

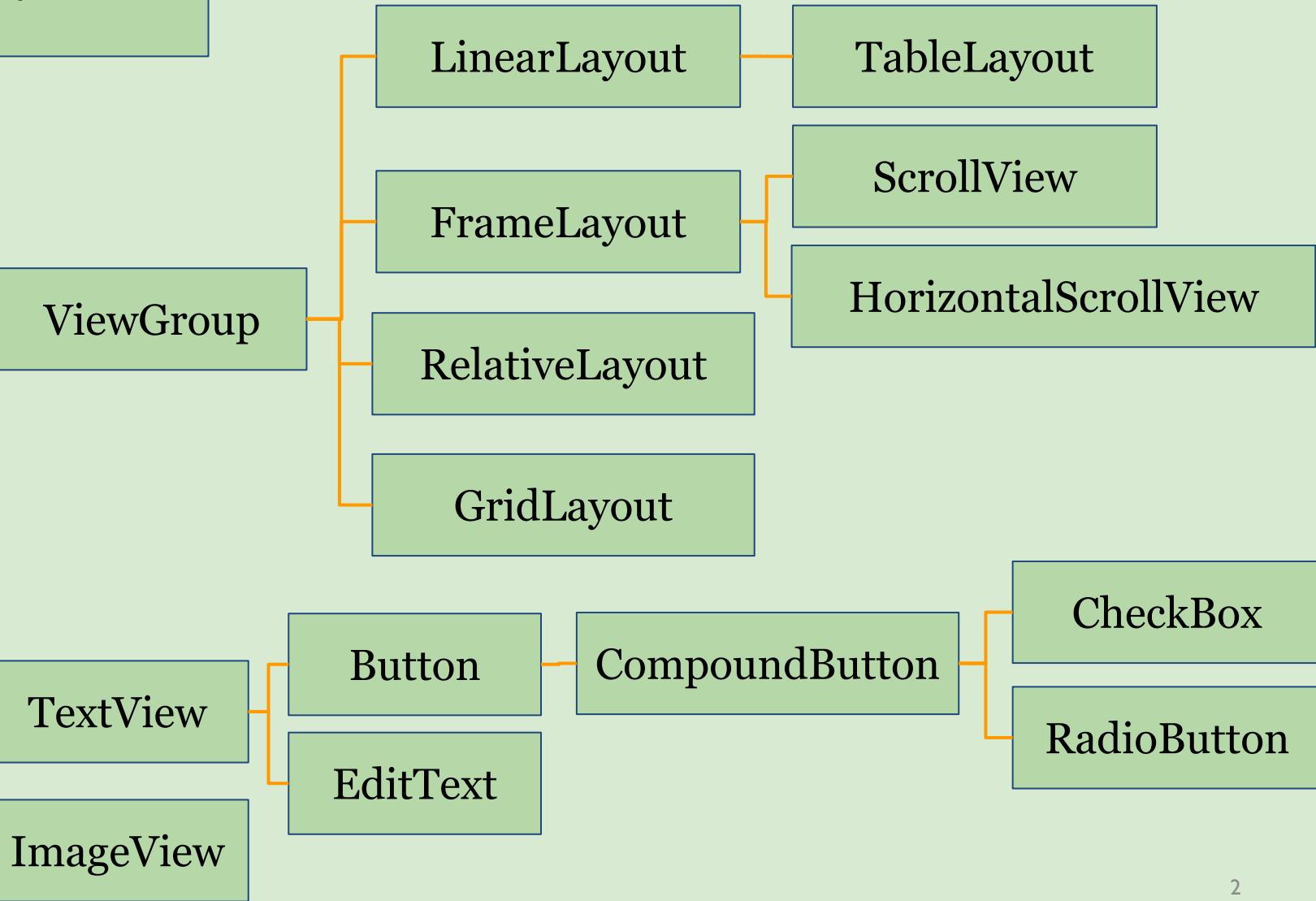
SEMINARIO DE LENGUAJES OPCIÓN ANDROID



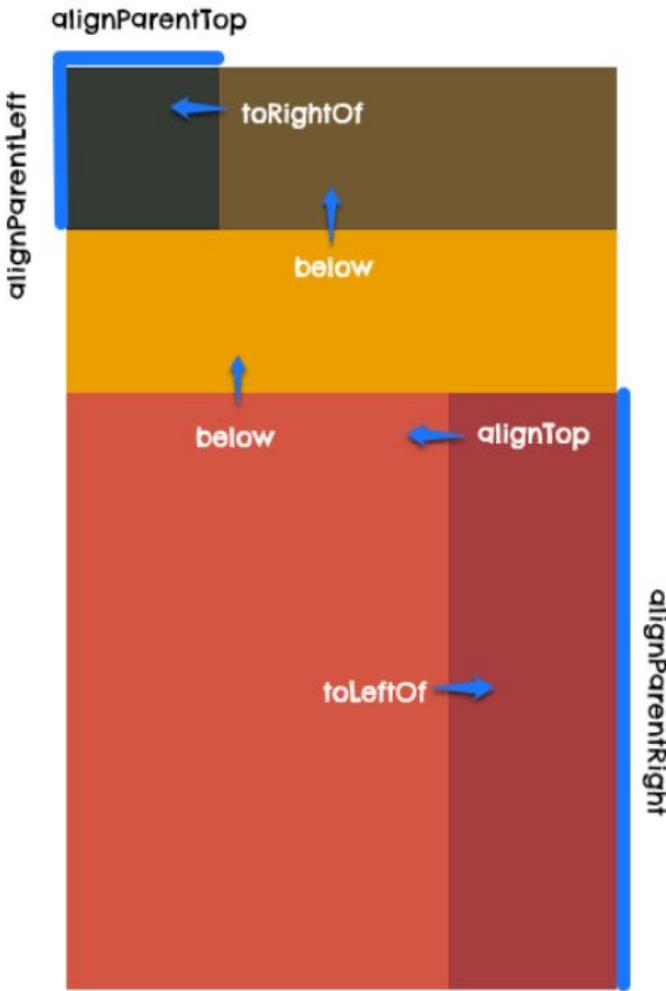
Layouts

Esp. Fernández Sosa Juan Francisco

Algunas Vistas/Views

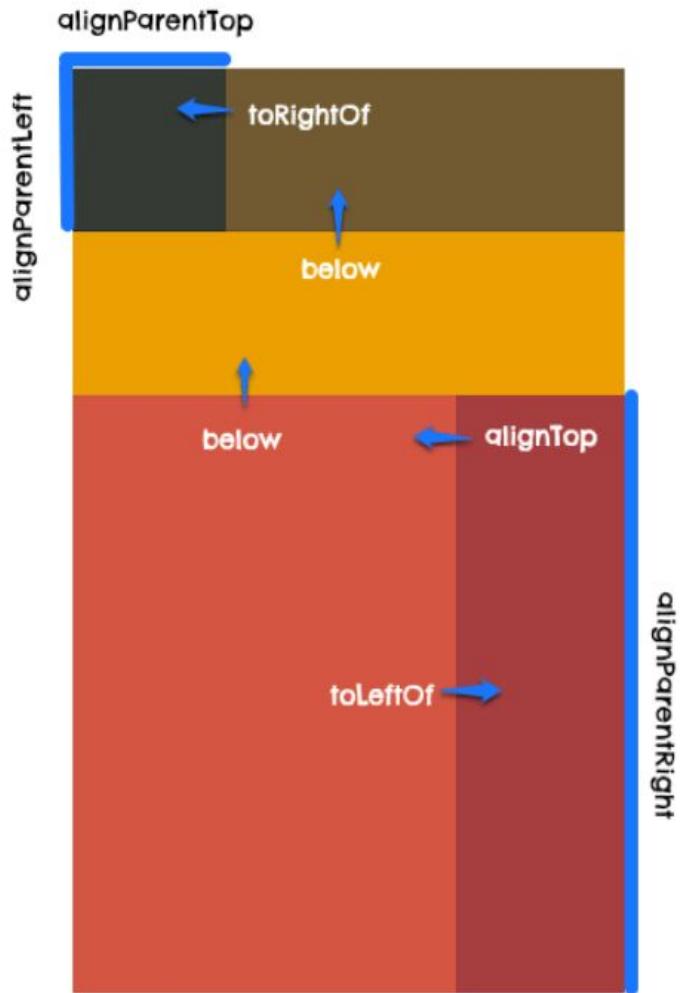


RelativeLayout



Es un **Layout** donde las posiciones de los elementos de un contenedor pueden ser descriptas en relación a otros elementos hijos o bien en relación con el padre, es decir, el propio **Layout**

RelativeLayout



Elimina la necesidad de tener **ViewGroups** anidados, mejorando la performance

RelativeLayout

The screenshot shows the Android Studio interface with two tabs: 'MainActivity.kt' and 'activity_main.xml'. The 'activity_main.xml' tab is active, displaying the XML code for a RelativeLayout. The code defines a button with specific dimensions, text size, and centering properties.

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5     <Button  
6         android:layout_width="100dp"  
7         android:layout_height="100dp"  
8         android:textSize="50dp"  
9         android:layout_centerVertical="true"  
10        android:text="1"  
11    />  
12 </RelativeLayout>
```

Probar el resultado de la siguiente disposición

RelativeLayout

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

- Top Bar:** Shows tabs for "MainActivity.kt" and "activity_main.xml".
- Code Editor:** Displays the XML code for a RelativeLayout. The code includes a button with various attributes like width, height, textSize, and layout_centerVertical set to true. A warning icon is visible in the top right of the editor.
- Palette:** A vertical bar on the right containing icons for different UI components.
- Preview:** A window showing a purple-themed application with a single button labeled "1" centered vertically within its container.
- Bottom Bar:** Standard Android Studio navigation icons.

Text Overlay: A green box on the left side contains the following text:

El atributo
layout_centerVertical
centra verticalmente el
View con respecto a los
límites de su contenedor

RelativeLayout

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

- Toolbar:** Standard Android Studio icons for file operations, navigation, and project management.
- Project Navigators:** Shows the project structure with "Layouts", "app", "src", "main", "res", and "layout" selected, and "activity_main.xml" open.
- Code Editor:** Displays the XML code for "activity_main.xml".

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/
```



```
    <Button
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:textSize="50dp"
        android:layout_centerHorizontal="true"
        android:text="1"/>
```



```
</RelativeLayout>
```
- Bottom Navigation:** Buttons for "Design" and "Text" views, and a status bar indicating "Gradle build finished in 6s 549ms (7 minutes ago)".
- Right Panel:** A large green area containing the text "Probar el resultado de la siguiente disposición".

Probar el resultado de la siguiente disposición

RelativeLayout

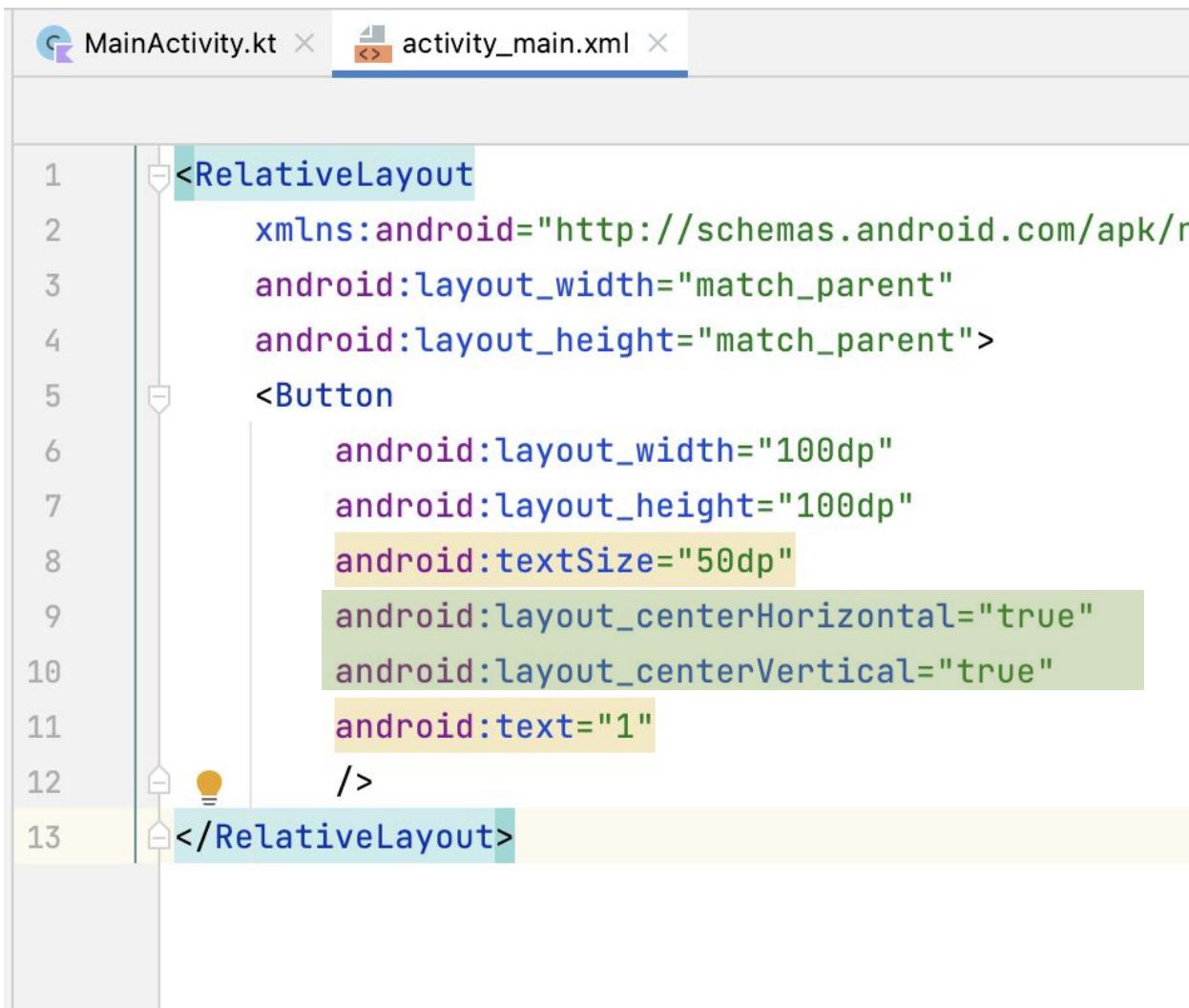
The screenshot shows the Android Studio interface. The code editor on the left displays the XML code for `activity_main.xml`:

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5         <Button  
6             android:layout_width="100dp"  
7             android:layout_height="100dp"  
8             android:textSize="50dp"  
9             android:layout_centerHorizontal="true"  
10            android:text="1"  
11        />  
12    </RelativeLayout>
```

The design preview on the right shows a purple-themed application window titled "My Application". Inside, there is a single button with the number "1" centered horizontally within its container.

**El atributo
`layout_centerHorizontal`
centra horizontalmente el
View con respecto a los límites
de su contenedor**

RelativeLayout

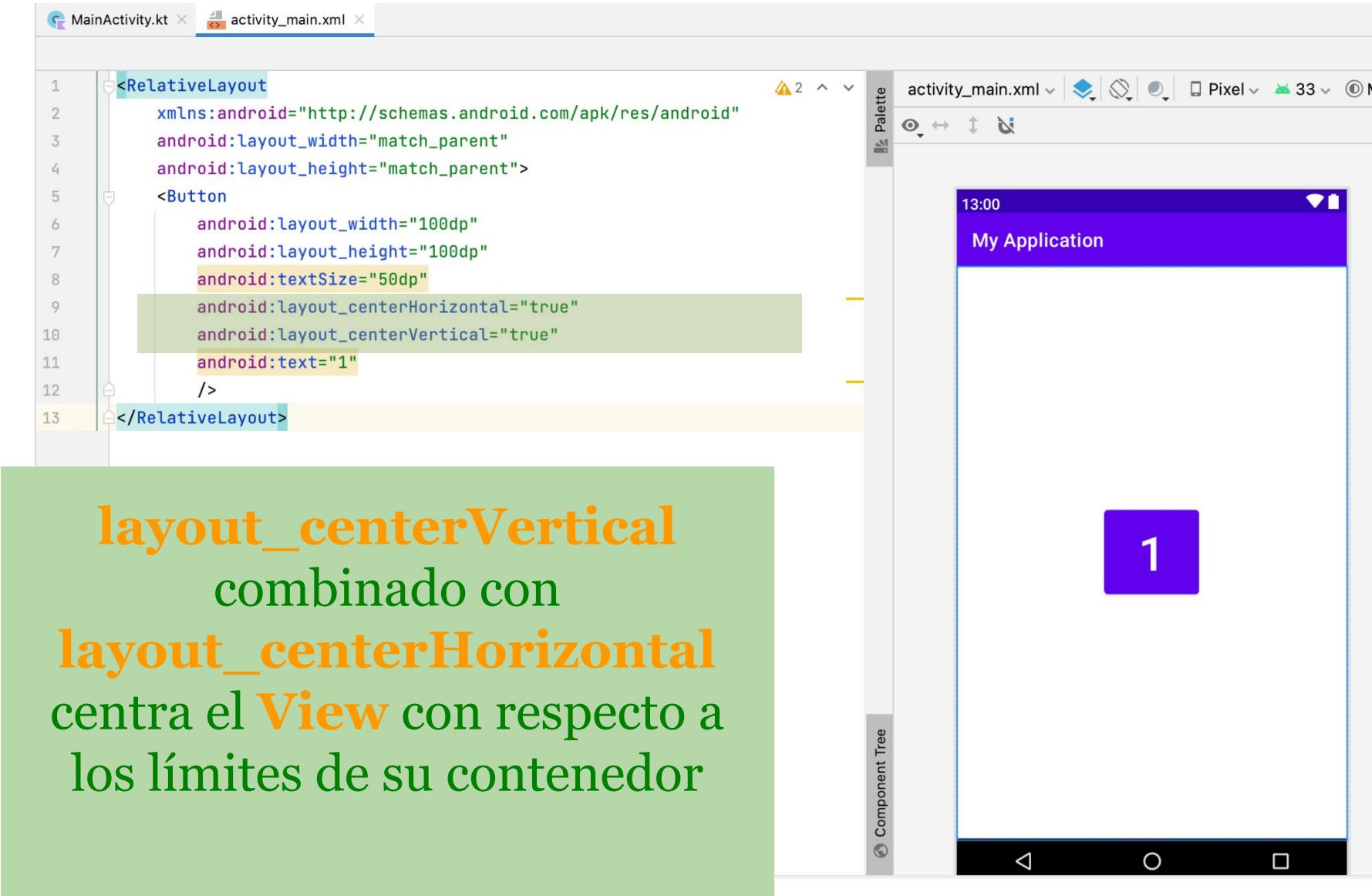


The screenshot shows the Android Studio interface with two tabs: 'MainActivity.kt' and 'activity_main.xml'. The 'activity_main.xml' tab is active, displaying the following XML code:

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/r  
3         android:layout_width="match_parent"  
4         android:layout_height="match_parent">  
5             <Button  
6                 android:layout_width="100dp"  
7                 android:layout_height="100dp"  
8                 android:textSize="50dp"  
9                     android:layout_centerHorizontal="true"  
10                    android:layout_centerVertical="true"  
11                    android:text="1"  
12            />  
13        </RelativeLayout>
```

Probar el resultado de la siguiente disposición

RelativeLayout



The screenshot shows the Android Studio interface with the code editor and design preview side-by-side.

Code Editor (activity_main.xml):

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5     <Button  
6         android:layout_width="100dp"  
7         android:layout_height="100dp"  
8         android:textSize="50dp"  
9         android:layout_centerHorizontal="true"  
10        android:layout_centerVertical="true"  
11        android:text="1"  
12    />  
13 </RelativeLayout>
```

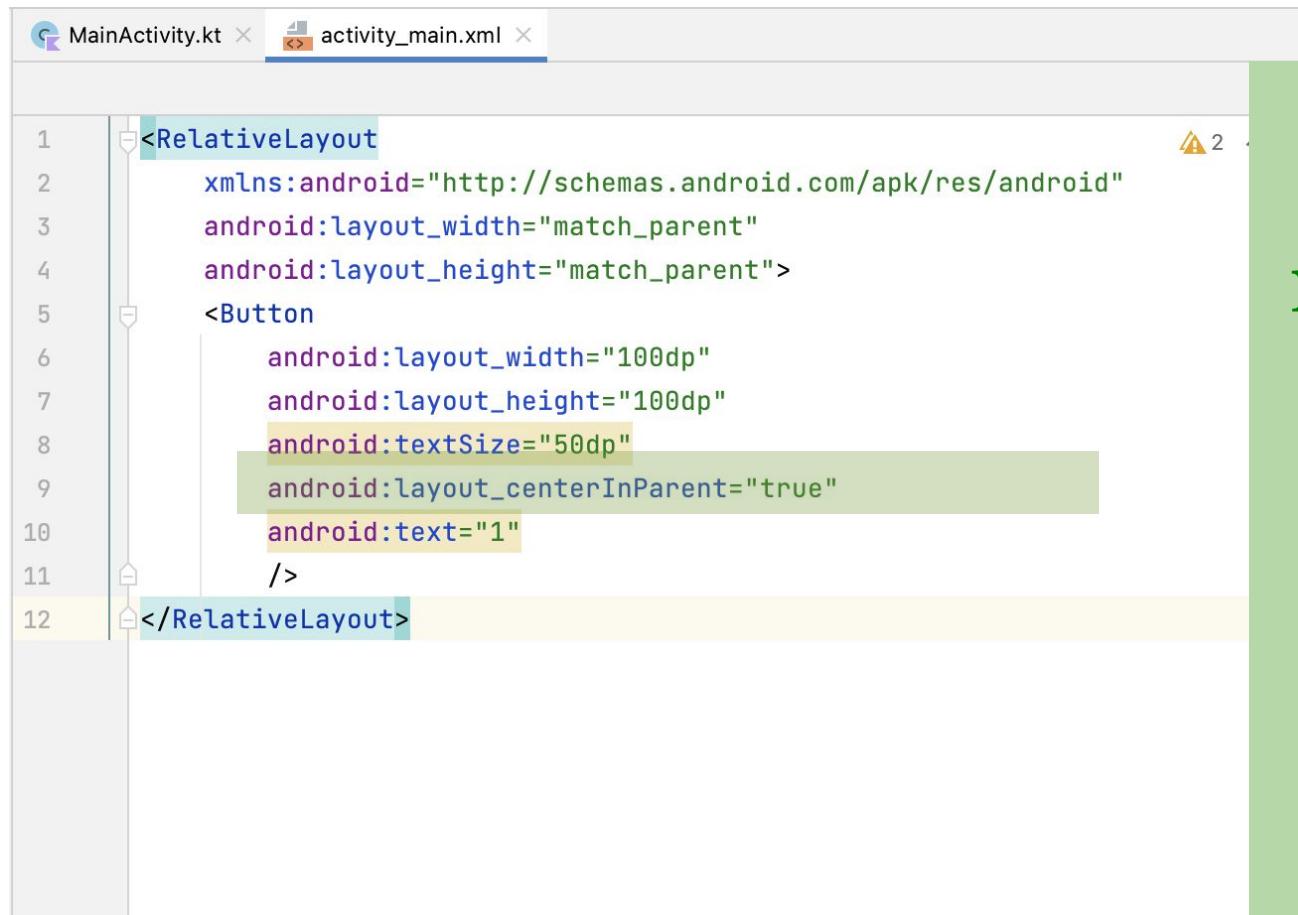
Design Preview:

The preview shows a purple header bar with the text "My Application". Below it is a white content area. In the center of the content area is a blue button with the number "1" on it. The button is centered both horizontally and vertically within its parent container.

Text Overlay (Green Box):

layout_centerVertical
combinado con
layout_centerHorizontal
centra el **View** con respecto a
los límites de su contenedor

RelativeLayout



The screenshot shows the Android Studio interface with the tab 'activity_main.xml' selected. The code editor displays the following XML:

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5     <Button  
6         android:layout_width="100dp"  
7         android:layout_height="100dp"  
8         android:textSize="50dp"  
9         android:layout_centerInParent="true"  
10        android:text="1"  
11    />  
12 </RelativeLayout>
```

The line 'android:layout_centerInParent="true"' is highlighted with a green background, and the line 'android:text="1"' is highlighted with a yellow background.

Probar el resultado de la siguiente disposición

RelativeLayout

The screenshot shows the Android Studio interface. On the left, the code editor displays `MainActivity.kt` and `activity_main.xml`. The XML code defines a `RelativeLayout` containing a `Button` with specific styling and centering attributes. On the right, the design tab shows a preview of the layout on a virtual device with the title "My Application" and a purple status bar. The button is centered within the parent view and has a large number "1" displayed on it.

```
<RelativeLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
    <Button  
        android:layout_width="100dp"  
        android:layout_height="100dp"  
        android:textSize="50dp"  
        android:layout_centerInParent="true"  
        android:text="1"  
    />  
</RelativeLayout>
```

layout_centerInParent:
Centra al **View** con respecto
a los límites de su
contenedor (mismo efecto
que el anterior)

RelativeLayout

The screenshot shows the Android Studio interface with the 'activity_main.xml' tab selected. The code editor displays the following XML configuration:

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5     <Button  
6         android:layout_width="100dp"  
7         android:layout_height="100dp"  
8         android:textSize="50dp"  
9         android:layout_centerInParent="true"  
10        android:layout_alignParentBottom="true"  
11        android:text="1"  
12    />  
13 </RelativeLayout>
```

The code editor includes syntax highlighting and a warning icon in the top right corner. The bottom right corner features a green sidebar with the text "Probar el resultado de la siguiente disposición".

Probar el resultado de la siguiente disposición

RelativeLayout

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

- MainActivity.kt**: An open Java file tab.
- activity_main.xml**: The currently selected XML layout file tab.
- Code Editor**: Displays the XML code for a RelativeLayout containing a single Button.

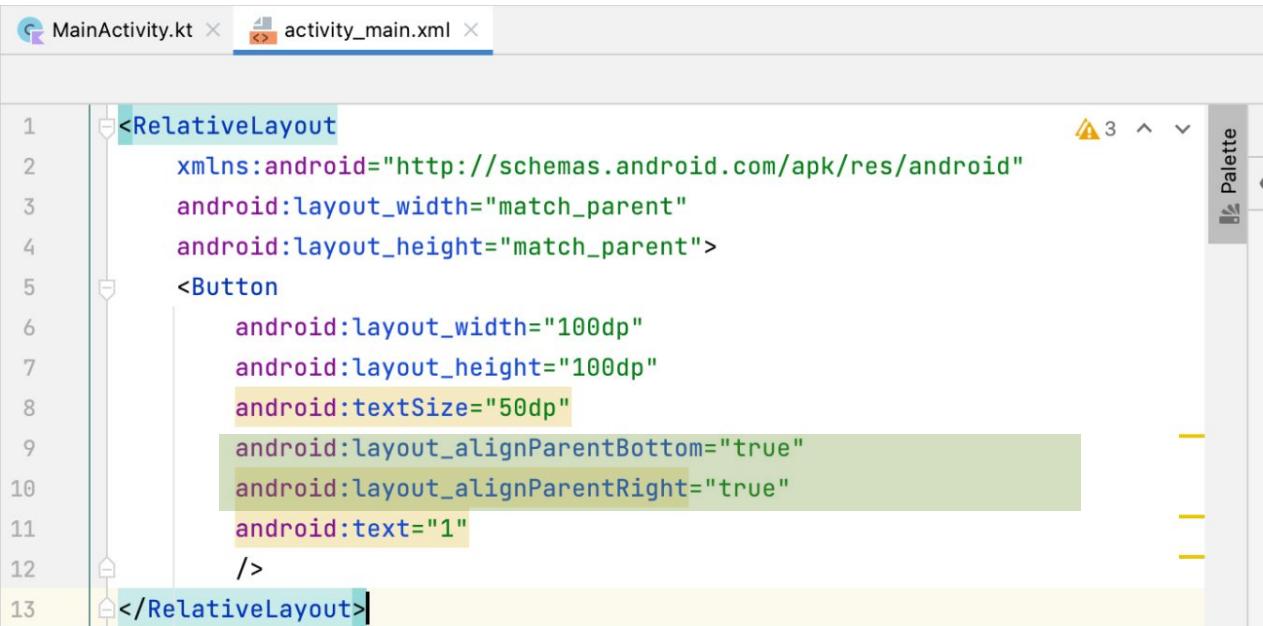
```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:textSize="50dp"
        android:layout_centerInParent="true"
        android:layout_alignParentBottom="true"
        android:text="1"/>
</RelativeLayout>
```
- Palette**: A panel on the right containing various UI component icons.
- Preview**: A visual representation of the layout on a virtual device screen titled "My Application". The screen shows a purple header bar with the time "13:00" and a white content area containing a centered button with the number "1".
- Component Tree**: A vertical tree view on the far right showing the hierarchy of the layout elements.
- Bottom Bar**: The standard Android Studio navigation bar with icons for back, forward, search, and others.

Text Overlay (Green Box):

layout_alignParentBottom: Alinea el borde inferior del View con el borde inferior del contenedor

Page Number: 14

RelativeLayout



The screenshot shows the Android Studio interface with the tab bar at the top showing 'MainActivity.kt' and 'activity_main.xml'. The code editor displays the following XML code:

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5     <Button  
6         android:layout_width="100dp"  
7         android:layout_height="100dp"  
8         android:textSize="50dp"  
9         android:layout_alignParentBottom="true"  
10        android:layout_alignParentRight="true"  
11        android:text="1"  
12    />  
13 </RelativeLayout>
```

The code is color-coded, and several attributes in the Button's definition are highlighted with a light green background: 'layout_alignParentBottom', 'layout_alignParentRight', and 'text'. There are three yellow warning icons in the top right corner of the code editor.

Probar el resultado de la siguiente disposición

RelativeLayout

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

- Top Bar:** Shows tabs for "MainActivity.kt" and "activity_main.xml".
- Code Editor (activity_main.xml):** Displays the XML code for a RelativeLayout. A green highlight covers the area from line 9 to line 11, specifically the attributes: `android:layout_alignParentBottom="true"`, `android:layout_alignParentRight="true"`, and `android:text="1"`.

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:textSize="50dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:text="1"/>
</RelativeLayout>
```
- Preview Window:** Shows a purple-themed application window titled "My Application". Inside, there is a white button with the text "1" at its bottom-right corner.
- Bottom Left Text Box:** A green box contains the text:

layout_alignParentRight:
Alinea el borde derecho
del **View** con el borde
derecho de su
contenedor
- Bottom Right Page Number:** A purple box in the bottom right corner contains the number "16".

RelativeLayout



The screenshot shows the Android Studio interface with the code editor open. The file is `activity_main.xml`. The code defines a `RelativeLayout` with a single child, a `Button`. The `Button` has the following attributes:

- `android:layout_width="100dp"`
- `android:layout_height="100dp"`
- `android:textSize="50dp"`
- `android:layout_alignParentBottom="true"`
- `android:layout_alignParentRight="true"`
- `android:layout_alignParentLeft="true"`
- `android:text="1"`

The code editor has syntax highlighting and a status bar at the top indicating 5 warnings.

Probar el
resultado de la
siguiente
disposición

RelativeLayout

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

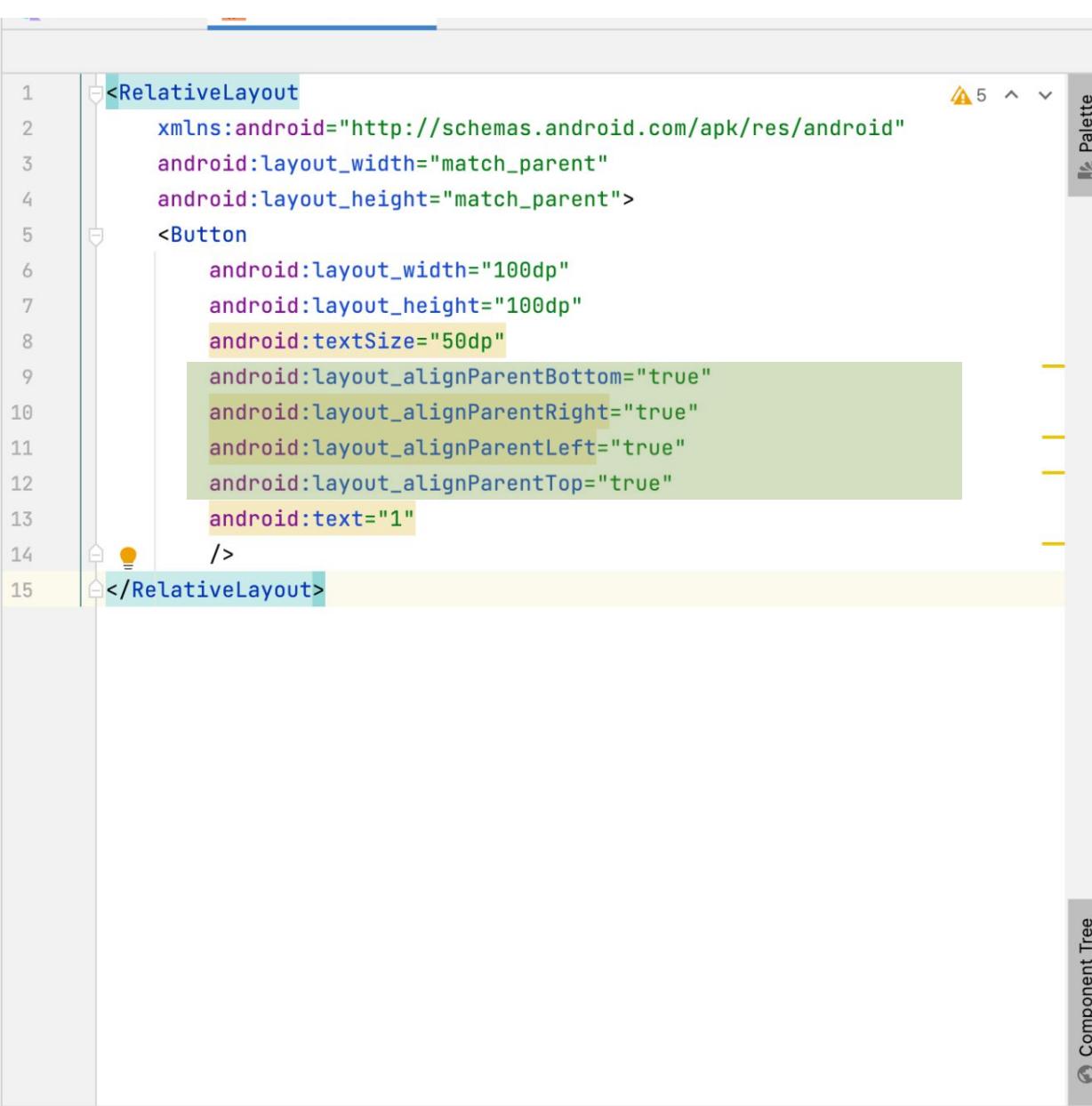
- Top Bar:** Shows tabs for "MainActivity.kt" and "activity_main.xml".
- Code Editor (Left):** Displays the XML code for a RelativeLayout. The code includes a Button with specific styling and alignment attributes.
- Preview (Right):** Shows a mobile device screen with a purple header bar displaying "13:00" and "My Application". The main content area contains a white button with the number "1" centered on it.
- Toolbars and Palettes:** Standard Android Studio toolbars and palettes are visible along the top and right edges.

Text Overlay (Bottom Left):

layout_alignParentLeft:
Alinea el borde izquierdo
del **View** con el borde
izquierdo de su
contenedor

La mayoría de los atributos que posicionan los **Views** dentro de un **RelativeLayout** afectan a uno de los bordes del **View**, es por eso que al combinarlos puede cambiar la dimensión de este elemento.

RelativeLayout



The screenshot shows the Android Studio XML layout editor. A RelativeLayout is defined with a single child, a Button. The Button has its width and height set to 100dp, and its text size is set to 50dp. It also has five alignment attributes: layout_alignParentBottom, layout_alignParentRight, layout_alignParentLeft, and layout_alignParentTop, all set to true. The text of the button is set to "1". The code is as follows:

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5     <Button  
6         android:layout_width="100dp"  
7         android:layout_height="100dp"  
8         android:textSize="50dp"  
9         android:layout_alignParentBottom="true"  
10        android:layout_alignParentRight="true"  
11        android:layout_alignParentLeft="true"  
12        android:layout_alignParentTop="true"  
13        android:text="1"  
14    />  
15 </RelativeLayout>
```

Probar el resultado de la siguiente disposición

RelativeLayout

The screenshot shows the Android Studio interface with the XML code for a RelativeLayout and its preview in the design tab.

XML Code:

```
1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5     <Button
6         android:layout_width="100dp"
7         android:layout_height="100dp"
8         android:textSize="50dp"
9         android:layout_alignParentBottom="true"
10        android:layout_alignParentRight="true"
11        android:layout_alignParentLeft="true"
12        android:layout_alignParentTop="true"
13        android:text="1"
14    />
15 </RelativeLayout>
```

Design Preview:

The preview shows a purple screen titled "My Application" with the time "13:00". A single button is displayed in the center. The button has a white background, a black border, and the number "1" in black text. It is aligned at the top edge of its parent container.

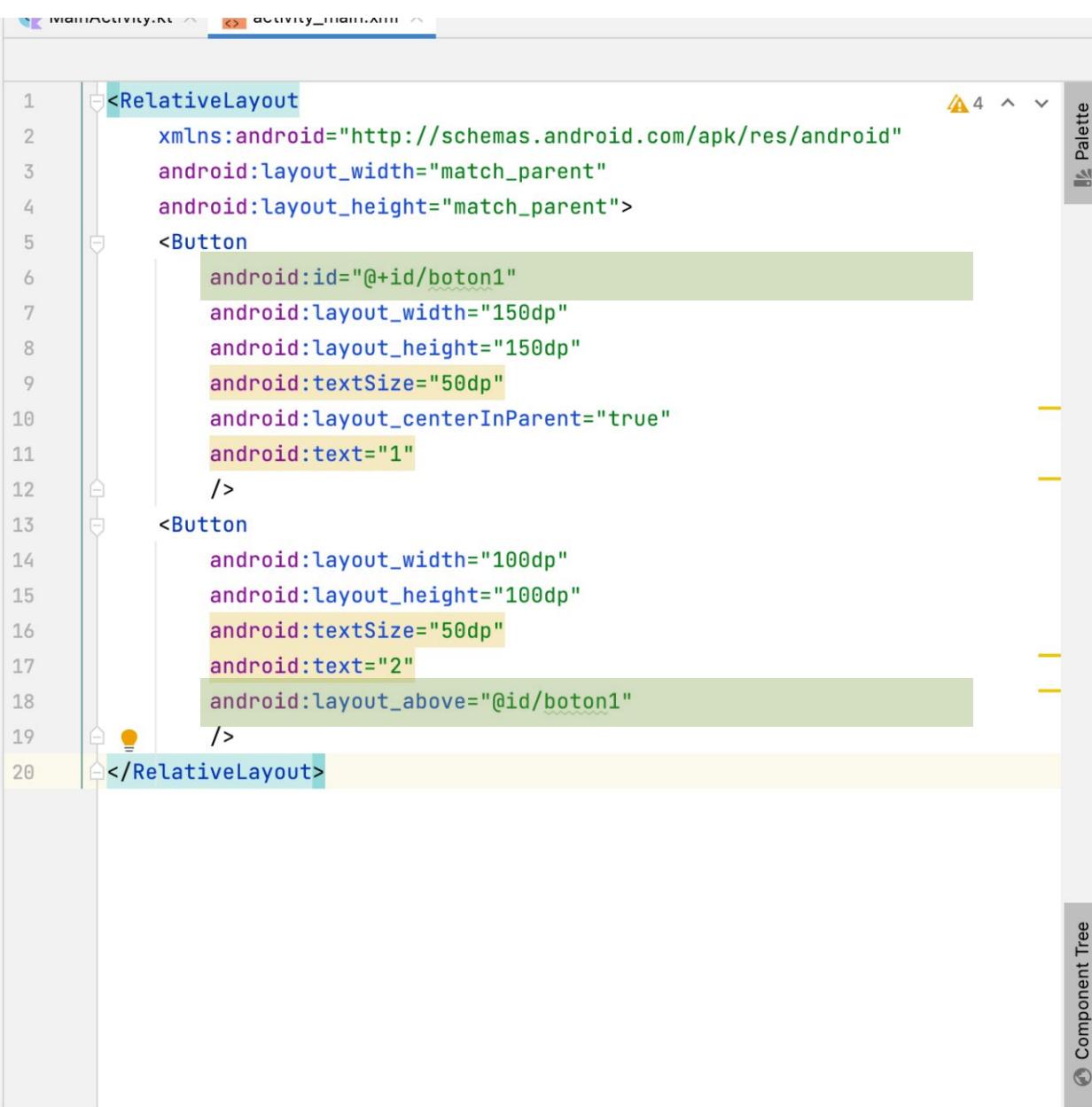
Text Overlay:

layout_alignParentTop:
Alinea el borde superior
del **View** con el borde
superior de su
contenedor

1

21

RelativeLayout



The screenshot shows the Android Studio interface with the XML code for a RelativeLayout. The code defines two buttons: one with a width of 150dp and height of 150dp, centered in its parent; and another with a width of 100dp and height of 100dp, positioned above the first button. Both buttons have a text size of 50dp. The XML code is as follows:

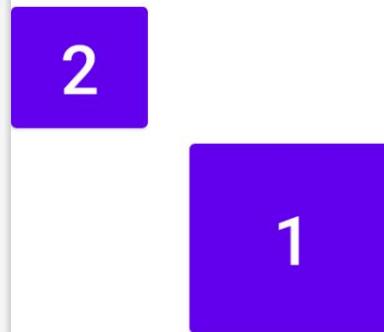
```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5     <Button  
6         android:id="@+id/boton1"  
7         android:layout_width="150dp"  
8         android:layout_height="150dp"  
9         android:textSize="50dp"  
10        android:layout_centerInParent="true"  
11        android:text="1"  
12    />  
13    <Button  
14        android:layout_width="100dp"  
15        android:layout_height="100dp"  
16        android:textSize="50dp"  
17        android:text="2"  
18        android:layout_above="@+id/boton1"  
19    />  
20 </RelativeLayout>
```

Probar el resultado de la siguiente disposición

RelativeLayout

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5     <Button  
6         android:id="@+id/boton1"  
7         android:layout_width="150dp"  
8         android:layout_height="150dp"  
9         android:textSize="50dp"  
10        android:layout_centerInParent="true"  
11        android:text="1"  
12    />  
13    <Button  
14        android:layout_width="100dp"  
15        android:layout_height="100dp"  
16        android:textSize="50dp"  
17        android:text="2"  
18        android:layout_above="@+id/boton1"  
19    />  
20 </RelativeLayout>
```

layout_above: Alinea el borde inferior del View con el borde superior de otro View



RelativeLayout

```
1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5     <Button
6         android:id="@+id/boton1"
7         android:layout_width="150dp"
8         android:layout_height="150dp"
9         android:textSize="50dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12    />
13    <Button
14        android:layout_width="100dp"
15        android:layout_height="100dp"
16        android:textSize="50dp"
17        android:text="2"
18        android:layout_alignTop="@+id/boton1"
19    />
20 </RelativeLayout>
```

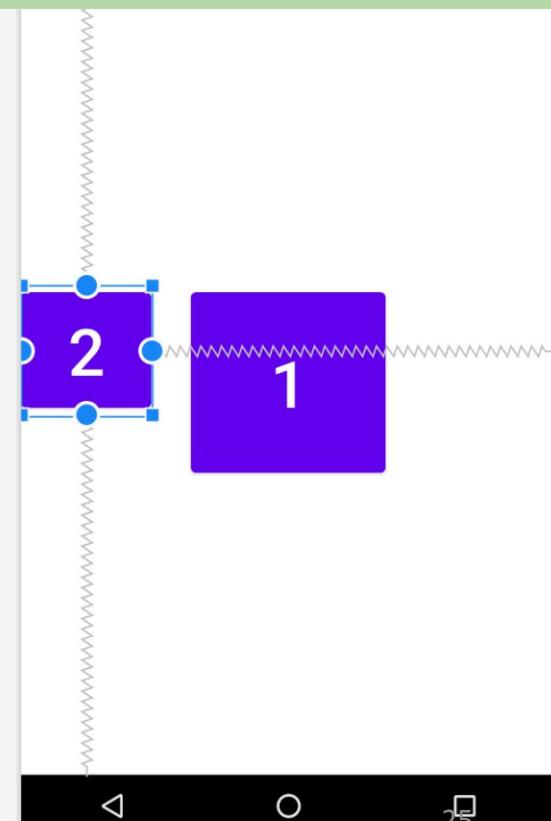
Probar el resultado de la siguiente disposición

RelativeLayout

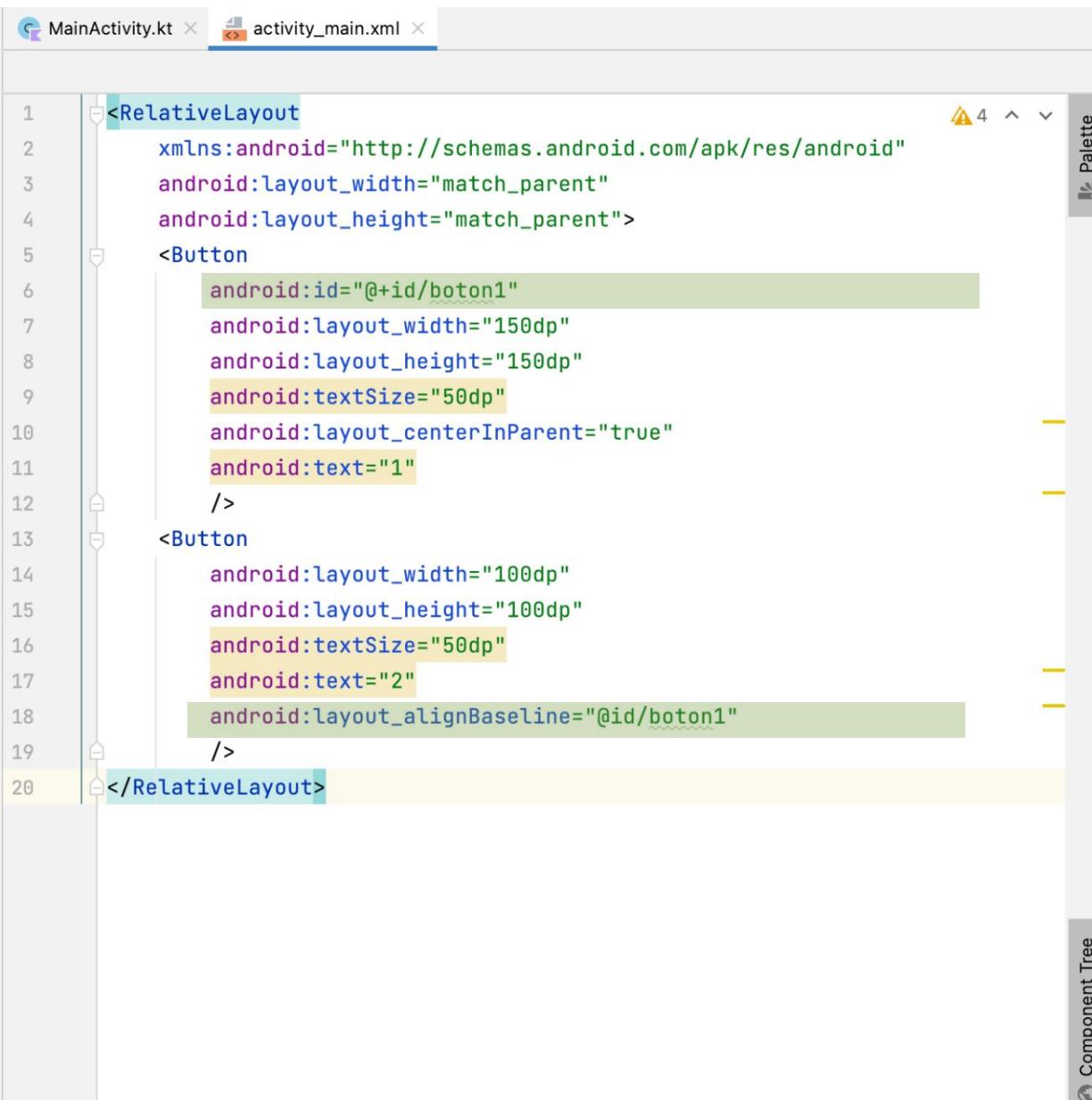
```
MainActivity.kt x activity_main.xml x

1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5     <Button
6         android:id="@+id/boton1"
7         android:layout_width="150dp"
8         android:layout_height="150dp"
9         android:textSize="50dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12    />
13    <Button
14        android:layout_width="100dp"
15        android:layout_height="100dp"
16        android:textSize="50dp"
17        android:text="2"
18        android:layout_alignTop="@+id/boton1"
19    />
20 </RelativeLayout>
```

layout_alignTop: Alinea el borde superior del **View** con el borde superior de otro **View**



RelativeLayout



The screenshot shows the Android Studio interface with the file `activity_main.xml` selected in the tab bar. The code editor displays the following XML configuration:

```
1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5     <Button
6         android:id="@+id/boton1"
7         android:layout_width="150dp"
8         android:layout_height="150dp"
9         android:textSize="50dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12    />
13    <Button
14        android:layout_width="100dp"
15        android:layout_height="100dp"
16        android:textSize="50dp"
17        android:text="2"
18        android:layout_alignBaseline="@+id/boton1"
19    />
20 </RelativeLayout>
```

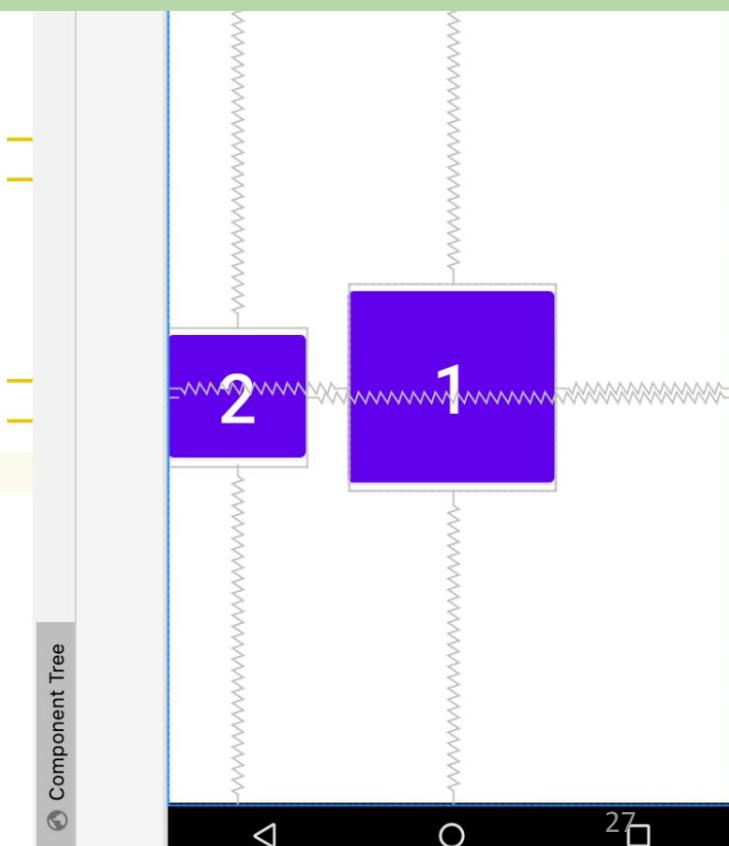
The code defines a `RelativeLayout` with two child `Button` elements. The first button has an ID of `@+id/boton1`, a width of 150dp, a height of 150dp, and a text size of 50dp. It is centered in its parent. The second button has a width of 100dp, a height of 100dp, and a text size of 50dp. Its text is set to "2". The `android:layout_alignBaseline` attribute of the second button is set to the baseline of the first button, which means they will be aligned at the same vertical level.

Probar el resultado de la siguiente disposición

RelativeLayout

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5  
6     <Button  
7         android:id="@+id/boton1"  
8         android:layout_width="150dp"  
9         android:layout_height="150dp"  
10        android:layout_centerInParent="true"  
11        android:text="1"  
12        android:textSize="50dp" />  
13  
14    <Button  
15        android:layout_width="100dp"  
16        android:layout_height="100dp"  
17        android:text="2"  
18        android:textSize="50dp"  
19        android:layout_alignBaseline="@+id/boton1"  
20    />  
21 </RelativeLayout>
```

layout_alignBaseline:
Alinea la línea base de un
View con la línea base de
otro **View**



RelativeLayout

```
1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8         android:layout_width="150dp"
9         android:layout_height="150dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12        android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_alignBottom="@+id/boton1"
20
21     />
22 </RelativeLayout>
```

Probar el resultado de la siguiente disposición

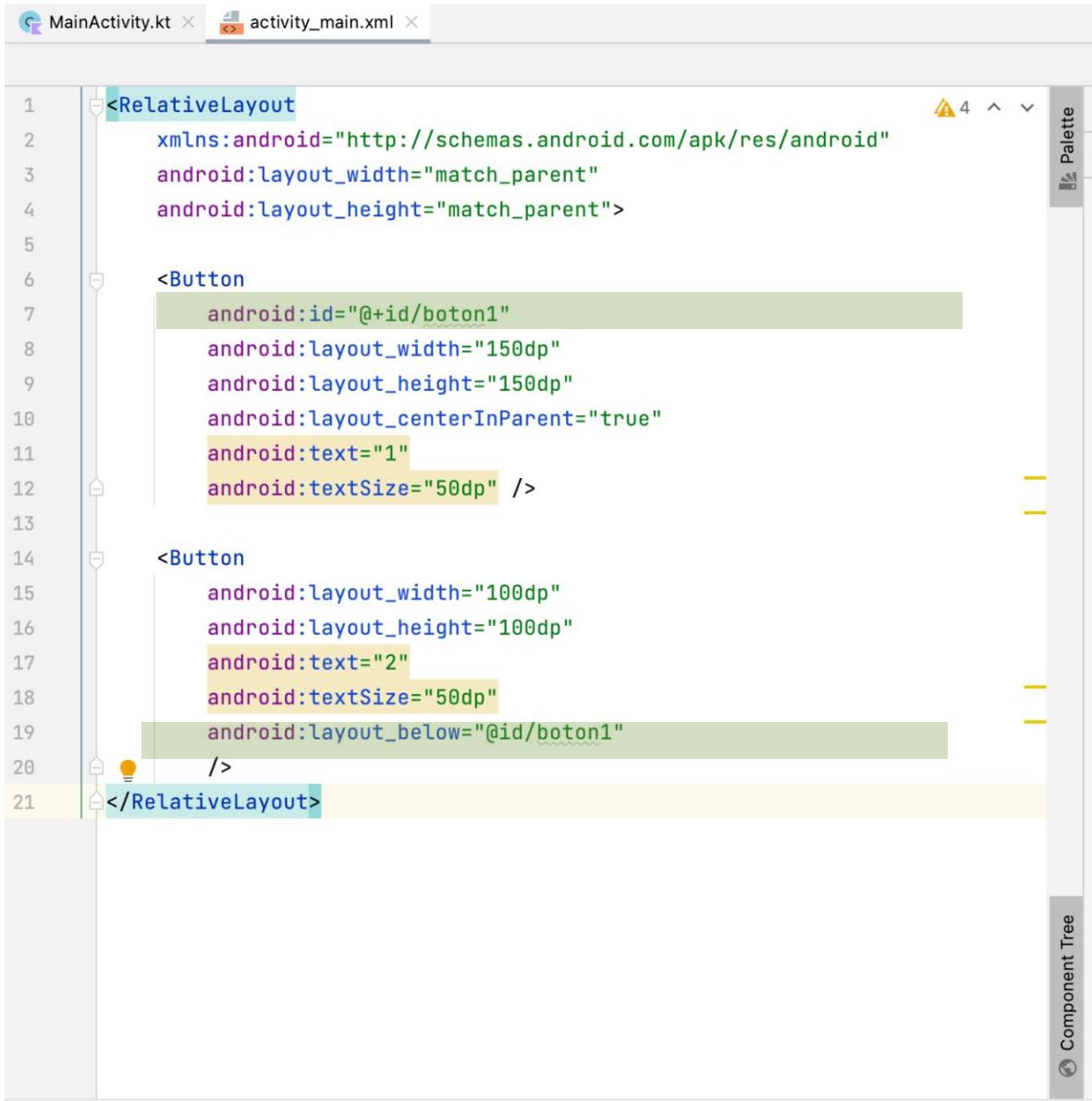
RelativeLayout

```
MainActivity.kt x activity_main.xml x
1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8         android:layout_width="150dp"
9         android:layout_height="150dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12        android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_alignBottom="@+id/boton1"
20     />
21 </RelativeLayout|
```

layout_alignBottom: Alinea el borde inferior de un **View** con el borde inferior de otro **View**



RelativeLayout



MainActivity.kt x activity_main.xml x

```
1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8         android:layout_width="150dp"
9         android:layout_height="150dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12        android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_below="@+id/boton1"
20     />
21 </RelativeLayout>
```

Palette Component Tree

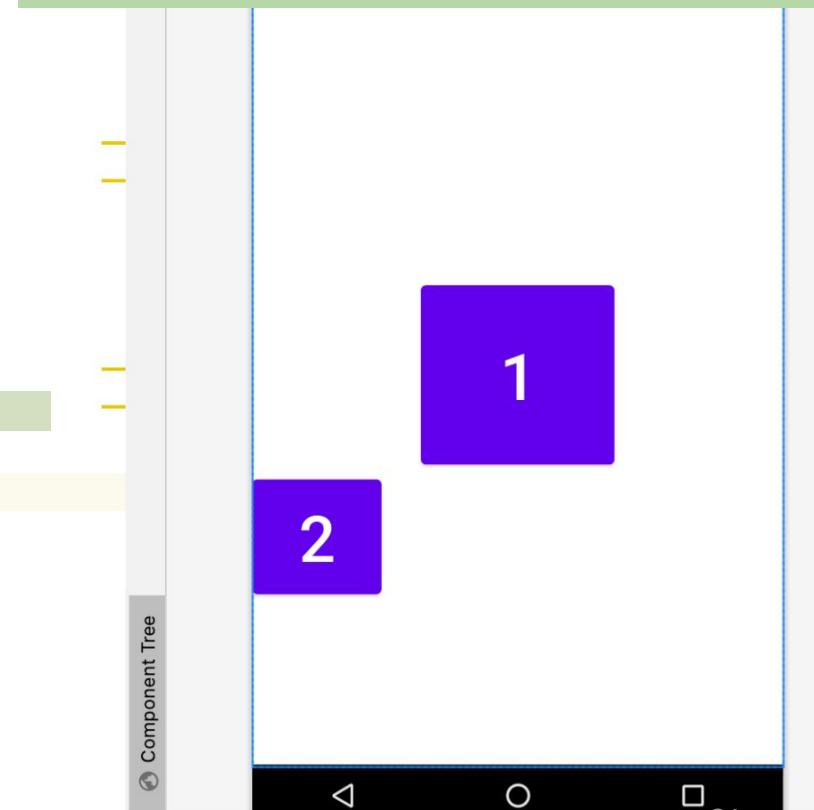
Probar el resultado de la siguiente disposición

RelativeLayout

```
MainActivity.kt x activity_main.xml x

1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8         android:layout_width="150dp"
9         android:layout_height="150dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12        android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_below="@+id/boton1"
20     />
21 </RelativeLayout>
```

layout_below: Alinea el borde superior de un View con el borde inferior de otro View



RelativeLayout

The screenshot shows the Android Studio interface with the tab bar at the top showing 'MainActivity.kt' and 'activity_main.xml'. The code editor displays the XML configuration for a RelativeLayout. The XML code is as follows:

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5  
6     <Button  
7         android:id="@+id/boton1"  
8         android:layout_width="150dp"  
9         android:layout_height="150dp"  
10        android:layout_centerInParent="true"  
11        android:text="1"  
12        android:textSize="50dp" />  
13  
14     <Button  
15         android:layout_width="100dp"  
16         android:layout_height="100dp"  
17         android:text="2"  
18         android:textSize="50dp"  
19         android:layout_below="@+id/boton1"  
20         android:layout_toLeftOf="@+id/boton1"  
21     />  
22 </RelativeLayout>
```

The code defines a RelativeLayout with two buttons. The first button has an id of @+id/boton1, a width of 150dp, a height of 150dp, and is centered in its parent. The second button has an id of @+id/boton1, a width of 100dp, a height of 100dp, and is positioned below the first button and to its left. The XML editor highlights the second button's layout parameters (below and toLeftOf) in green.

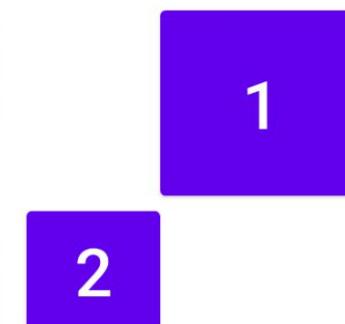
Probar el resultado de la siguiente disposición

RelativeLayout

```
MainActivity.kt x activity_main.xml x

1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8         android:layout_width="150dp"
9         android:layout_height="150dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12        android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_below="@+id/boton1"
20         android:layout_toLeftOf="@+id/boton1"
21     />
22 </RelativeLayout>
```

layout_toLeftOf: Alinea el borde derecho de un View con el borde izquierdo de otro View



RelativeLayout

```
1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8         android:layout_width="150dp"
9         android:layout_height="150dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12        android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_below="@+id/boton1"
20         android:layout_alignLeft="@+id/boton1"
21     />
22 </RelativeLayout>
```

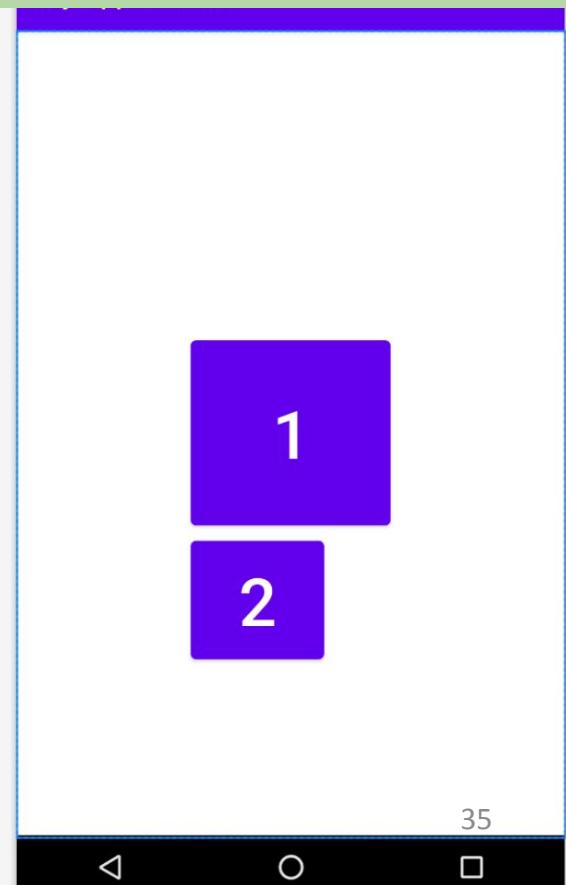
Probar el resultado de la siguiente disposición

RelativeLayout

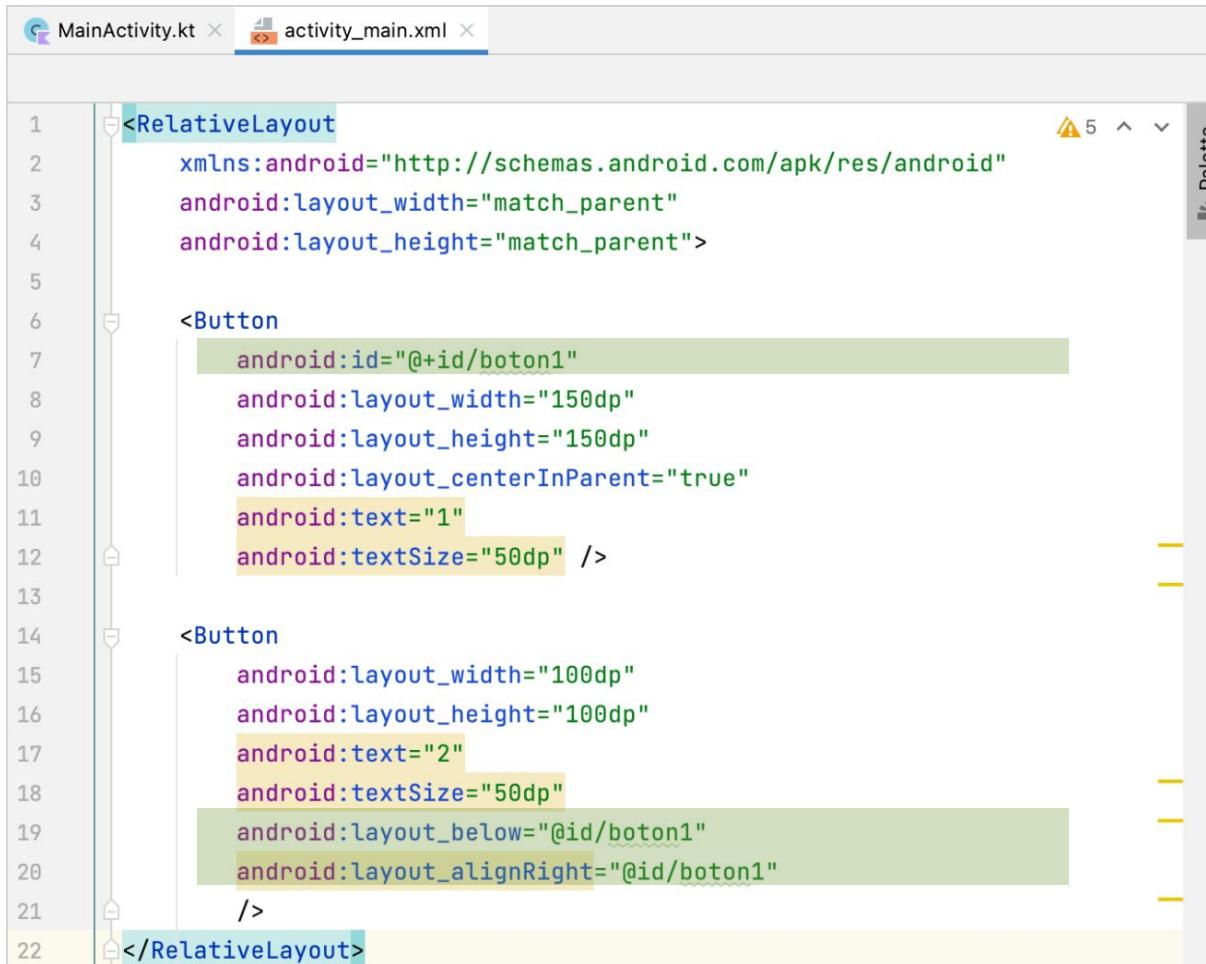
```
MainActivity.kt x activity_main.xml x

1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8         android:layout_width="150dp"
9         android:layout_height="150dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12        android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_below="@+id/boton1"
20         android:layout_alignLeft="@+id/boton1"
21     />
22 </RelativeLayout>
```

layout_alignLeft: Alinea el borde izquierdo de un View con el borde izquierdo de otro View



RelativeLayout



The screenshot shows the Android Studio interface with the tab bar at the top showing 'MainActivity.kt' and 'activity_main.xml'. The code editor displays the XML configuration for a RelativeLayout. The XML code is as follows:

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5  
6     <Button  
7         android:id="@+id/boton1"  
8         android:layout_width="150dp"  
9         android:layout_height="150dp"  
10        android:layout_centerInParent="true"  
11        android:text="1"  
12        android:textSize="50dp" />  
13  
14     <Button  
15         android:layout_width="100dp"  
16         android:layout_height="100dp"  
17         android:text="2"  
18         android:textSize="50dp"  
19         android:layout_below="@+id/boton1"  
20         android:layout_alignRight="@+id/boton1"  
21     />  
22 </RelativeLayout>
```

The code defines a RelativeLayout with two buttons. The first button has an id of @+id/boton1, a width of 150dp, a height of 150dp, is centered in its parent, and contains the text "1" with a size of 50dp. The second button has a width of 100dp, a height of 100dp, contains the text "2" with a size of 50dp, is positioned below the first button, and aligns its right edge with the right edge of the first button.

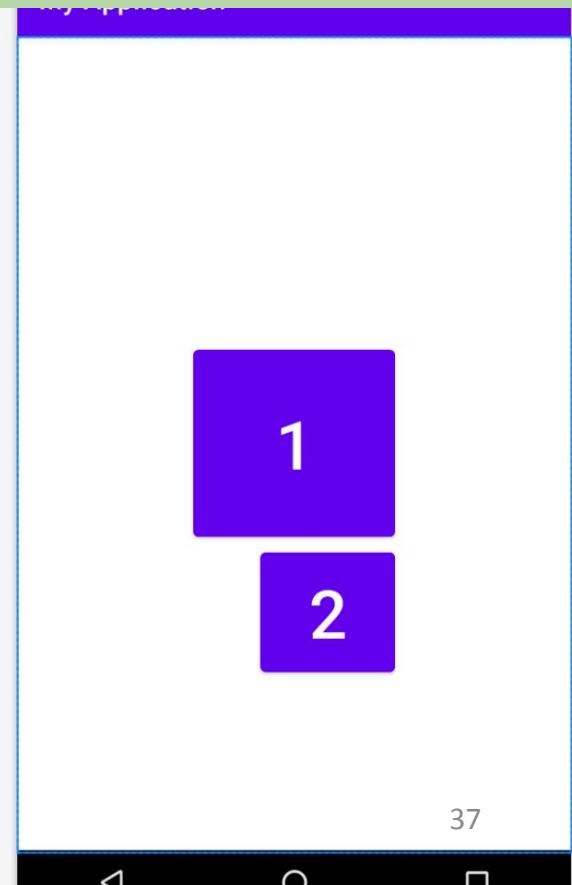
Probar el resultado de la siguiente disposición

RelativeLayout

```
MainActivity.kt x activity_main.xml x

1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8         android:layout_width="150dp"
9         android:layout_height="150dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12        android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_below="@+id/boton1"
20         android:layout_alignRight="@+id/boton1"
21     />
22 </RelativeLayout>
```

layout_alignRight: Alinea el borde derecho de un View con el borde derecho de otro View



RelativeLayout

The screenshot shows the Android Studio interface with the activity_main.xml file selected in the top bar. The code editor displays the following XML configuration:

```
1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8         android:layout_width="150dp"
9         android:layout_height="150dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12        android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_below="@+id/boton1"
20         android:layout_toRightOf="@+id/boton1"
21     />
22 </RelativeLayout>
```

The XML code defines a `RelativeLayout` with two children: two `Button` elements. The first button has a width of 150dp, height of 150dp, and is centered in its parent. The second button has a width of 100dp, height of 100dp, and is positioned below the first button and to its right. Both buttons have a text size of 50dp.

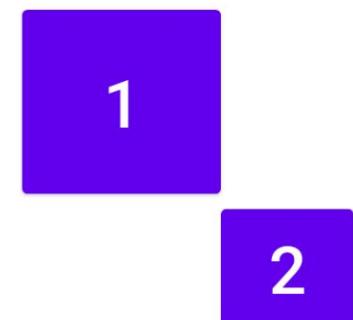
Probar el resultado de la siguiente disposición

RelativeLayout

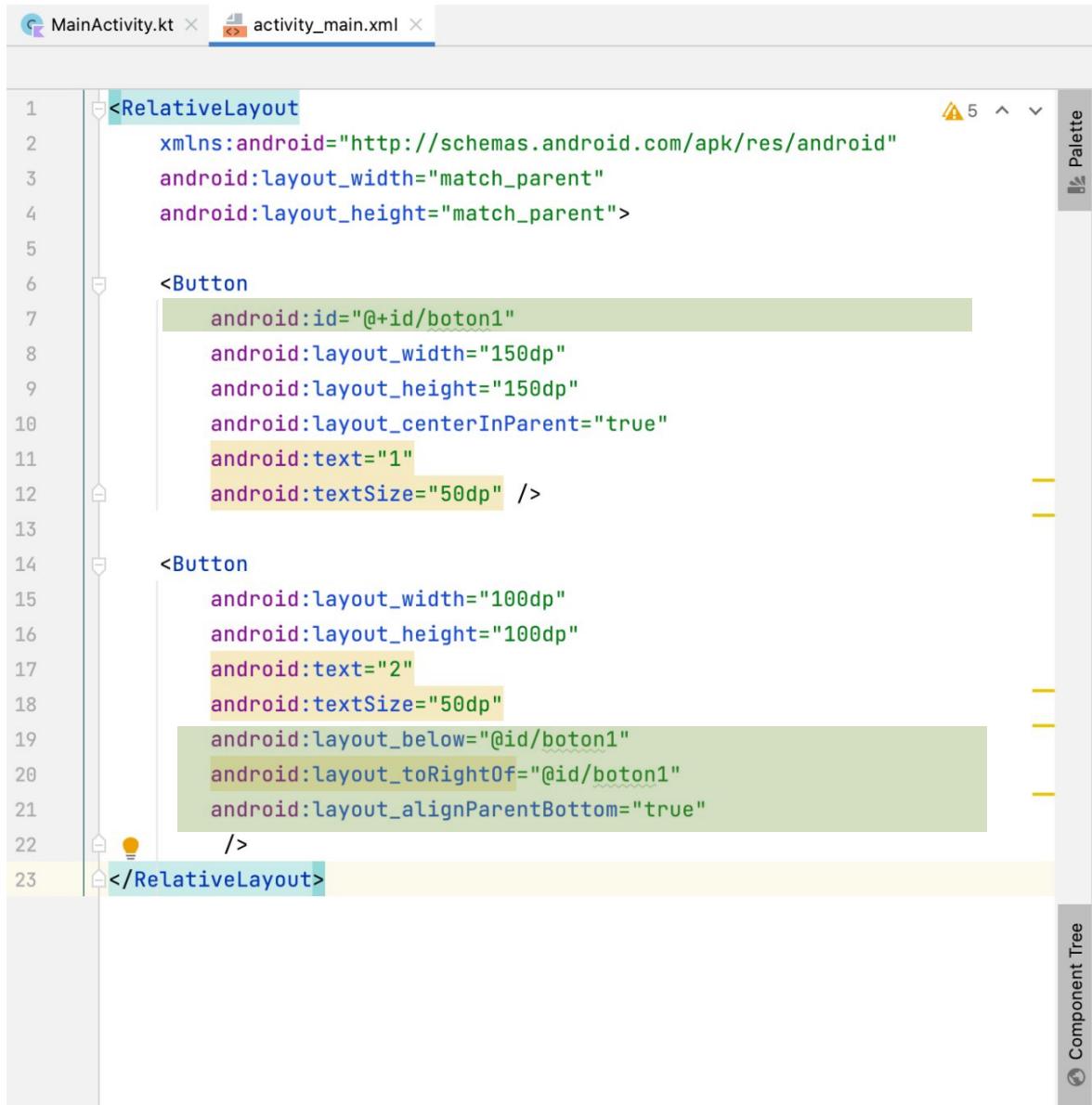
```
MainActivity.kt x activity_main.xml x

1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8         android:layout_width="150dp"
9         android:layout_height="150dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12        android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_below="@+id/boton1"
20         android:layout_toRightOf="@+id/boton1"
21         />
22 </RelativeLayout>
```

layout_toRightOf: Alinea el borde izquierdo de un **View** con el borde derecho de otro **View**



RelativeLayout



The screenshot shows the Android Studio interface with the activity_main.xml file selected in the tab bar. The code editor displays the following XML configuration:

```
1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8         android:layout_width="150dp"
9         android:layout_height="150dp"
10        android:layout_centerInParent="true"
11        android:text="1"
12        android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_below="@+id/boton1"
20         android:layout_toRightOf="@+id/boton1"
21         android:layout_alignParentBottom="true"
22     />
23 </RelativeLayout>
```

The code defines a `RelativeLayout` with two children: a first `Button` and a second `Button`. The first button has dimensions of 150dp by 150dp, is centered in its parent, and contains the text "1". The second button has dimensions of 100dp by 100dp, is positioned below the first button, to the right of it, and aligned to the bottom of its parent, containing the text "2". A warning icon (yellow triangle) is visible in the top right corner of the code editor.

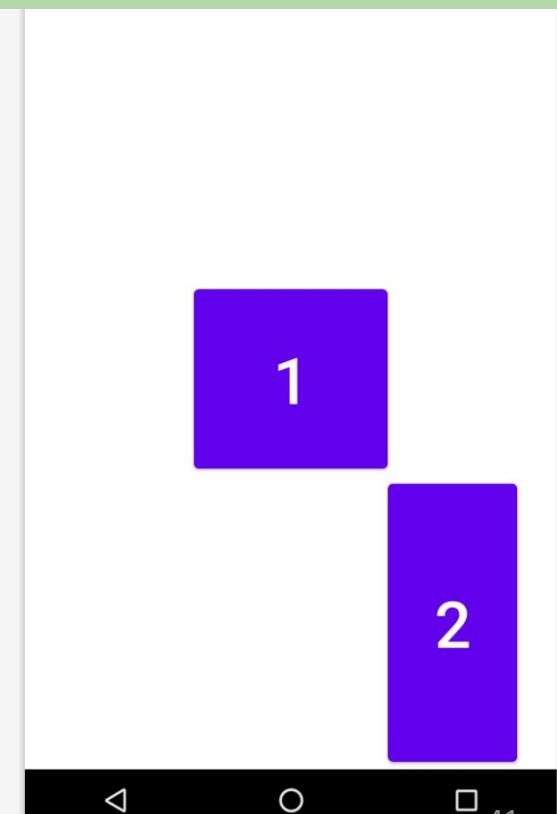
Probar el resultado de la siguiente disposición

RelativeLayout

```
MainActivity.kt x activity_main.xml x

1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3         android:layout_width="match_parent"
4         android:layout_height="match_parent">
5
6     <Button
7         android:id="@+id/boton1"
8             android:layout_width="150dp"
9             android:layout_height="150dp"
10            android:layout_centerInParent="true"
11            android:text="1"
12            android:textSize="50dp" />
13
14     <Button
15         android:layout_width="100dp"
16         android:layout_height="100dp"
17         android:text="2"
18         android:textSize="50dp"
19         android:layout_below="@+id/boton1"
20         android:layout_toRightOf="@+id/boton1"
21         android:layout_alignParentBottom="true"
22     />
23 </RelativeLayout>
```

Se pueden combinar distintas disposiciones para obtener resultados muy variados



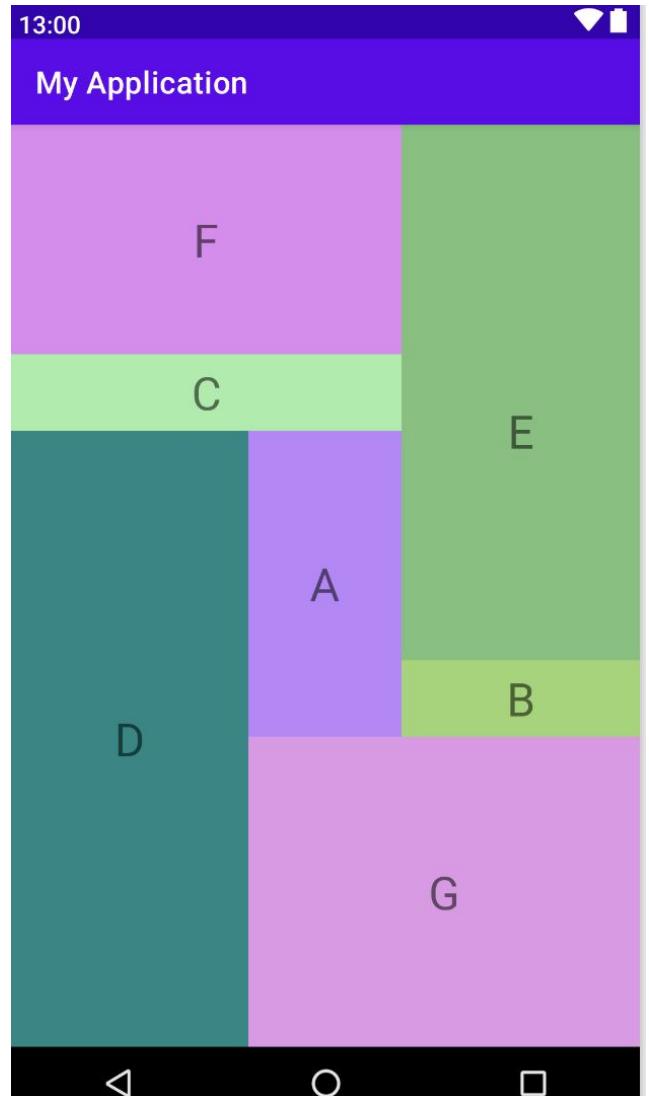
Relacionado con las alineaciones horizontales y teniendo en cuenta los idiomas **RTL** también existen:

- **layout_alignParentStart**
- **layout_alignParentEnd**
- **layout_alignStart**
- **layout_alignEnd**
- **layout_toStartOf**
- **layout_toEndOf**

Para el idioma español **Start** funciona como **Left** y **End** funciona como **Right**. Para los idiomas **RTL** justo lo contrario

Ejercicio

- Diseñar la siguiente interfaz
- los **views** son **TextView** a los que se les ha cambiado el color del background
- Utilizar el atributo **gravity** para centrar el texto dentro del TextView



```
<RelativeLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

```
        <TextView  
            android:id="@+id/botonF"  
            android:layout_width="200dp"  
            android:layout_height="150dp"  
            android:text="F"  
            android:gravity="center"  
            android:textSize="30dp"  
            android:background="#E189F1"  
            android:layout_alignRight="@id/botonA"  
            android:layout_alignParentLeft="true"/>
```

```
        <TextView  
            android:id="@+id/botonE"  
            android:layout_width="150dp"  
            android:layout_height="150dp"  
            android:text="E"  
            android:gravity="center"  
            android:textSize="30dp"  
            android:layout_alignParentRight="true"  
            android:layout_alignParentTop="true"  
            android:background="#79C17A"  
            android:layout_toRightOf="@id/botonF"  
            android:layout_alignBottom="@id/botonB"  
        />
```

```
<TextView
```

```
    android:id="@+id/botonC"  
    android:layout_width="150dp"  
    android:layout_height="50dp"  
    android:text="C"  
    android:gravity="center"  
    android:textSize="30dp"  
    android:layout_below="@id/botonF"  
    android:background="#A1EDA8"  
    android:layout_alignRight="@id/botonA"  
    android:layout_alignParentLeft="true"  
/>
```

```
<TextView
```

```
    android:id="@+id/botonD"  
    android:layout_width="150dp"  
    android:layout_height="50dp"  
    android:text="D"  
    android:gravity="center"  
    android:textSize="30dp"  
    android:layout_below="@id/botonC"  
    android:background="@color/teal_700"  
    android:layout_toLeftOf="@id/botonA"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentBottom="true"  
/>
```

```
<TextView
```

```
    android:id="@+id/botonA"  
    android:layout_width="100dp"  
    android:layout_height="200dp"  
    android:text="A"  
    android:textSize="30dp"  
    android:layout_below="@id/botonC"  
    android:background="@color/purple_200"  
    android:gravity="center"  
    android:layout_centerInParent="true"  
/>
```

```
<TextView
```

```
    android:id="@+id/botonG"  
    android:layout_width="150dp"  
    android:layout_height="50dp"  
    android:text="G"  
    android:gravity="center"  
    android:textSize="30dp"  
    android:layout_below="@id/botonA"  
    android:background="#E398E8"
```

```
    android:layout_alignParentBottom="true"
```

```
    android:layout_alignParentRight="true"
```

```
    android:layout_toRightOf="@id/botonD"  
/>
```

```
<TextView
```

```
    android:id="@+id/botonB"  
    android:layout_width="150dp"  
    android:layout_height="50dp"  
    android:text="B"  
    android:gravity="center"  
    android:textSize="30dp"  
    android:layout_above="@id/botonG"  
    android:background="#99D56F"
```

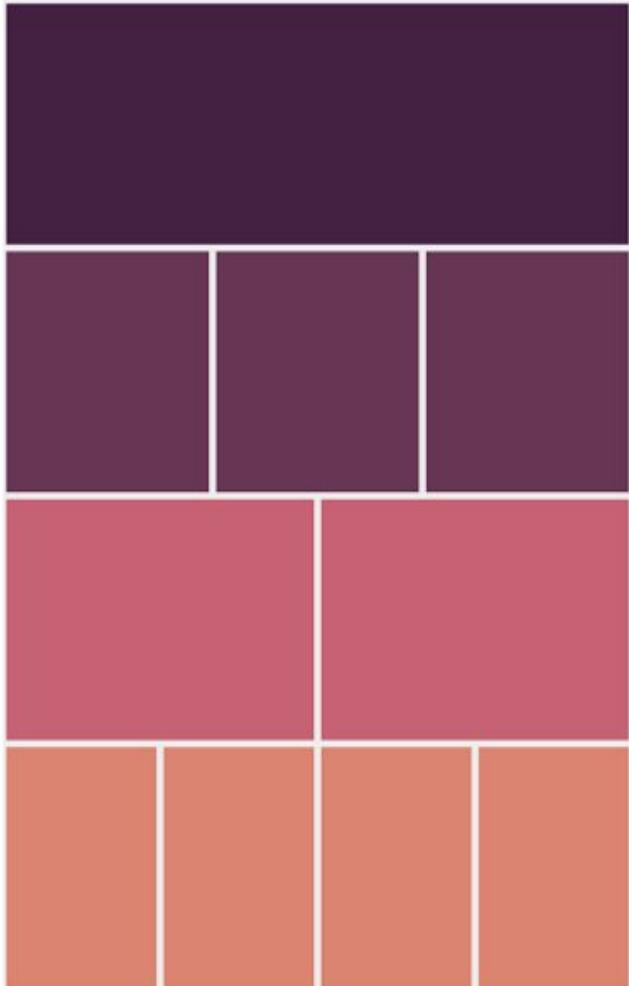
```
    android:layout_alignParentRight="true"
```

```
    android:layout_toRightOf="@id/botonA"  
/>
```

```
</RelativeLayout>
```

Possible solución

TableLayout

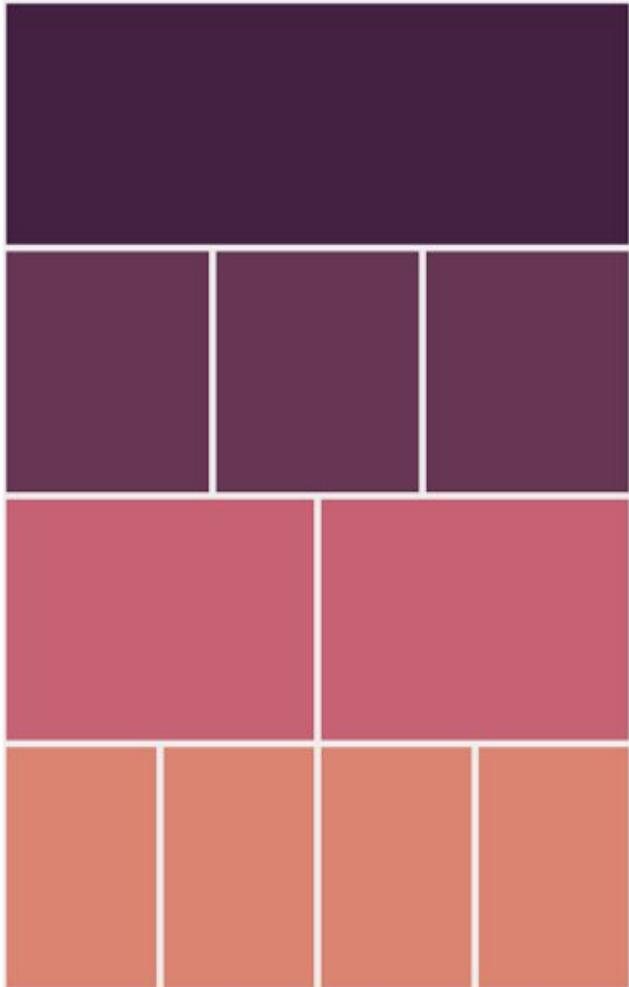


Es un **ViewGroup** que ubica a sus hijos en filas y columnas

Generalmente está compuesto de objetos **TableRow**.

Cada **TableRow** puede tener 0, 1 o más celdas.

TableLayout

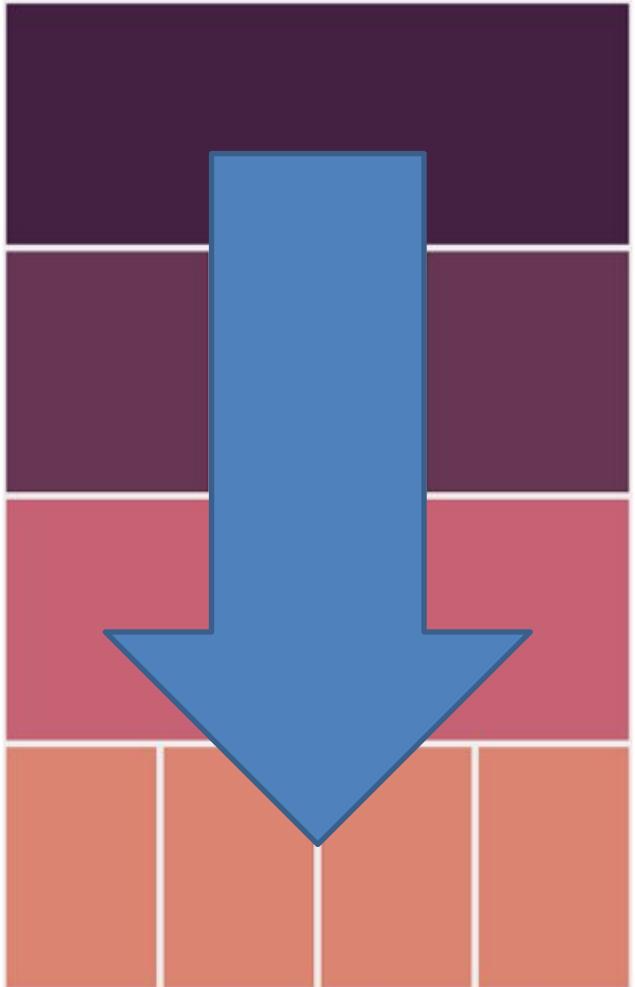


La tabla tendrá tantas columnas como aquel **TableRow** con más celdas.

Una celda podría expandirse ocupando más de una columna.

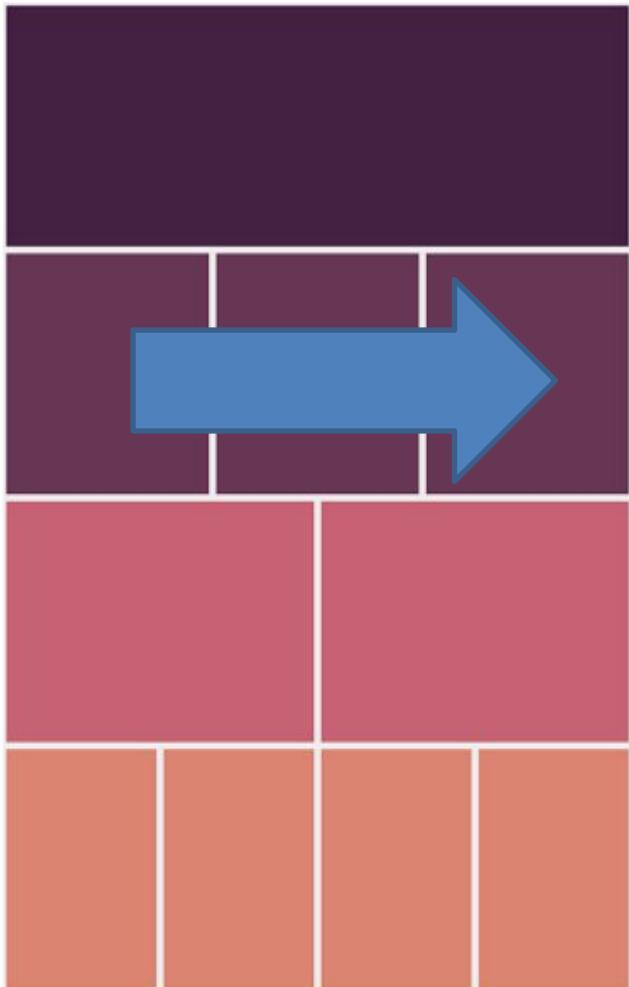
Un **TableLayout** no contempla bordes. Simplemente organiza elementos visuales

TableLayout



Un **TableLayout** es una especialización de un **LinearLayout** con orientación vertical.

TableLayout



Asimismo, cada TableRow es una especialización de un LinearLayout con orientación horizontal.

Ejemplo de TableLayout

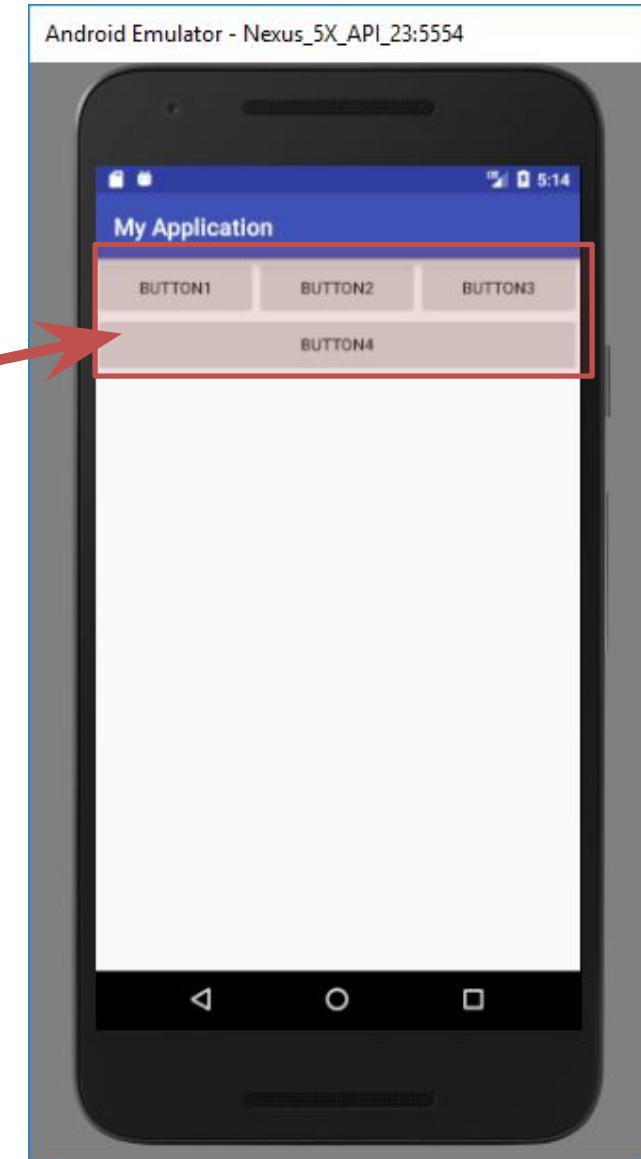
```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="*">
    <TableRow>
        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="button1" />
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="button2" />
        <Button
            android:id="@+id/button3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="button3" />
    </TableRow>
    <TableRow>
        <Button
            android:id="@+id/button4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_span="3"
            android:text="button4" />
    </TableRow>
</TableLayout>
```



Ejemplo de TableLayout

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="*">
    <TableRow>
        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <Button
            android:id="@+id/button3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </TableRow>
    <TableRow>
        <Button
            android:id="@+id/button4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_span="3"
            android:text="button4" />
    </TableRow>
</TableLayout>
```

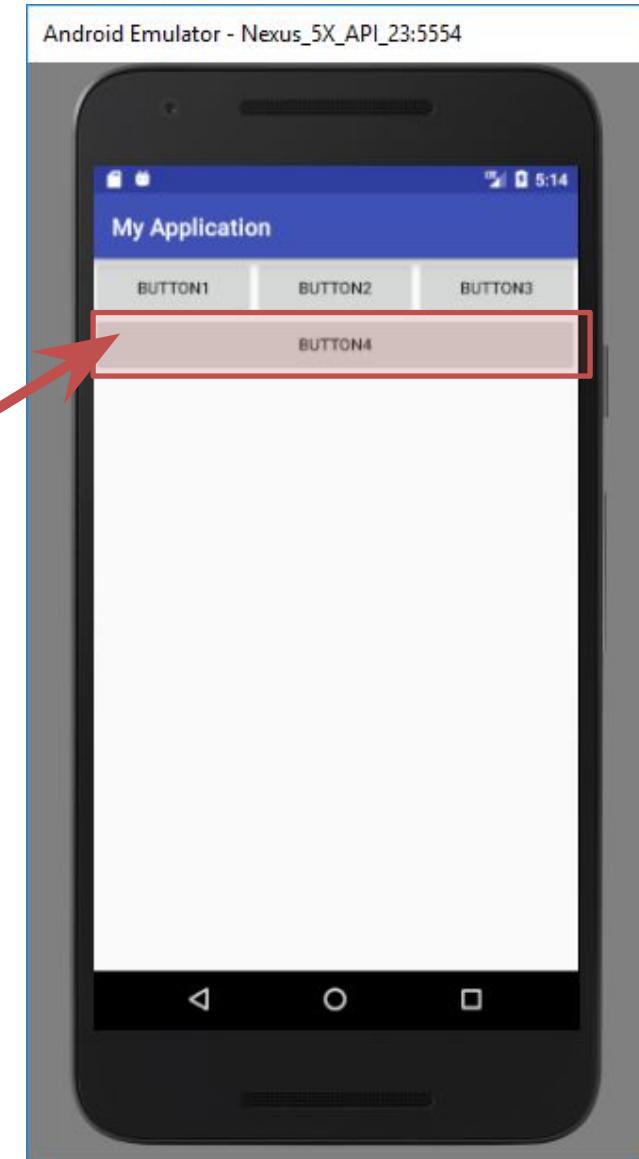
Estira las columnas para ocupar todo el ancho del TableLayout



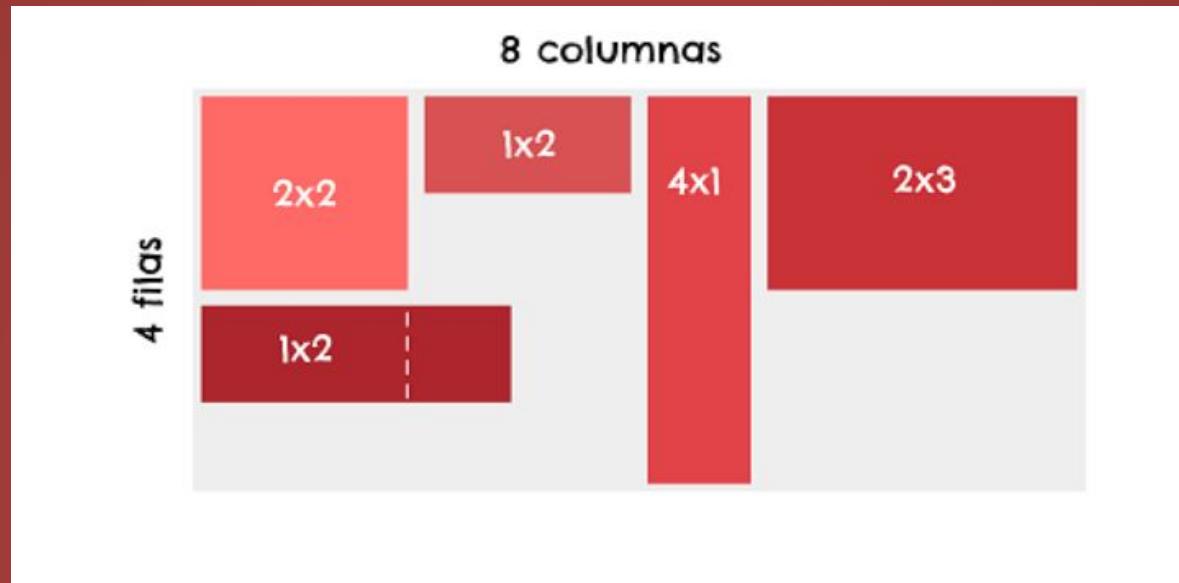
Ejemplo de TableLayout

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="*">
    <TableRow>
        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="button1" />
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="button2" />
        <Button
            android:id="@+id/button3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="button3" />
    </TableRow>
    <TableRow>
        <Button
            android:id="@+id/button4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_span="3"
            android:text="button4" />
    </TableRow>
</TableLayout>
```

La vista ocupa 3 columnas

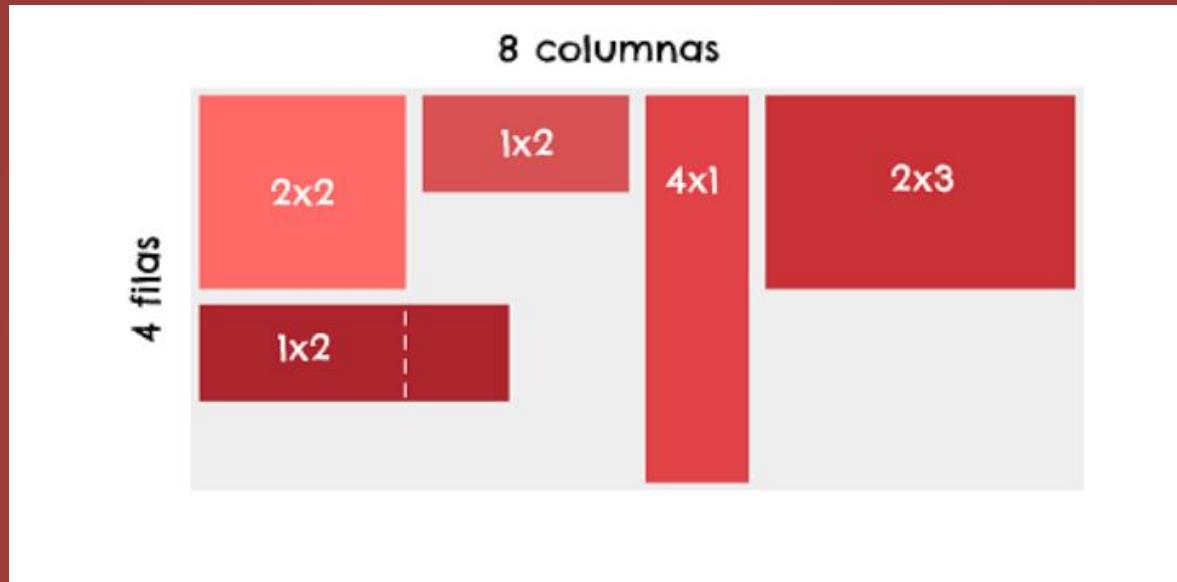


GridLayout



- Incluido a partir de Andriod 4.0 (API 14)
- Es un `ViewGroup` que ubica a sus hijos en una grilla rectangular.
Es posible especificar cantidad de filas y columnas.
A diferencia del `TableLayout`, es posible expandir una celda de forma horizontal o vertical.

GridLayout



- No existe un concepto análogo a **TableRow**. Los elementos hijos se irán colocando ordenadamente por filas o columnas (dependiendo de la propiedad `android:orientation`) hasta completar el número de filas o columnas.

Ejemplo de GridLayout

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="4"
    android:orientation="horizontal" >

    <Button android:text="Botón 1.1" />
    <Button android:text="Botón 1.2" />
    <Button android:text="Botón 1.3"
        android:layout_rowSpan="2"/>
    <Button android:text="Botón 1.4" />

    <Button android:text="Botón 2.1"
        android:layout_columnSpan="2" />
    <Button android:text="Botón 2.4" />

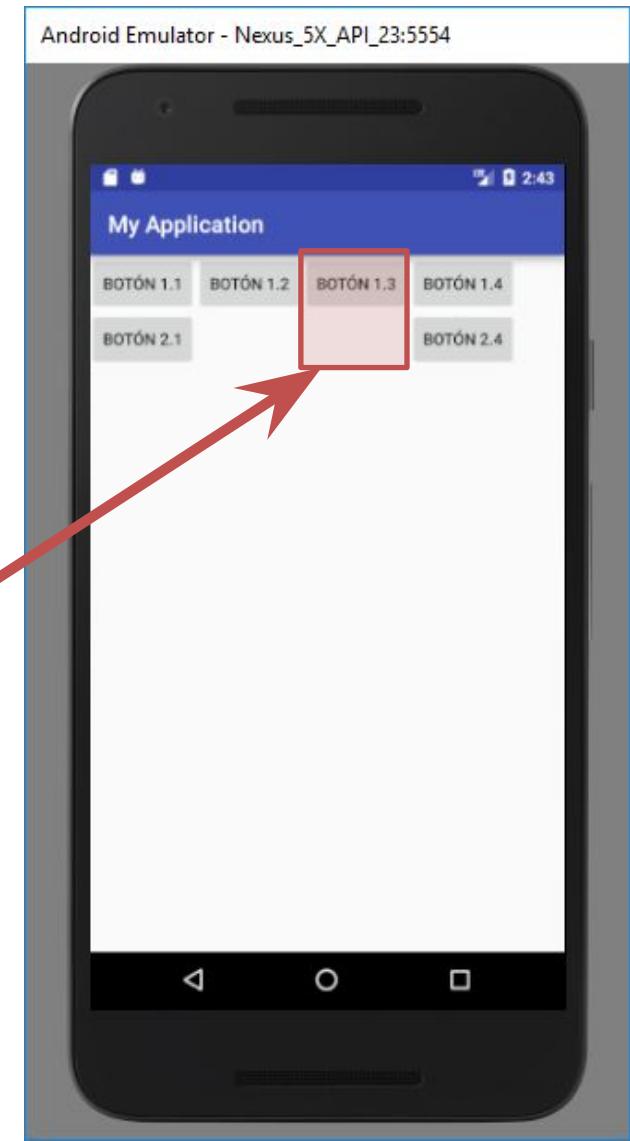
</GridLayout>
```



Ejemplo de GridLayout

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:columnCount="4"  
    android:orientation="horizontal" >  
  
    <Button android:text="Botón 1.1" />  
    <Button android:text="Botón 1.2" />  
    <Button android:text="Botón 1.3"  
        android:layout_rowSpan="2"/>  
    <Button android:text="Botón 1.4" />  
  
    <Button android:text="Botón 2.1"  
        android:layout_columnSpan="2" />  
    <Button android:text="Botón 2.4" />  
  
</GridLayout>
```

La vista ocupa 2 filas



Ejemplo de GridLayout

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:columnCount="4"  
    android:orientation="horizontal" >  
  
    <Button android:text="Botón 1.1" />  
    <Button android:text="Botón 1.2" />  
    <Button android:text="Botón 1.3"  
        android:layout_rowSpan="2"/>  
    <Button android:text="Botón 1.4" />  
  
    <Button android:text="Botón 2.1"  
        android:layout_columnSpan="2"/>  
    <Button android:text="Botón 2.4" />  
  
</GridLayout>
```

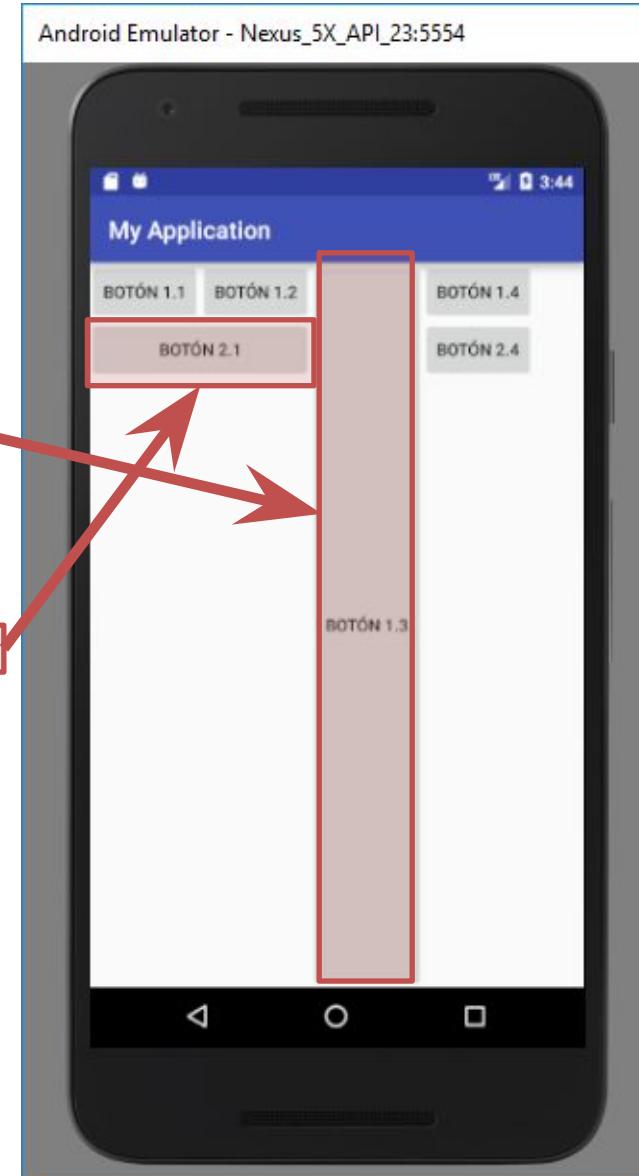
La vista ocupa 2 columnas



Ejemplo de GridLayout

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:columnCount="4"  
    android:orientation="horizontal" >  
  
<Button android:text="Botón 1.1" />  
<Button android:text="Botón 1.2" />  
<Button android:text="Botón 1.3"  
        android:layout_rowSpan="2"  
        android:layout_gravity="fill_vertical"/>  
<Button android:text="Botón 1.4" />  
  
<Button android:text="Botón 2.1"  
        android:layout_columnSpan="2"  
        android:layout_gravity="fill_horizontal"/>  
<Button android:text="Botón 2.4" />  
  
</GridLayout>
```

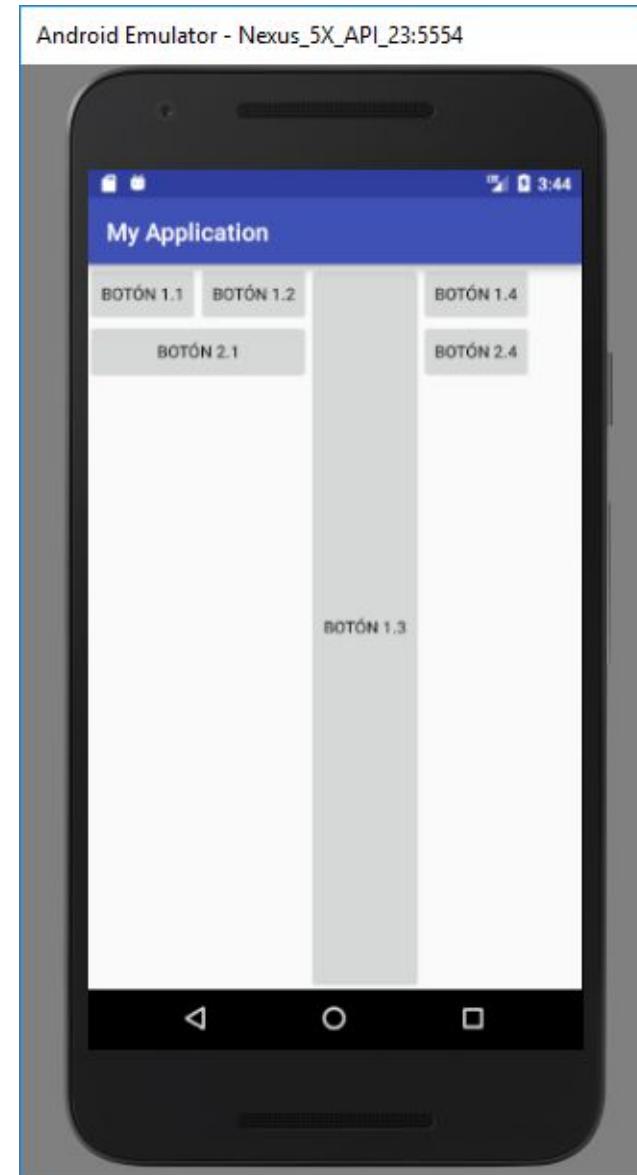
Es posible expandir las
vistas para que ocupen
toda la celda



Ejemplo de GridLayout

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:columnCount="4"  
    android:orientation="horizontal" >  
  
<Button android:text="Botón 1.1" />  
<Button android:text="Botón 1.2" />  
<Button android:text="Botón 1.3"  
        android:layout_rowSpan="2"  
        android:layout_gravity="fill_vertical"/>  
<Button android:text="Botón 1.4" />  
  
<Button android:text="Botón 2.1"  
        android:layout_columnSpan="2"  
        android:layout_gravity="fill_horizontal"/>  
<Button android:text="Botón 2.4" />  
  
</GridLayout>
```

¿Cuál es la razón para que el botón 1.3 se expanda al total del dispositivo?



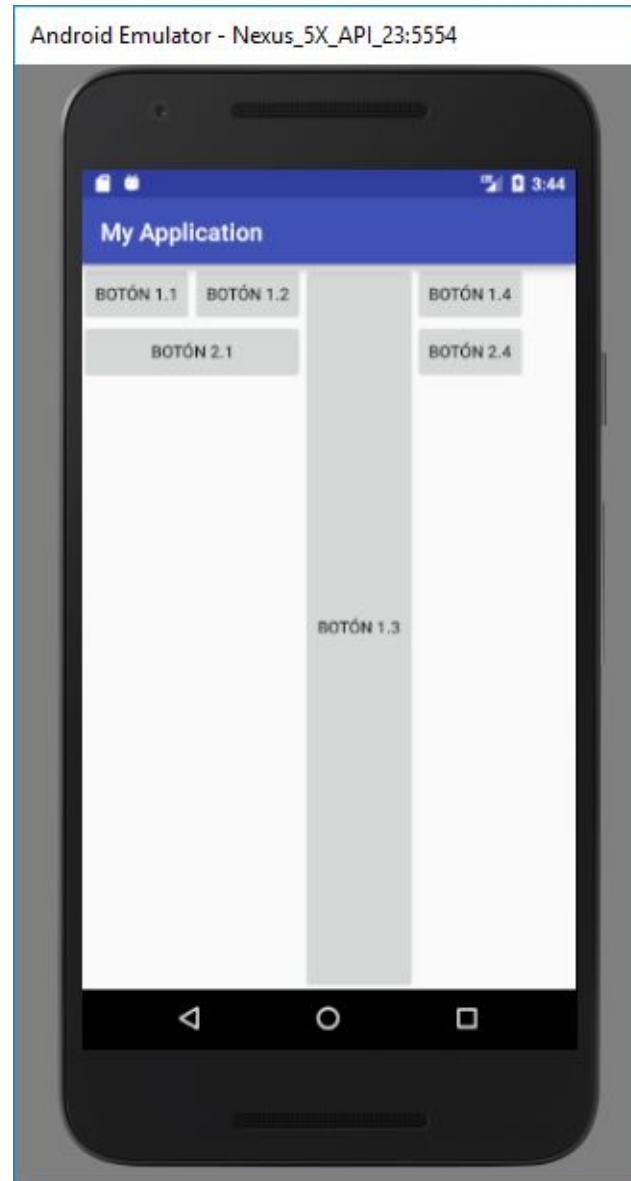
Ejemplo de GridLayout

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="4"
    android:orientation="horizontal" >

    <Button android:text="Botón 1.1" />
    <Button android:text="Botón 1.2" />
    <Button android:text="Botón 1.3"
        android:layout_rowSpan="2"
        android:layout_gravity="fill_vertical"/>
    <Button android:text="Botón 1.4" />

    <Button android:text="Botón 2.1"
        android:layout_columnSpan="2"
        android:layout_gravity="fill_horizontal"/>
    <Button android:text="Botón 2.4" />

</GridLayout>
```



Ejemplo de GridLayout

```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:columnCount="4"  
    android:orientation="horizontal" >  
  
<Button android:text="Botón 1.1" />  
<Button android:text="Botón 1.2" />  
<Button android:text="Botón 1.3"  
        android:layout_rowSpan="2"  
        android:layout_gravity="fill_vertical"/>  
<Button android:text="Botón 1.4" />  
  
<Button android:text="Botón 2.1"  
        android:layout_columnSpan="2"  
        android:layout_gravity="fill_horizontal"/>  
<Button android:text="Botón 2.4" />  
  
</GridLayout>
```

Cambiar el valor a wrap_content
y probar en emulador

