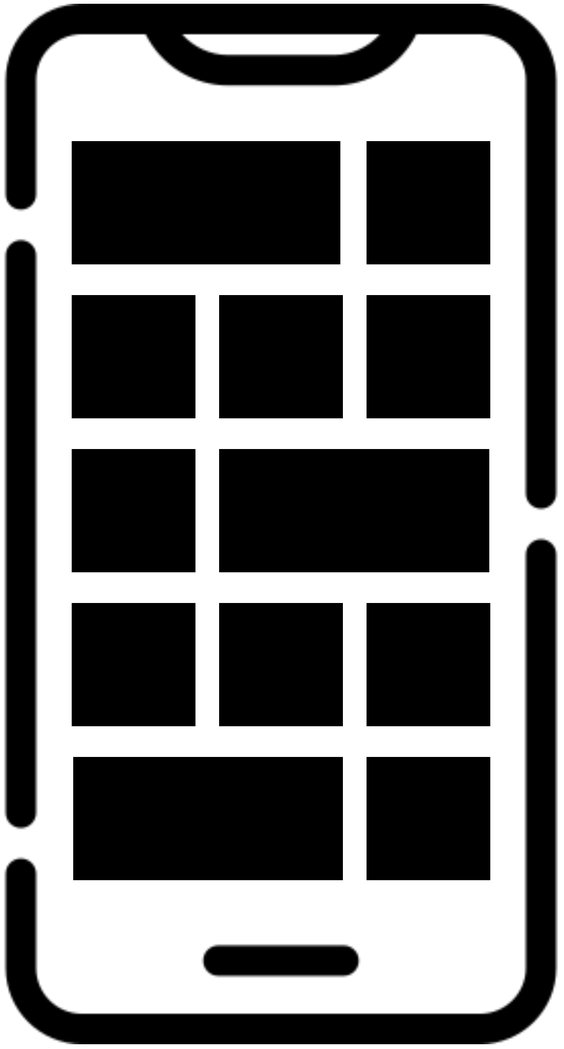
```
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:layout_editor_absoluteX="1dp"
    tools:layout_editor_absoluteY="1dp">

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

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

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

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</TableLayout>
```

Las columnas se crean dinámicamente añadiendo elementos

Número de columnas = La que más columnas tenga

3 TableLayout

Podemos hacer que el contenido de una celda ocupe varias celdas



android:layout_span

Del elemento que hemos añadido

3 TableLayout

Podemos hacer que el contenido de una celda ocupe varias celdas

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

    <Button
        android:id="@+id/button11"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_span="2"
        android:text="Button" />

    <Button
        android:id="@+id/button12"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button" />

    <Button
        android:id="@+id/button13"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button" />
</TableRow>
```

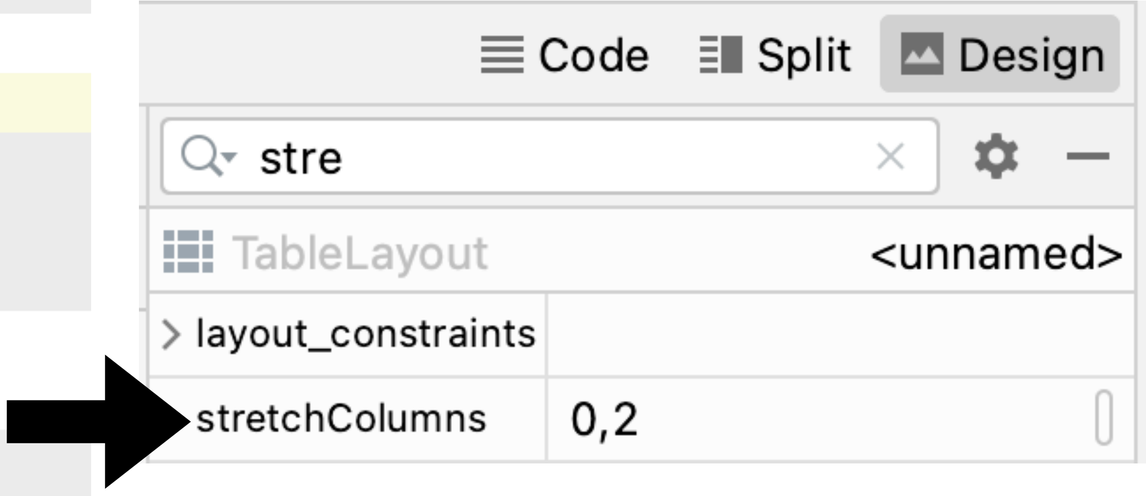
3 TableLayout

El ancho de las columnas → Mayor columna → Pero podemos cambiarlo

- **android:stretchColumns:** Podremos indicar qué columnas se van a poder expandir para ocupar el espacio libre que han dejado las demás columnas a la derecha.
- **android:shrinkColumns:** Nos permitirá indicar qué columnas se van a poder contraer para dejar espacio, y que se puedan salir por la derecha.
- **android:collapseColumns:** Con esta propiedad vamos a poder indicar qué columnas de nuestra tabla queremos ocultar de forma completa.

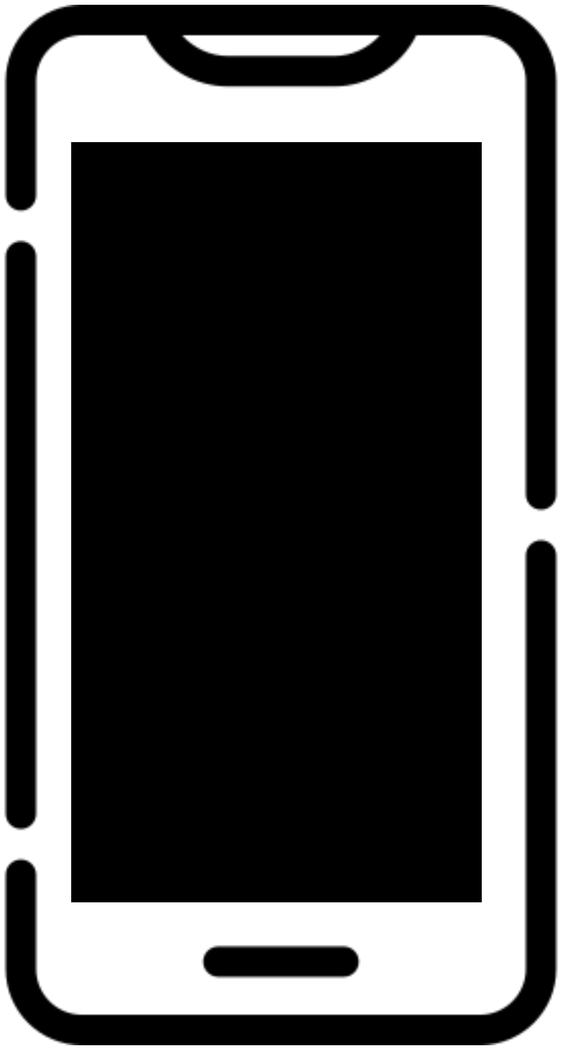
3 TableLayout

```
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:layout_editor_absoluteX="1dp"
    tools:layout_editor_absoluteY="1dp"
    android:stretchColumns="0,2">

    <TableRow
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:gravity="center">

        <Button
            android:id="@+id/button8"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Button" />
```



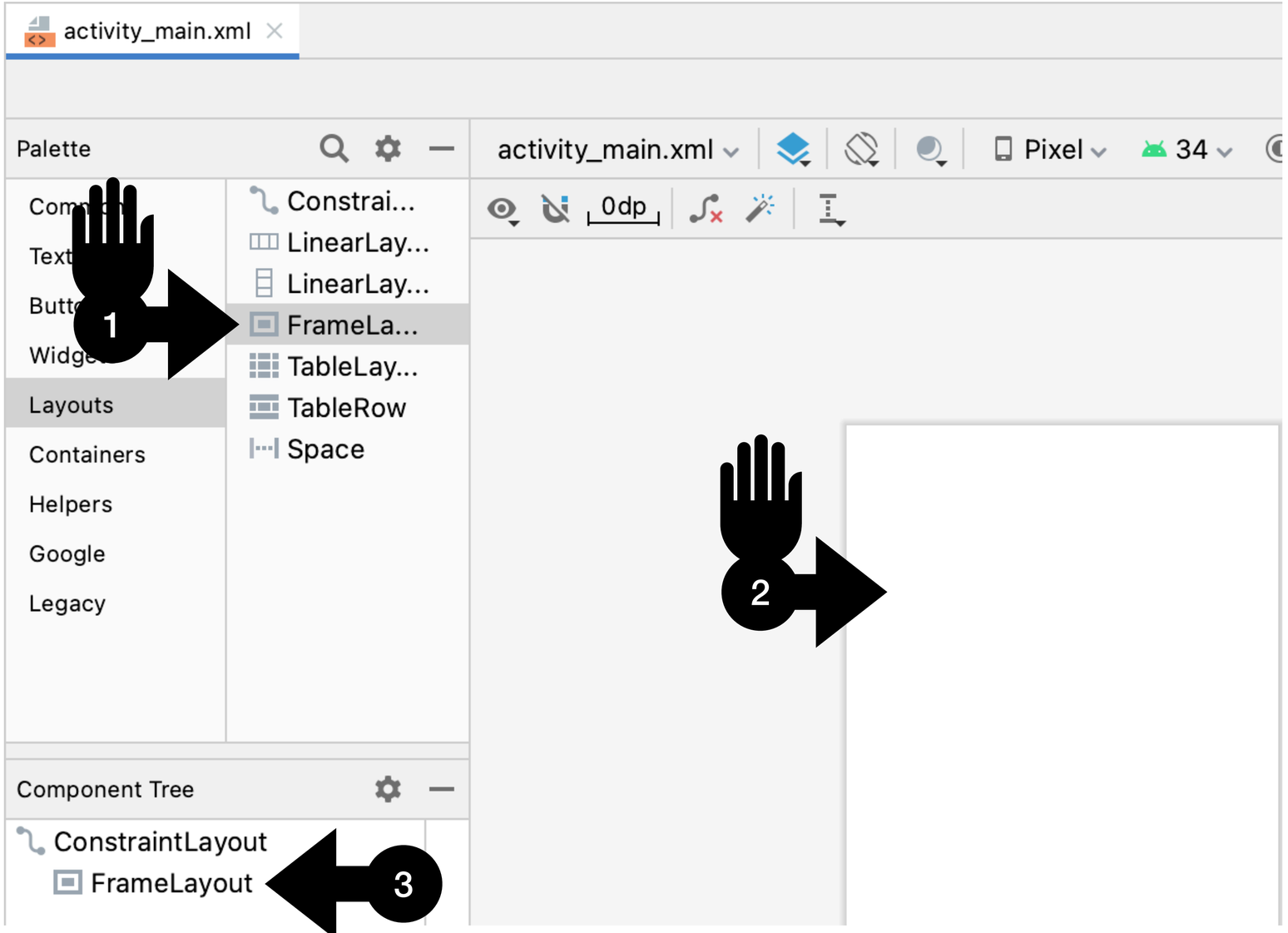
4 FrameLayout

Es un layout muy sencillo que permite poca configuración



Se suele utilizar para colocar un único elemento (vista) en su interior

4 FrameLayout



4 FrameLayout

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://s
    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:background="#FFFFFF"
    tools:context=".MainActivity">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="1dp"
        tools:layout_editor_absoluteY="1dp">

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

Hay que indicar el ancho y el alto

Android:layout_width

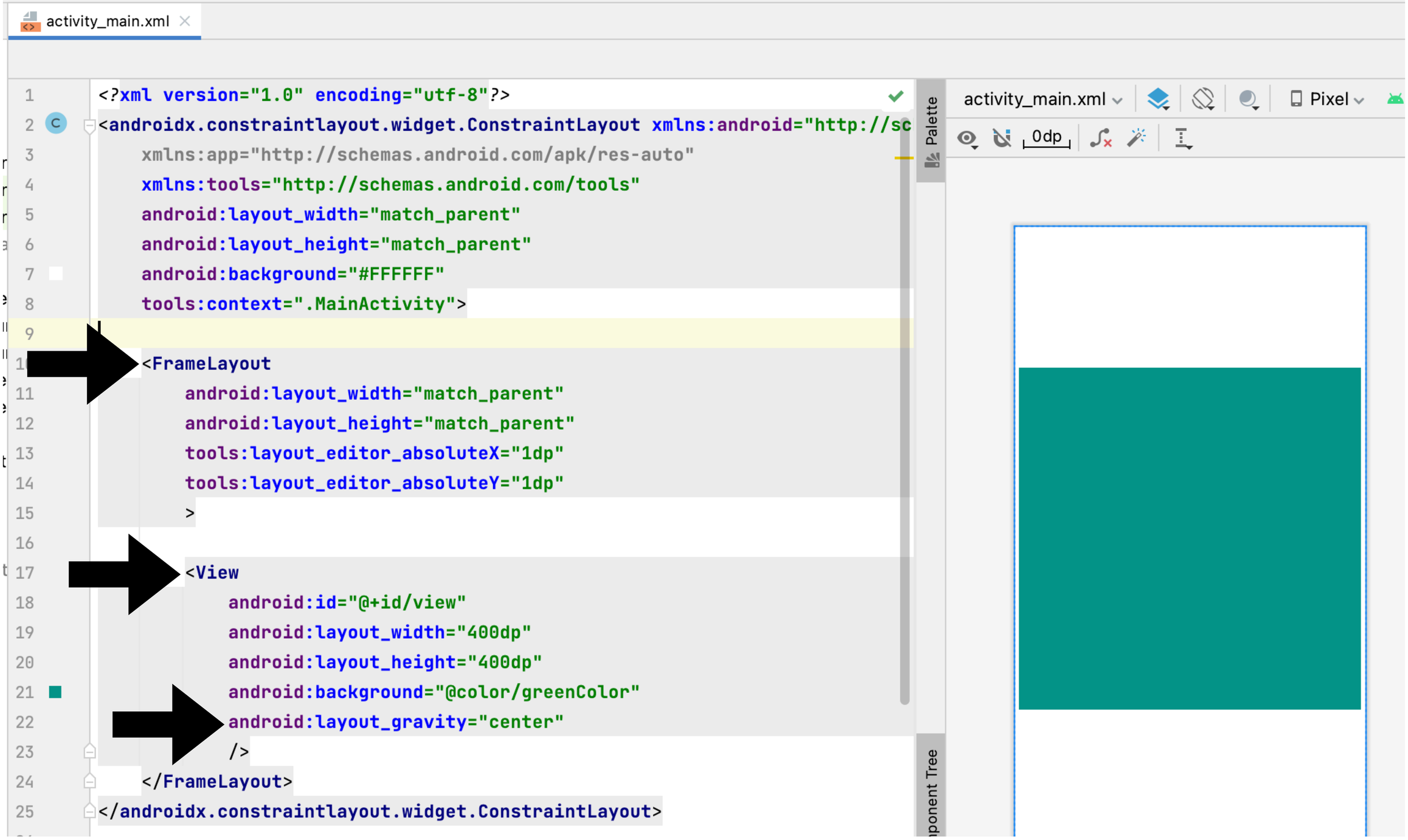
Android:layout_height

4 FrameLayout

Hay que indicar el ancho y el alto

Android:layout_width

Android:layout_height


activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://sc
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:background="#FFFFFF"
tools:context=".MainActivity">
<FrameLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:layout_editor_absoluteX="1dp"
tools:layout_editor_absoluteY="1dp"
>
<View
android:id="@+id/view"
android:layout_width="400dp"
android:layout_height="400dp"
android:background="@color/greenColor"
android:layout_gravity="center"
/>
</FrameLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
Palette
activity_main.xml
Pixel
0dp

5 ConstraintLayout

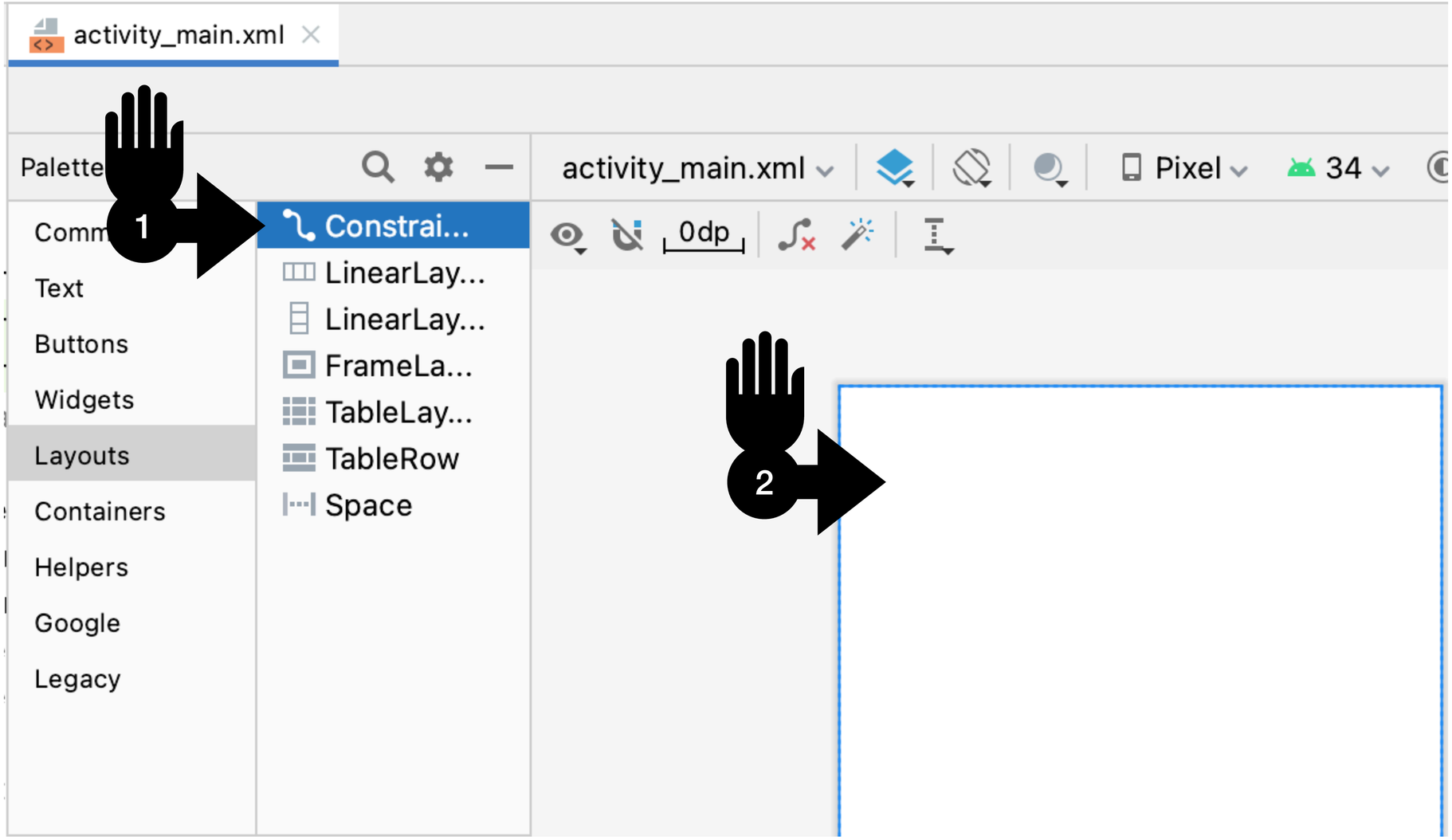
Es el tipo más complejo y que más configuración necesita

Nos permite hacer diseños mucho más sofisticados

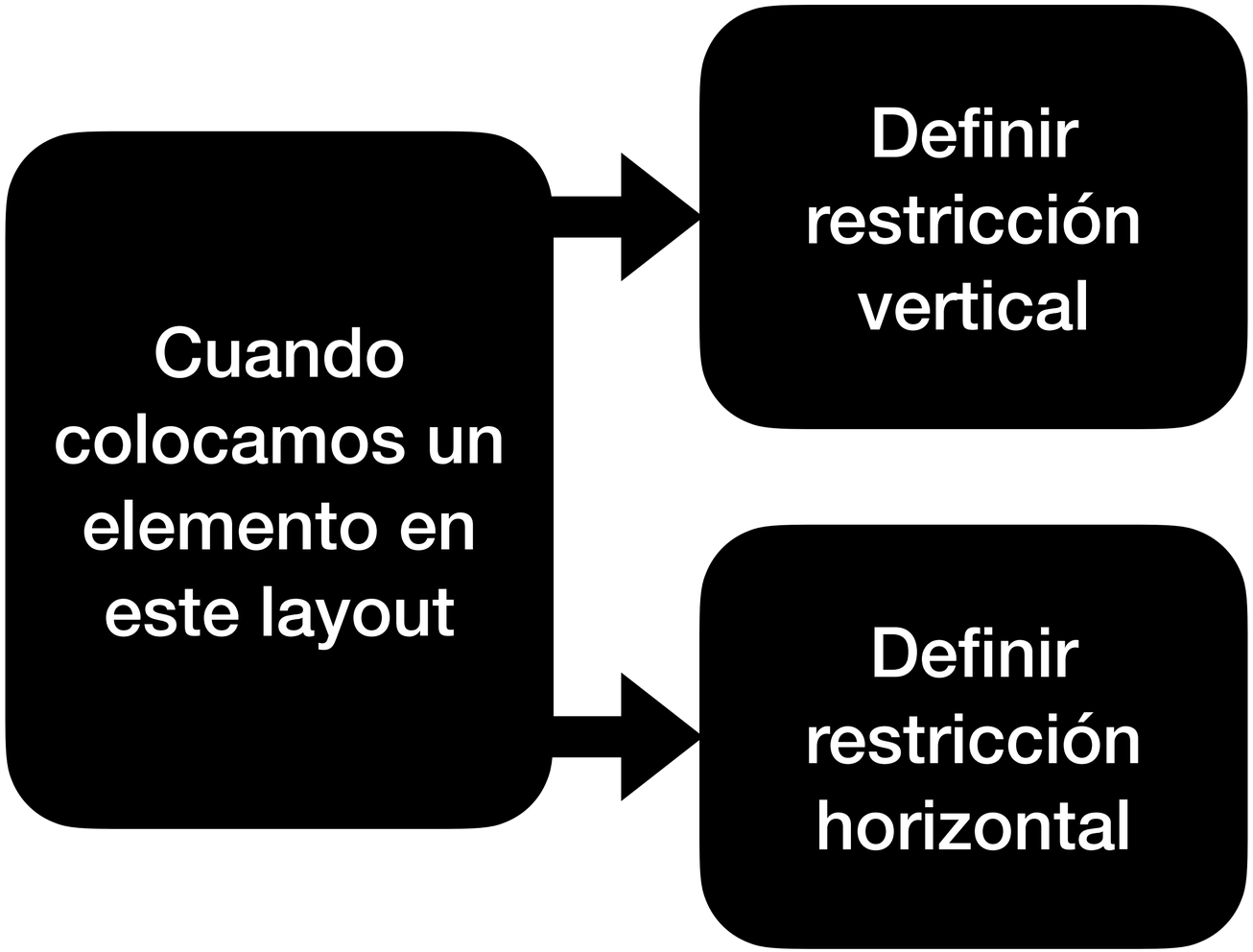
No se recomienda tocar el código generado por su complejidad

Utilizaremos la interfaz de diseño de Android

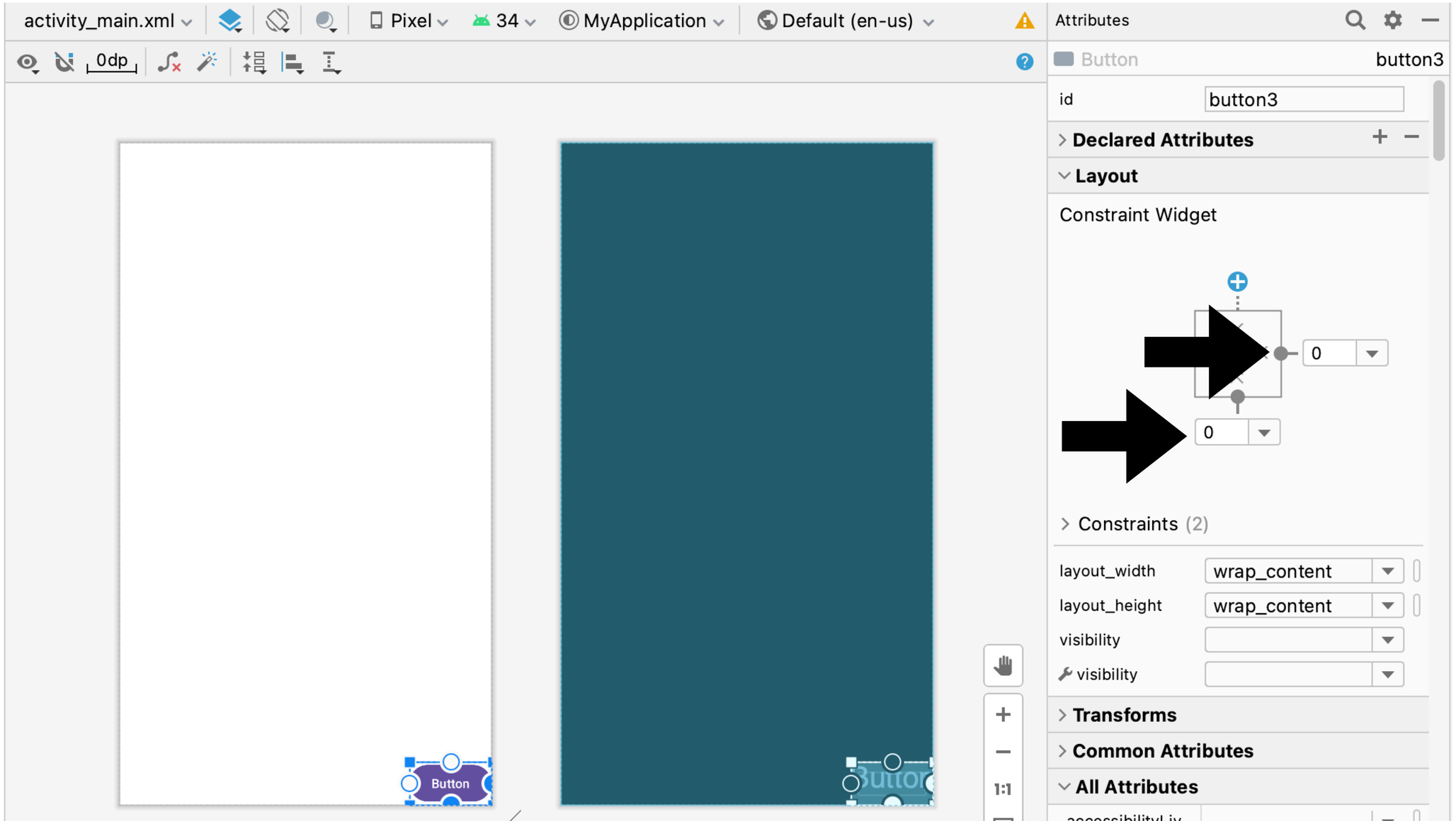
5 ConstraintLayout



5 ConstraintLayout



5 ConstraintLayout



6 Pestañas

No lo vamos a ver por ahora