

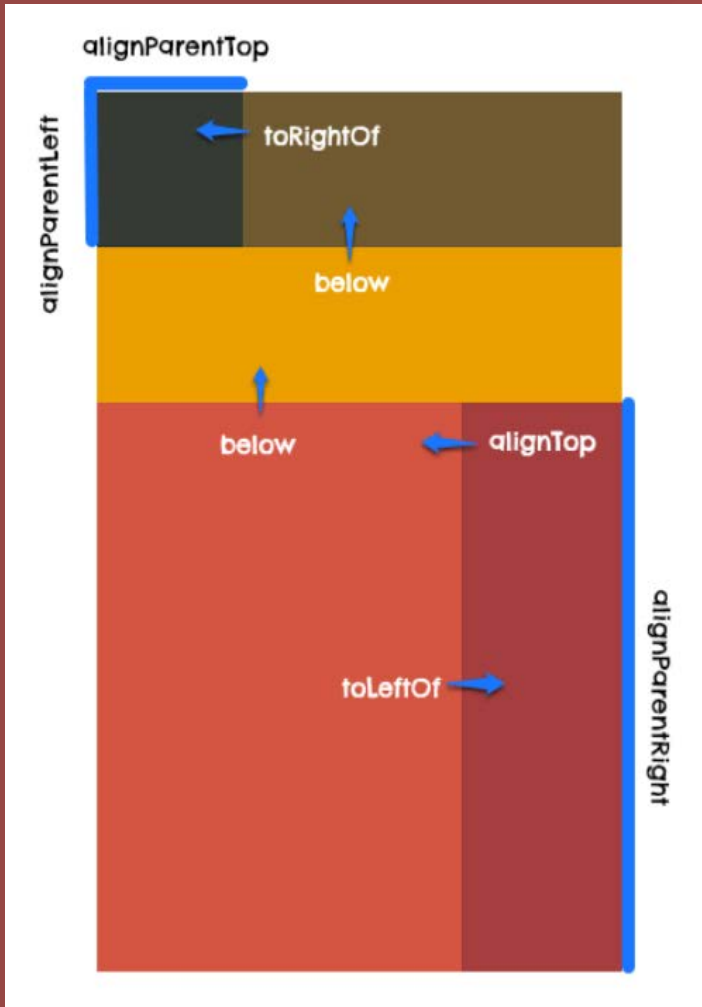
SEMINARIO DE LENGUAJES OPCIÓN ANDROID



Layouts

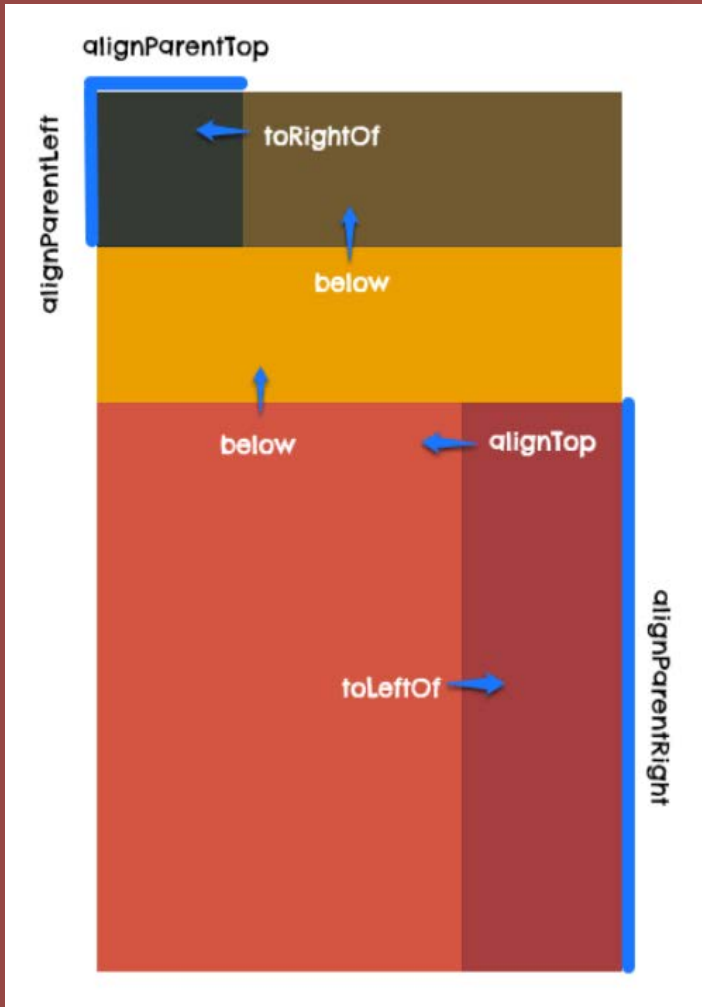
Esp. Delía Lisandro, Mg. Corbalán Leonardo

RelativeLayout



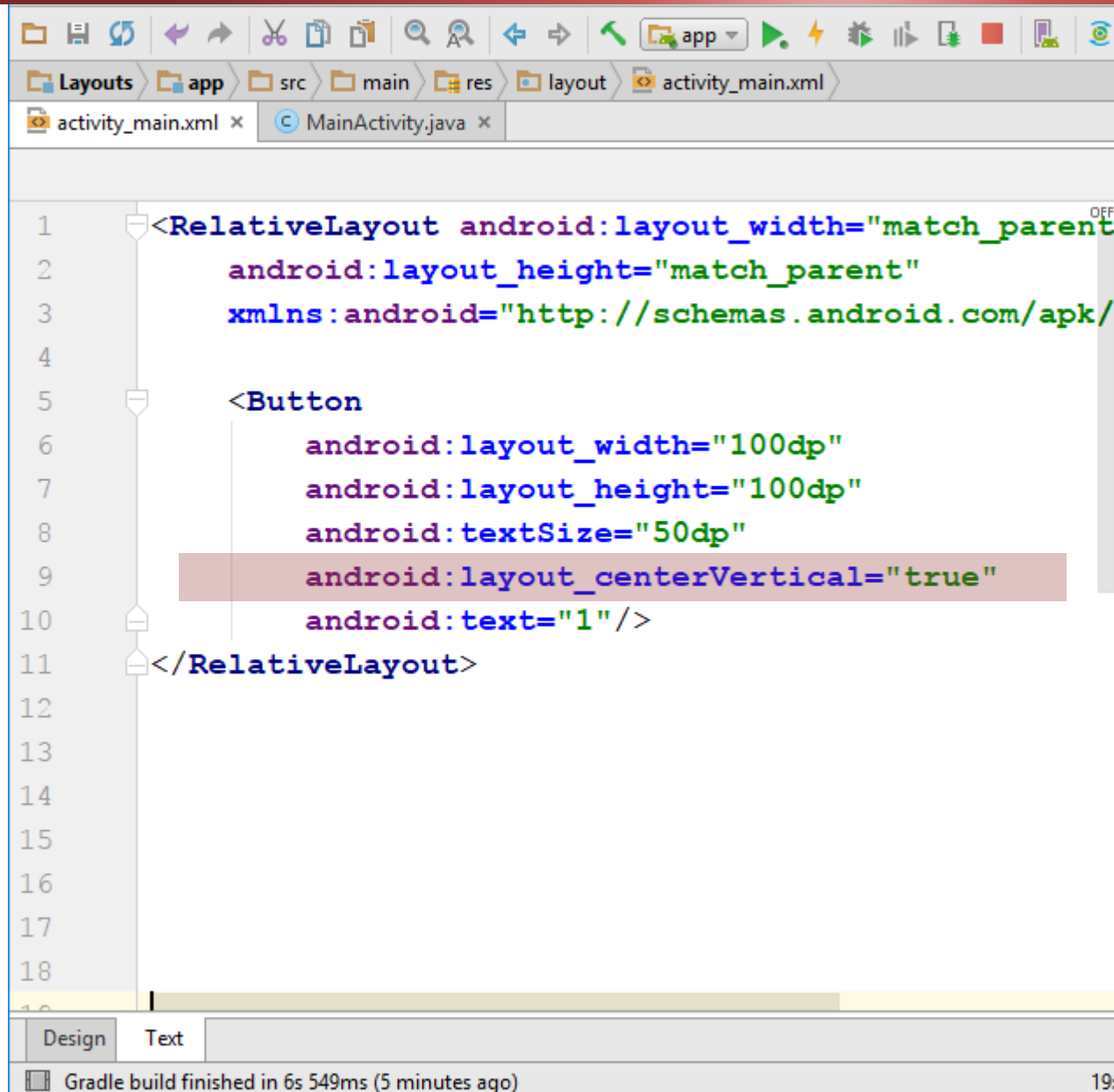
Es un **Layout** donde las posiciones de los elementos contenidos (hijos) pueden ser descritas en relación con otros elementos hijos o bien en relación con el padre, es decir, el propio **Layout**

RelativeLayout



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

RelativeLayout



Probar el
resultado de la
siguiente
disposición

RelativeLayout

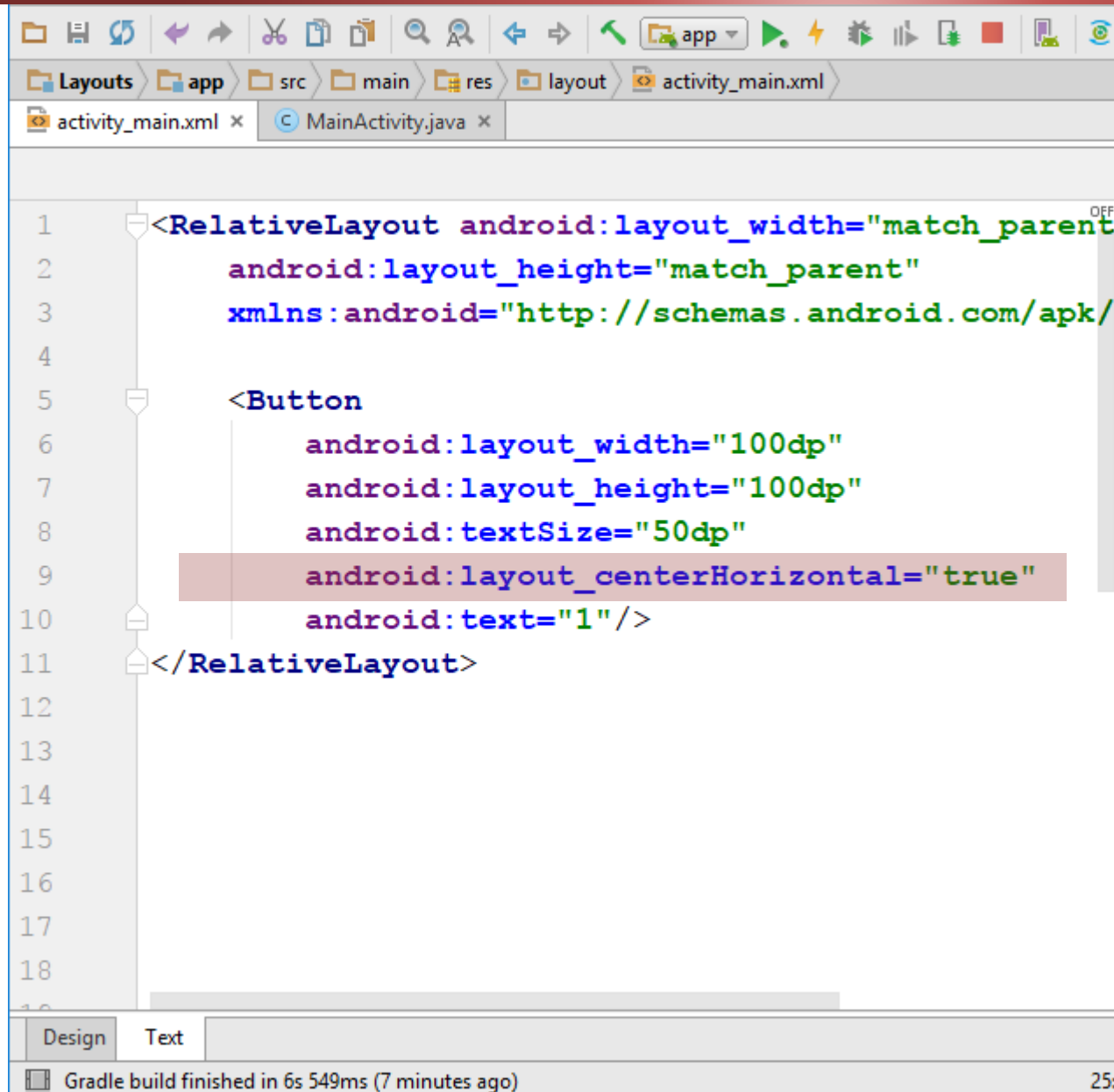
layout_centerVertical: Centra al **View** verticalmente con respecto a los límites de su contenedor

The screenshot displays the Android Studio IDE. On the left, the XML editor shows the following code:

```
1 <RelativeLayout android:layout_width="match_parent"
2   android:layout_height="match_parent"
3   xmlns:android="http://schemas.android.com/apk/
4
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 </RelativeLayout>
```

The line `android:layout_centerVertical="true"` is highlighted with a red background. On the right, the Design tab shows a visual representation of the layout on a mobile device. A blue header bar at the top contains the text "Layouts". Below it, a gray square with the number "1" is centered vertically within the main content area. The device's status bar at the top shows the time as 7:00. The bottom of the screen shows the Android navigation bar with back, home, and recent apps icons. The bottom status bar of the IDE shows "Gradle build finished in 6s 549ms (5 minutes ago)", the time "19:1", and the encoding "UTF-8".

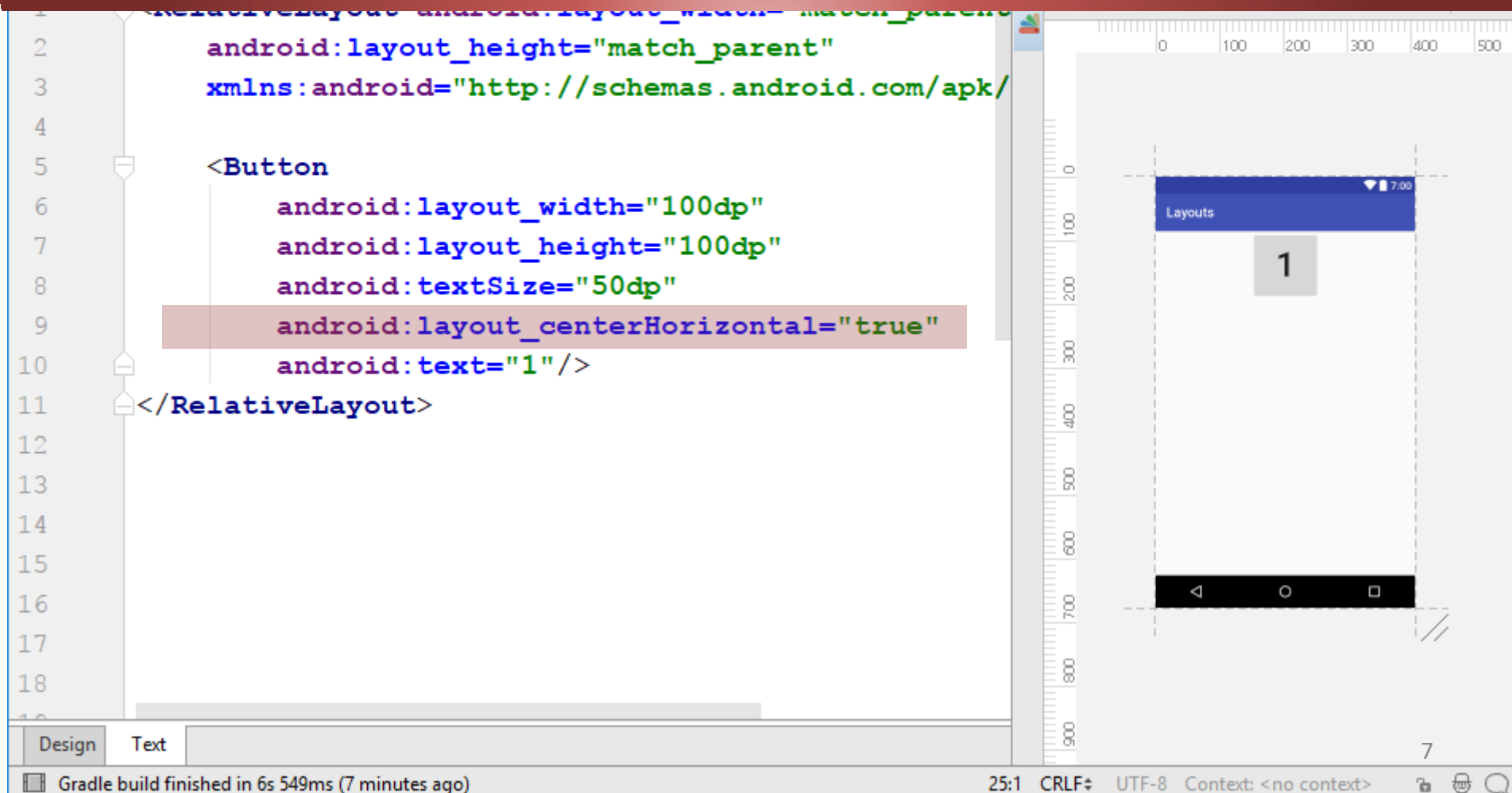
RelativeLayout



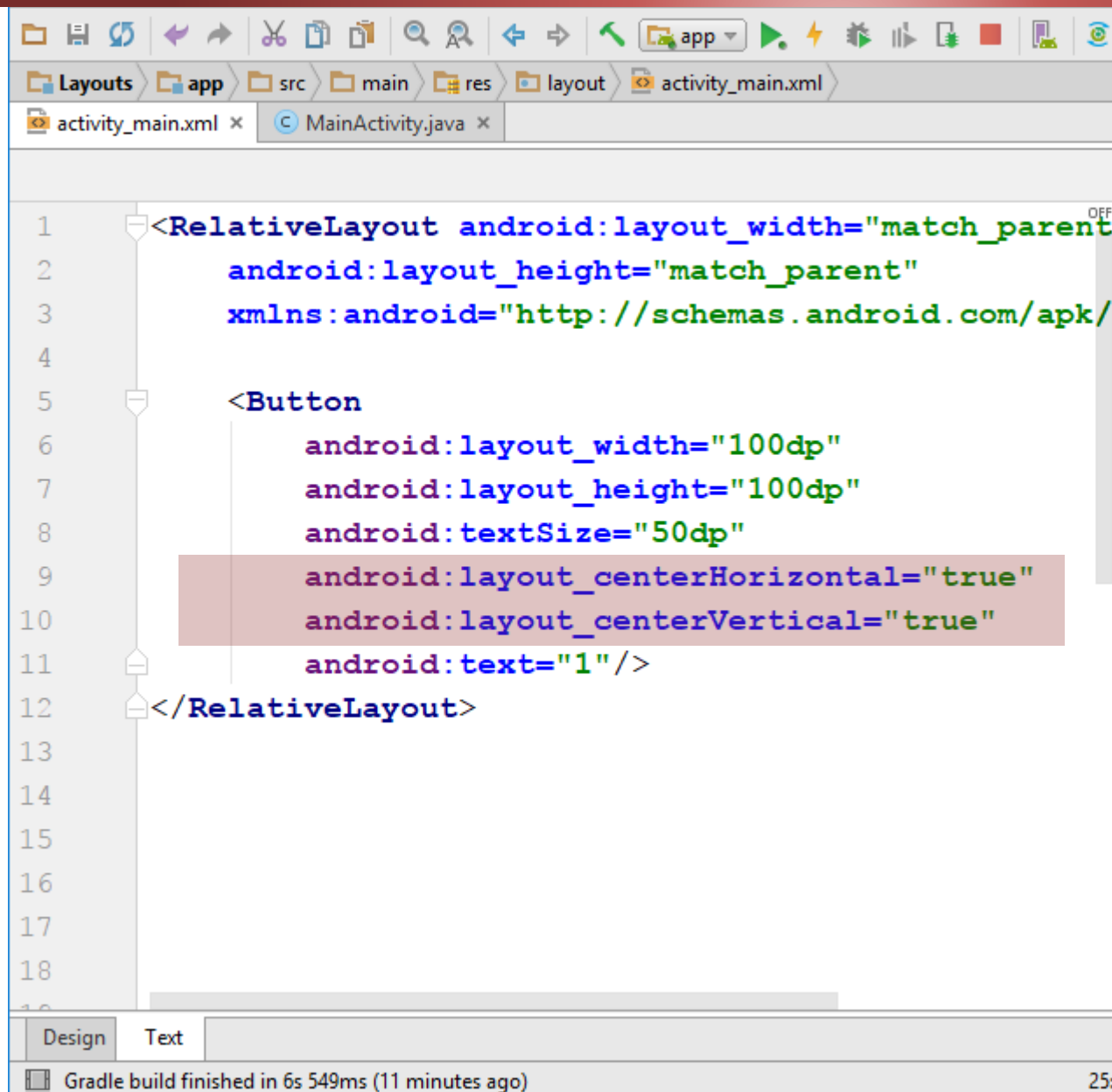
Probar el
resultado de la
siguiente
disposición

RelativeLayout

layout_centerHorizontal: Centra al **View** horizontalmente con respecto a los límites de su contenedor



RelativeLayout

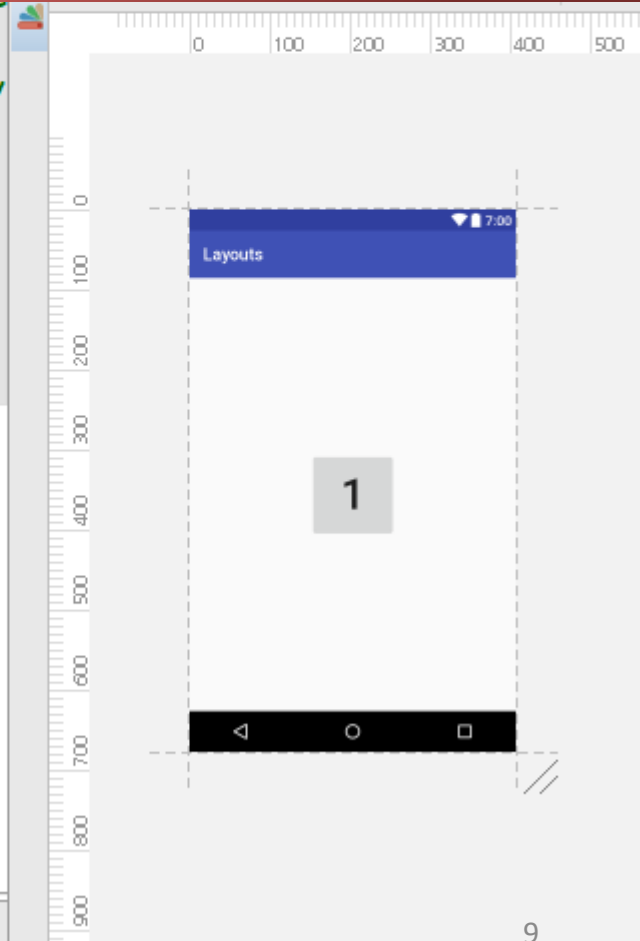


Probar el
resultado de la
siguiente
disposición

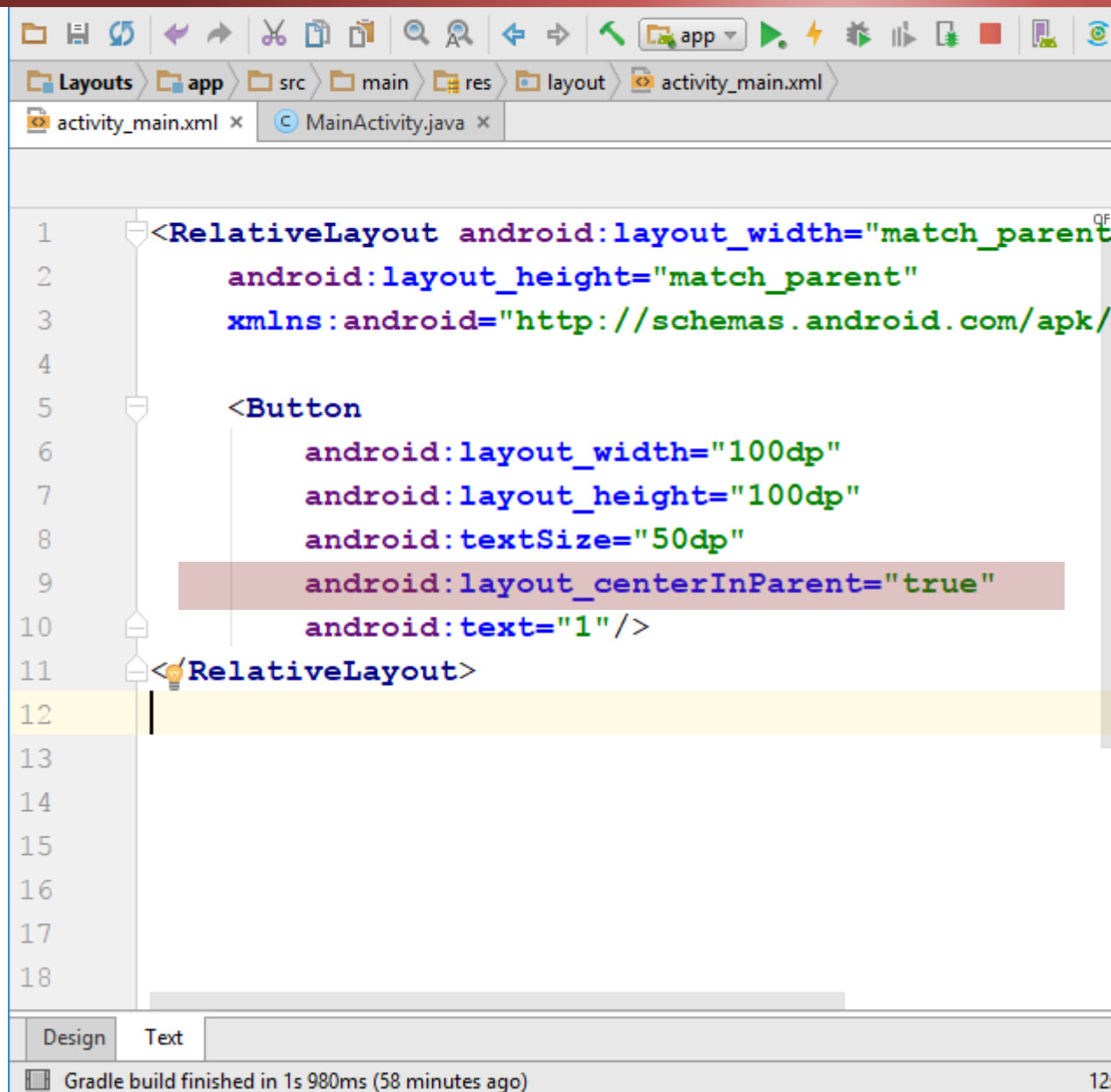
RelativeLayout

`layout_centerVertical` combinado con `layout_centerHorizontal` central al **View** con respecto a los límites de su contenedor

```
1 <RelativeLayout android:layout_width="match_parent"
2   android:layout_height="match_parent"
3   xmlns:android="http://schemas.android.com/apk/
4
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 </RelativeLayout>
```



RelativeLayout



```
1 <RelativeLayout android:layout_width="match_parent"
2     android:layout_height="match_parent"
3     xmlns:android="http://schemas.android.com/apk/
4
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 </RelativeLayout>
```

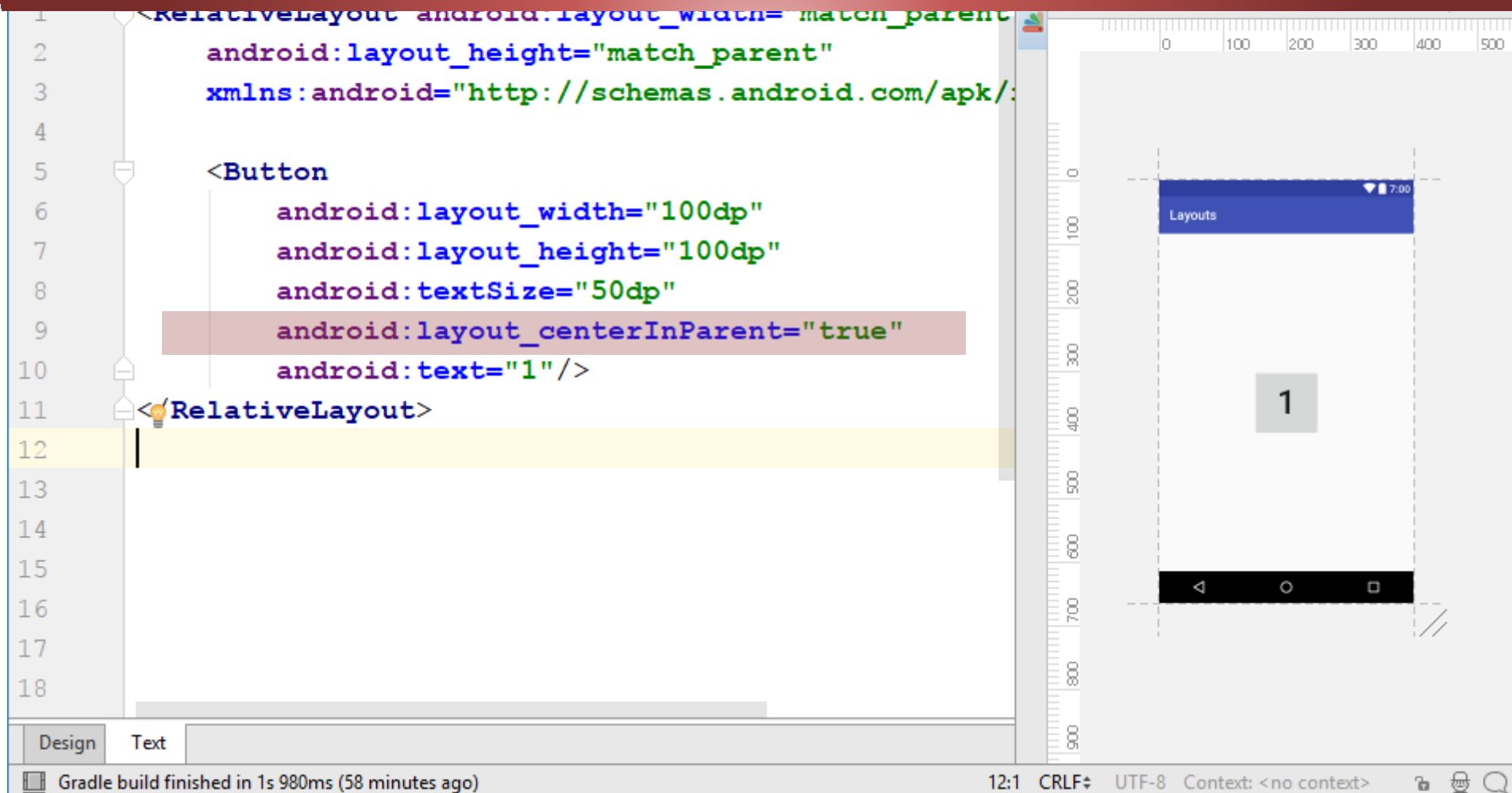
Design Text

Gradle build finished in 1s 980ms (58 minutes ago)

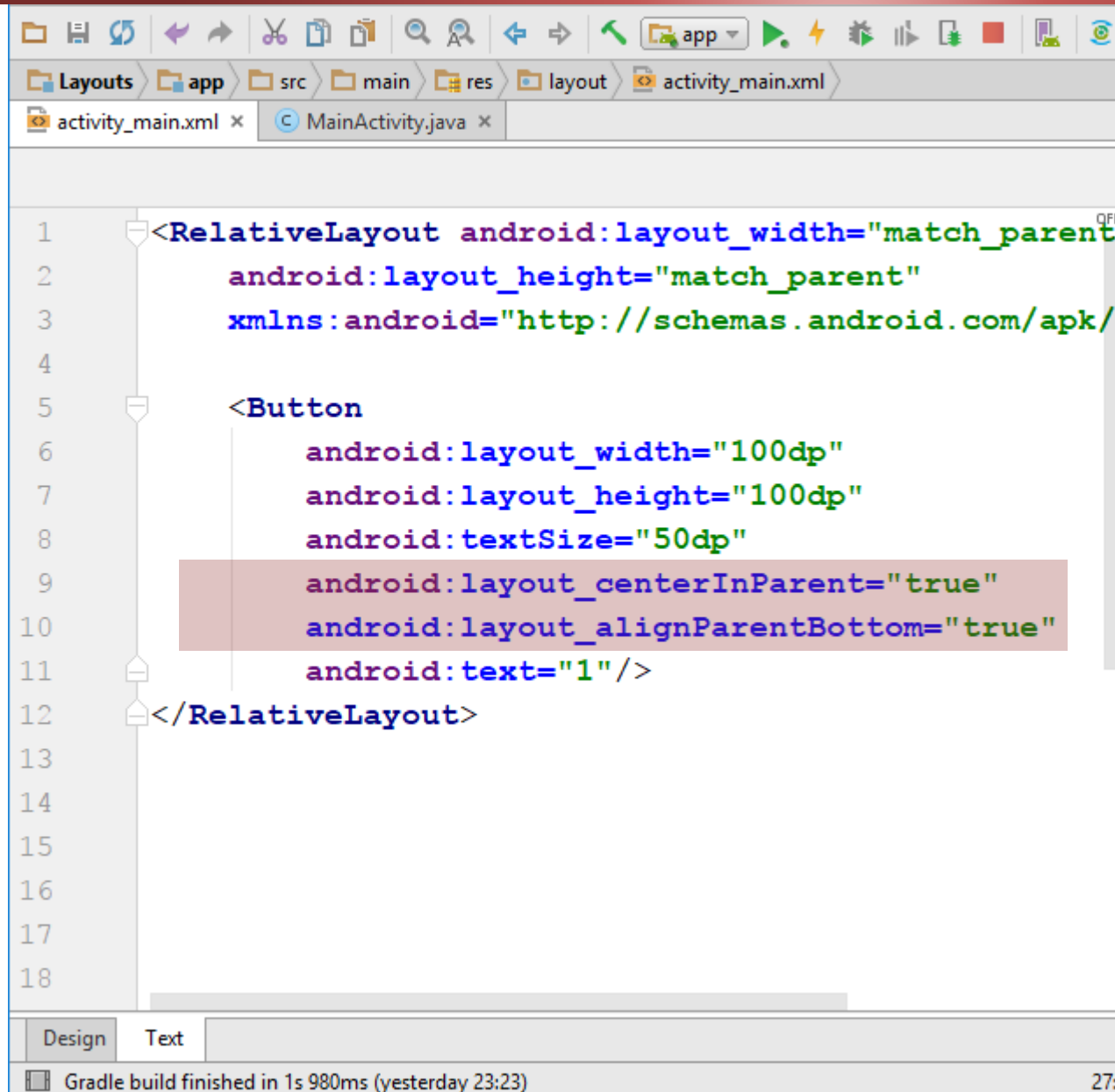
Probar el
resultado de la
siguiente
disposición

RelativeLayout

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



RelativeLayout



Probar el
resultado de la
siguiente
disposición

RelativeLayout

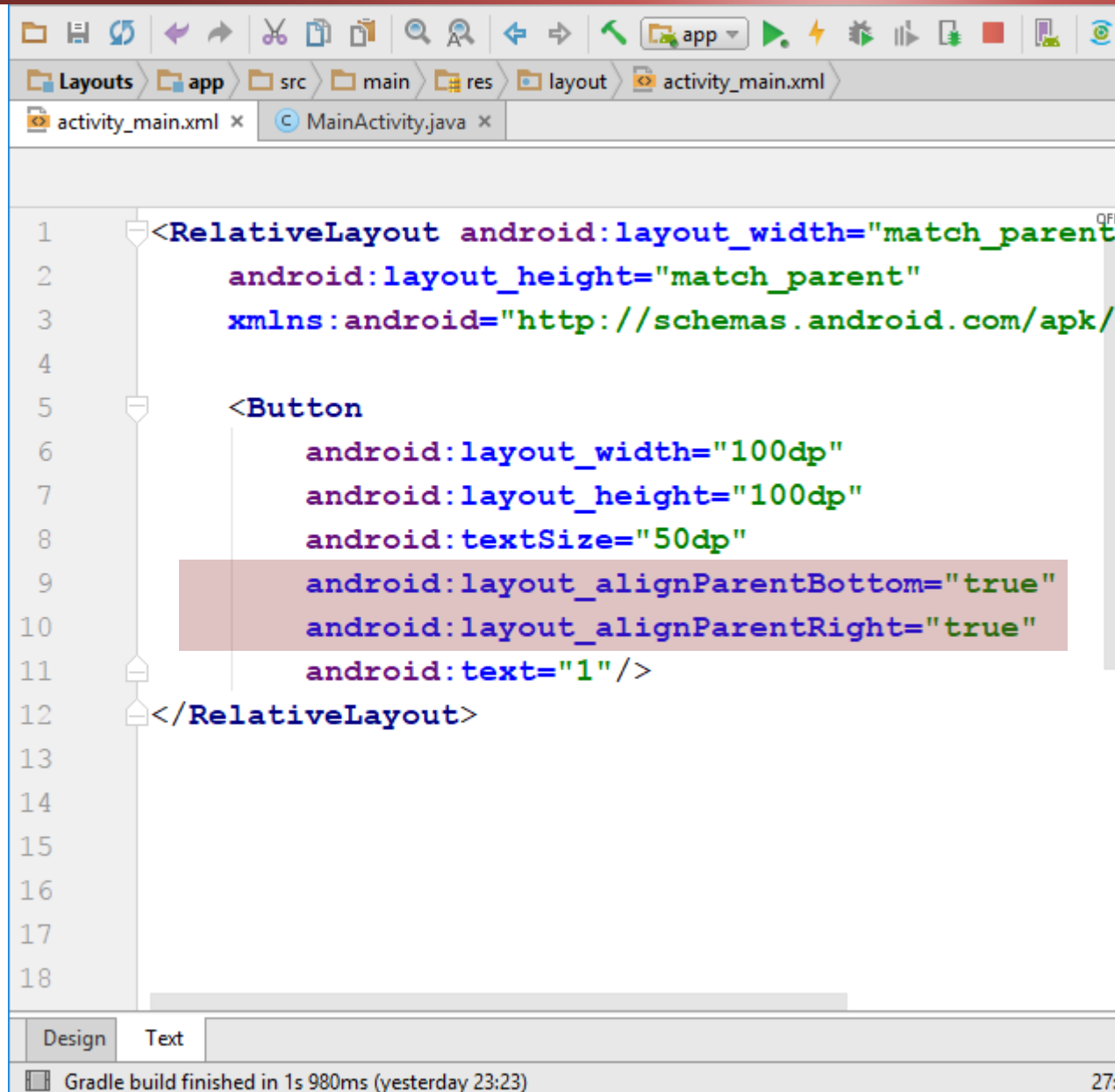
layout_alignParentBottom: Alinea el borde inferior del **View** con el borde inferior del contenedor

The screenshot displays the Android Studio IDE. On the left, the XML editor shows the following code:

```
1 <RelativeLayout android:layout_width="match_parent"
2   android:layout_height="match_parent"
3   xmlns:android="http://schemas.android.com/apk/
4
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 </RelativeLayout>
```

The line `android:layout_alignParentBottom="true"` is highlighted with a red background. On the right, the Design tab shows a visual representation of the layout on a mobile device. A blue header bar at the top contains the text "Layouts". A large grey rectangle with the number "1" is positioned at the bottom of the screen, aligned with the bottom edge of the device frame. The status bar at the very bottom shows the time as 7:00. The bottom of the screen features a navigation bar with three icons: a back arrow, a circle, and a square. The status bar at the bottom of the IDE shows "Gradle build finished in 1s 980ms (yesterday 23:23)", "27:1 CRLF UTF-8 Context: <no context>", and icons for a folder, a bug, and a comment.

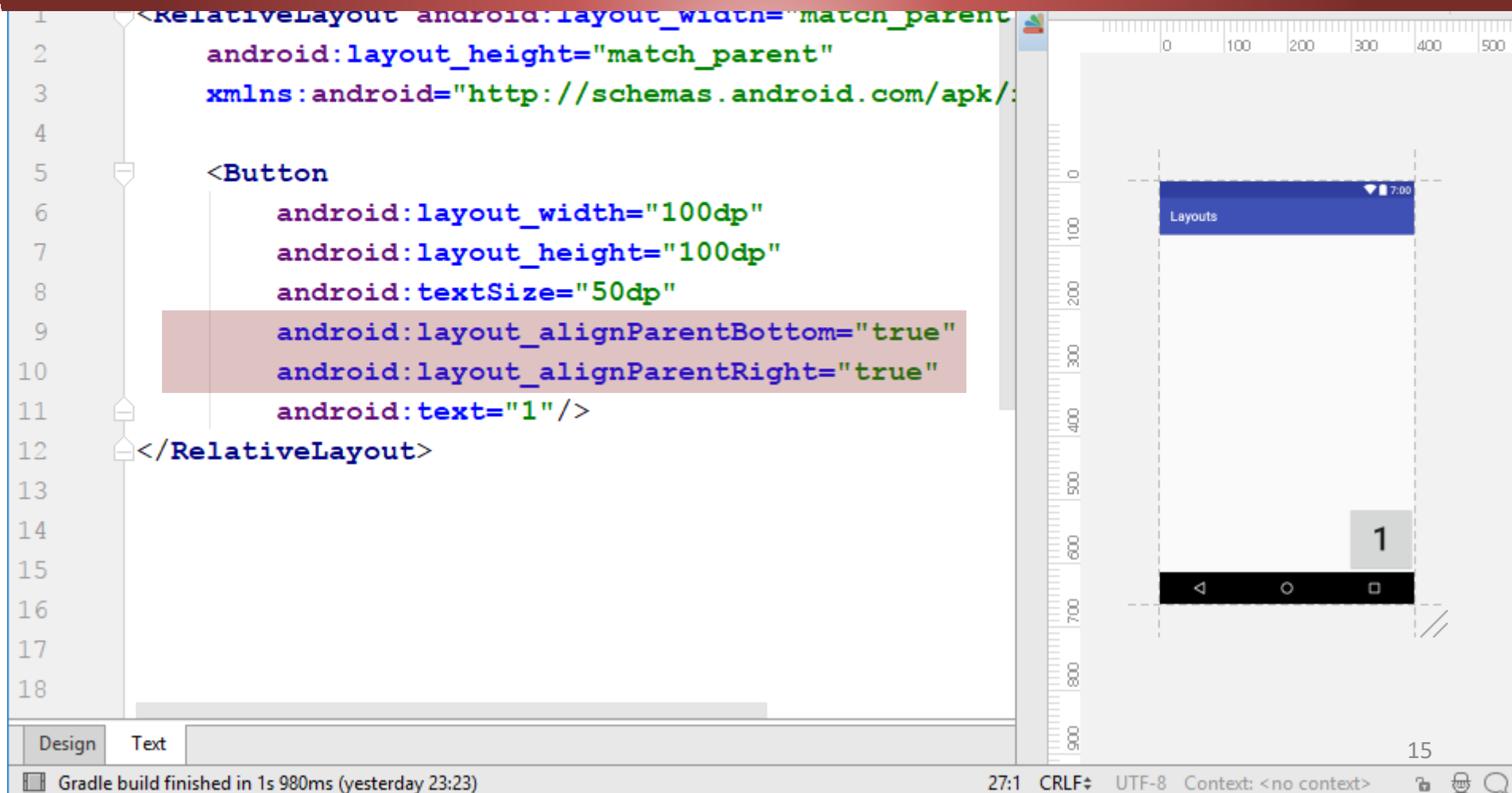
RelativeLayout



Probar el
resultado de la
siguiente
disposición

RelativeLayout

layout_alignParentRight: Alinea el borde derecho del **View** con el borde derecho de su contenedor



The screenshot displays the Android Studio interface. On the left, the XML code for a RelativeLayout is shown in the 'Text' tab. The code defines a RelativeLayout with a width and height of 'match_parent'. Inside, a Button is defined with a width and height of '100dp', a text size of '50dp', and is aligned to the bottom and right of the parent. The 'Design' tab on the right shows a visual representation of this layout on a mobile device screen. The screen has a blue header bar with the text 'Layouts' and a status bar at the top showing the time as 7:00. A large grey rectangle with the number '1' is positioned in the bottom right corner of the screen, demonstrating the effect of the `layout_alignParentRight="true"` attribute. The bottom of the screen shows the Android navigation bar with back, home, and recent apps buttons. The status bar at the very bottom indicates the build status: 'Gradle build finished in 1s 980ms (yesterday 23:23)'. The bottom right corner of the interface shows the page number '15' and various system icons.

```
1 <RelativeLayout android:layout_width="match_parent"
2   android:layout_height="match_parent"
3   xmlns:android="http://schemas.android.com/apk/
4
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 </RelativeLayout>
```

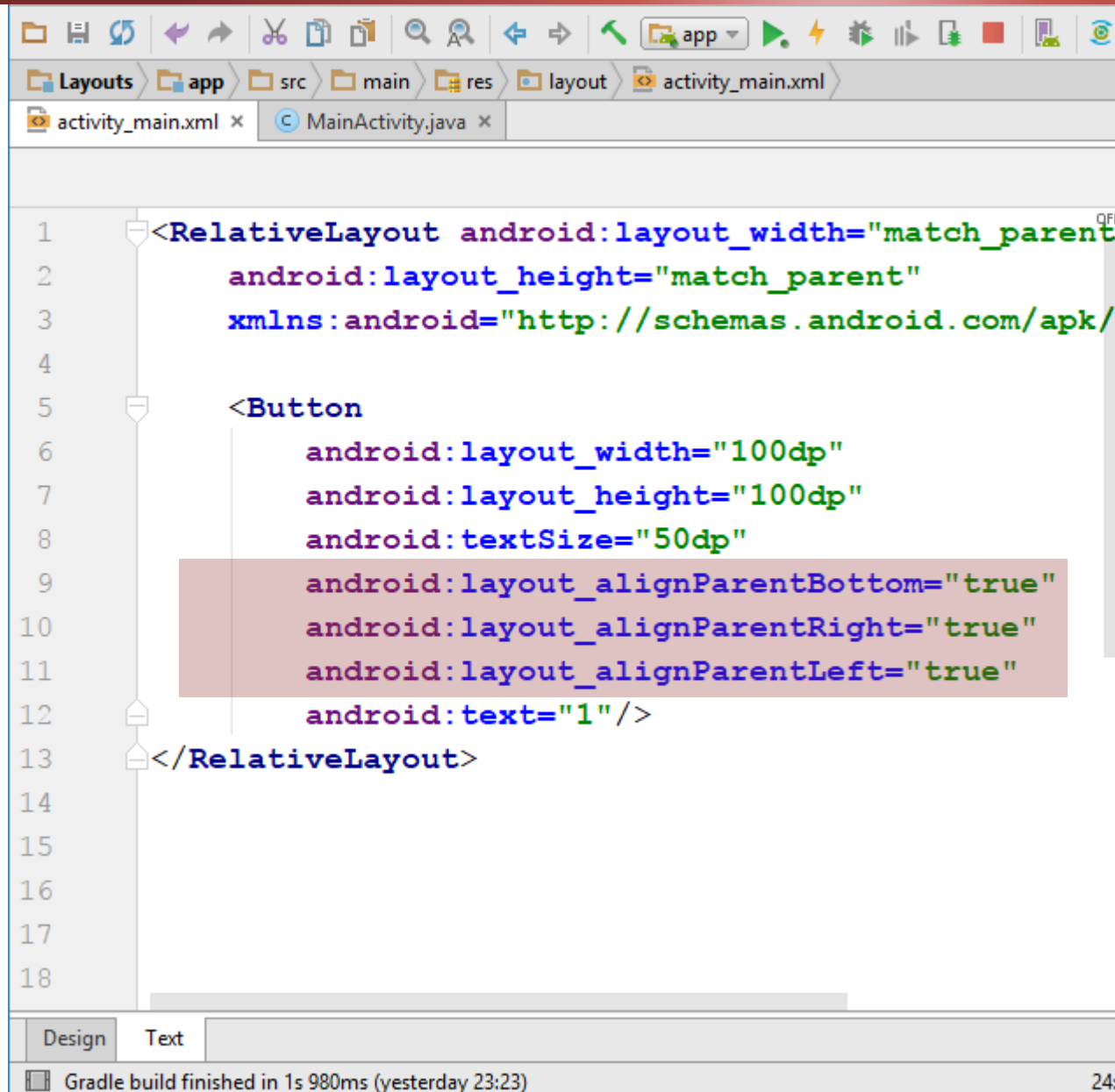
Design | Text

Gradle build finished in 1s 980ms (yesterday 23:23)

27:1 CRLF UTF-8 Context: <no context>

15

RelativeLayout



Probar el
resultado de la
siguiente
disposición

RelativeLayout

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

The screenshot displays the Android Studio IDE. On the left, the XML code for a `RelativeLayout` is shown in the 'Text' tab. The code defines a `RelativeLayout` with a width and height of `match_parent`. Inside, a `Button` is defined with a width and height of `100dp` and a text size of `50dp`. The `Button` is positioned at the bottom-right corner of the `RelativeLayout` using `layout_alignParentBottom="true"`, `layout_alignParentRight="true"`, and `layout_alignParentLeft="true"`. The `Button` text is set to `"1"`. The right side of the screen shows the 'Design' tab, which visualizes the layout on a mobile device screen. The `Button` is shown as a grey rectangle with the number '1' in the center, aligned to the bottom-right of the screen. The status bar at the top of the preview shows the time as 7:00. The bottom of the screen shows the 'Gradle build finished' message and the system tray with the time 24:1, encoding UTF-8, and context '<no context>'.

```
1 <RelativeLayout android:layout_width="match_parent"
2   android:layout_height="match_parent"
3   xmlns:android="http://schemas.android.com/apk/
4
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:text="1"/>
13 </RelativeLayout>
14
15
16
17
18
```

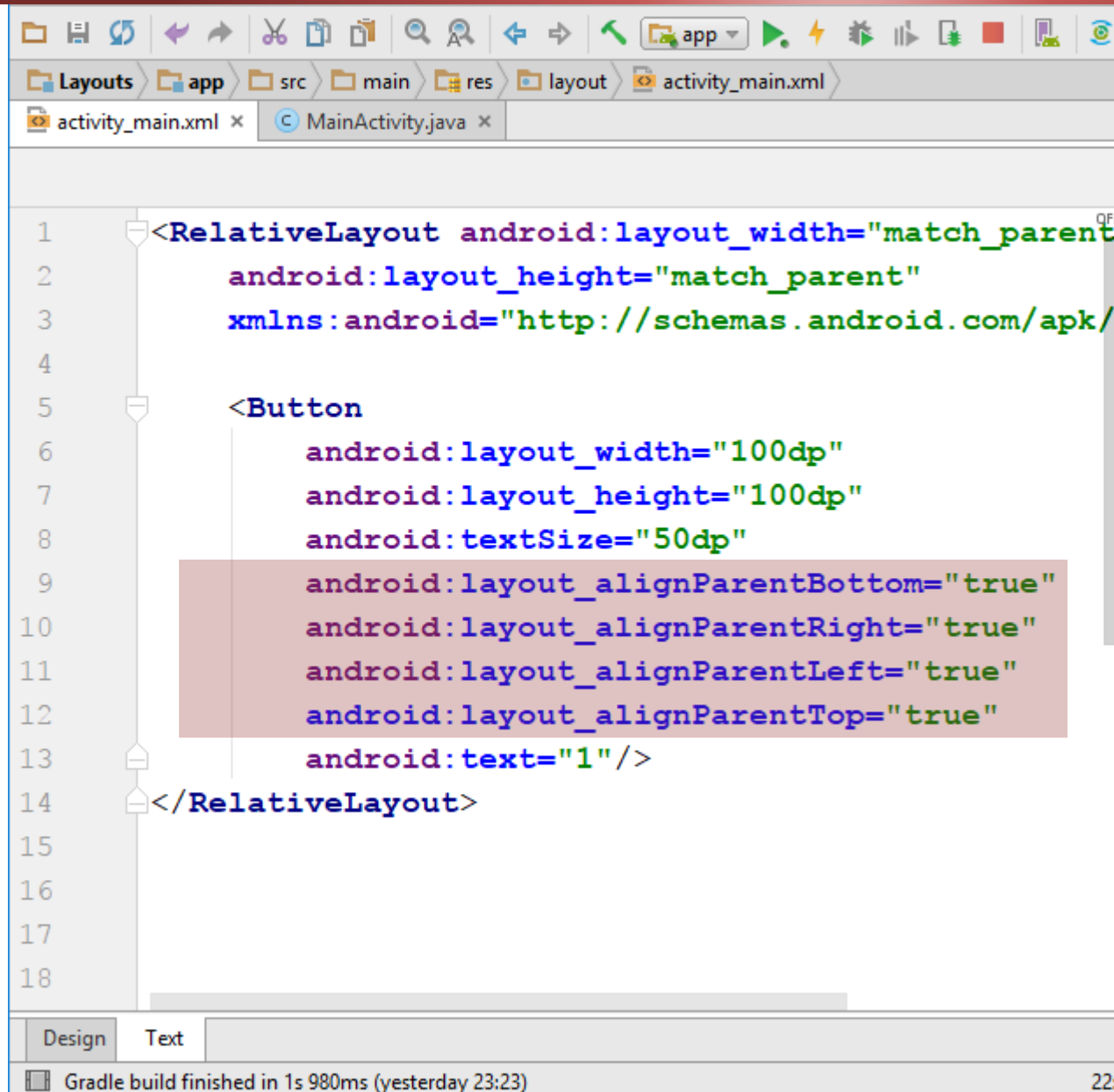
Design Text

Gradle build finished in 1s 980ms (yesterday 23:23)

24:1 CRLF UTF-8 Context: <no context>

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



```
1 <RelativeLayout android:layout_width="match_parent"
2   android:layout_height="match_parent"
3   xmlns:android="http://schemas.android.com/apk/
4
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 </RelativeLayout>
15
16
17
18
```

Design Text

Gradle build finished in 1s 980ms (yesterday 23:23)

Probar el
resultado de la
siguiente
disposición

RelativeLayout

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

The screenshot displays the Android Studio interface. On the left, the XML code for a `RelativeLayout` is shown. A `Button` is defined with the following attributes:

```
<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_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:text="1"/>

</RelativeLayout>
```

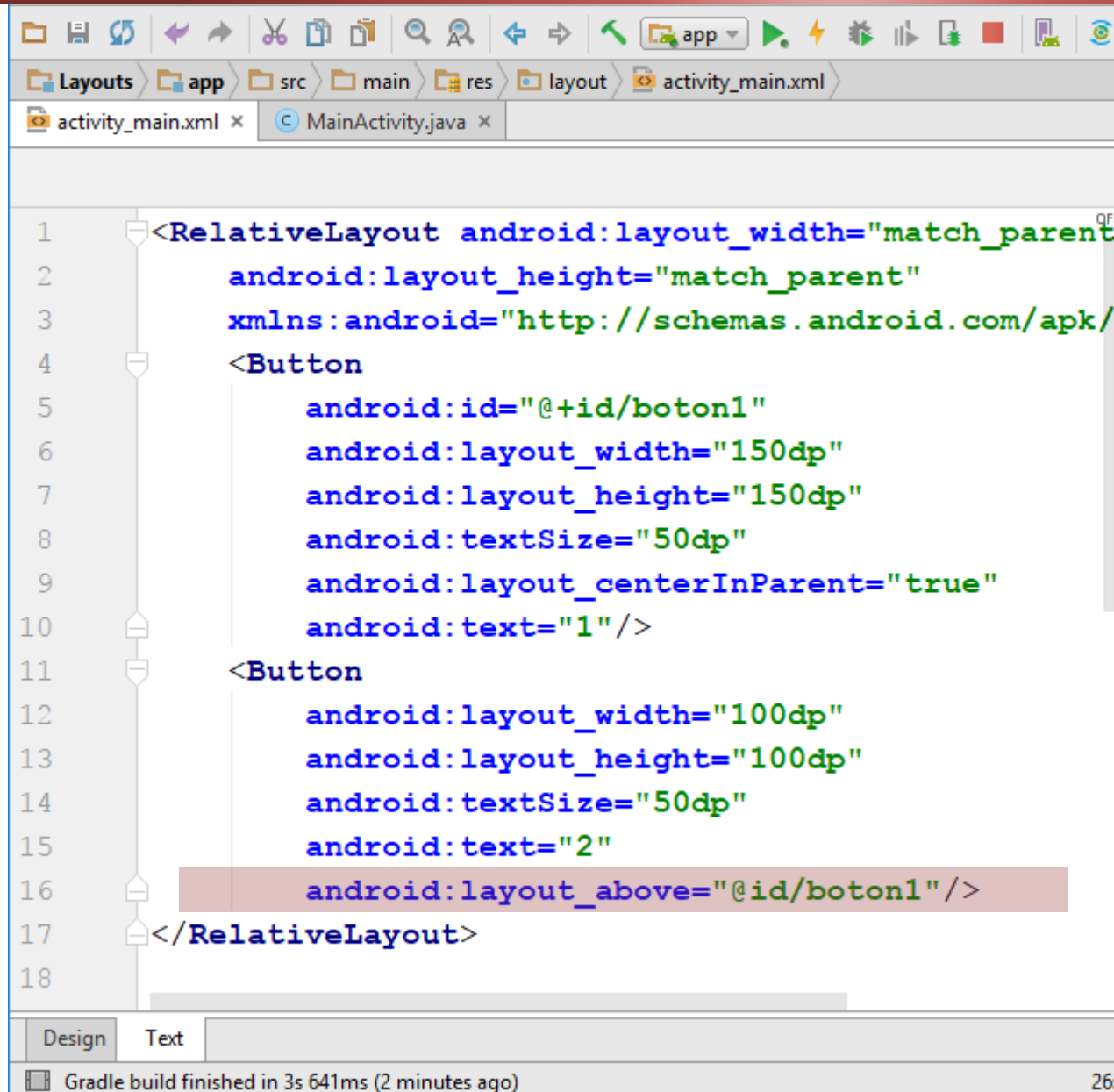
The attributes `android:layout_alignParentBottom="true"`, `android:layout_alignParentRight="true"`, `android:layout_alignParentLeft="true"`, and `android:layout_alignParentTop="true"` are highlighted with a red box, indicating that the button is aligned to all four edges of the parent container.

On the right, the 'Design' tab shows a visual representation of the layout. It features a large gray rectangle representing the `RelativeLayout` container. A smaller blue rectangle, representing the `Button`, is positioned at the top center of the container. The button contains the text '1'. The interface includes a ruler at the top and bottom, and a status bar at the bottom showing the time as 7:00.

At the bottom of the screen, the status bar indicates the following information:

- Gradle build finished in 1s 980ms (yesterday 23:23)
- 22:1 CRLF UTF-8 Context: <no context>
- Page number: 20

RelativeLayout



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

Design Text

Gradle build finished in 3s 641ms (2 minutes ago)

Probar el
resultado de la
siguiente
disposición

RelativeLayout

layout_above: Alinea el borde inferior de un **View** con el borde superior de otro **View**

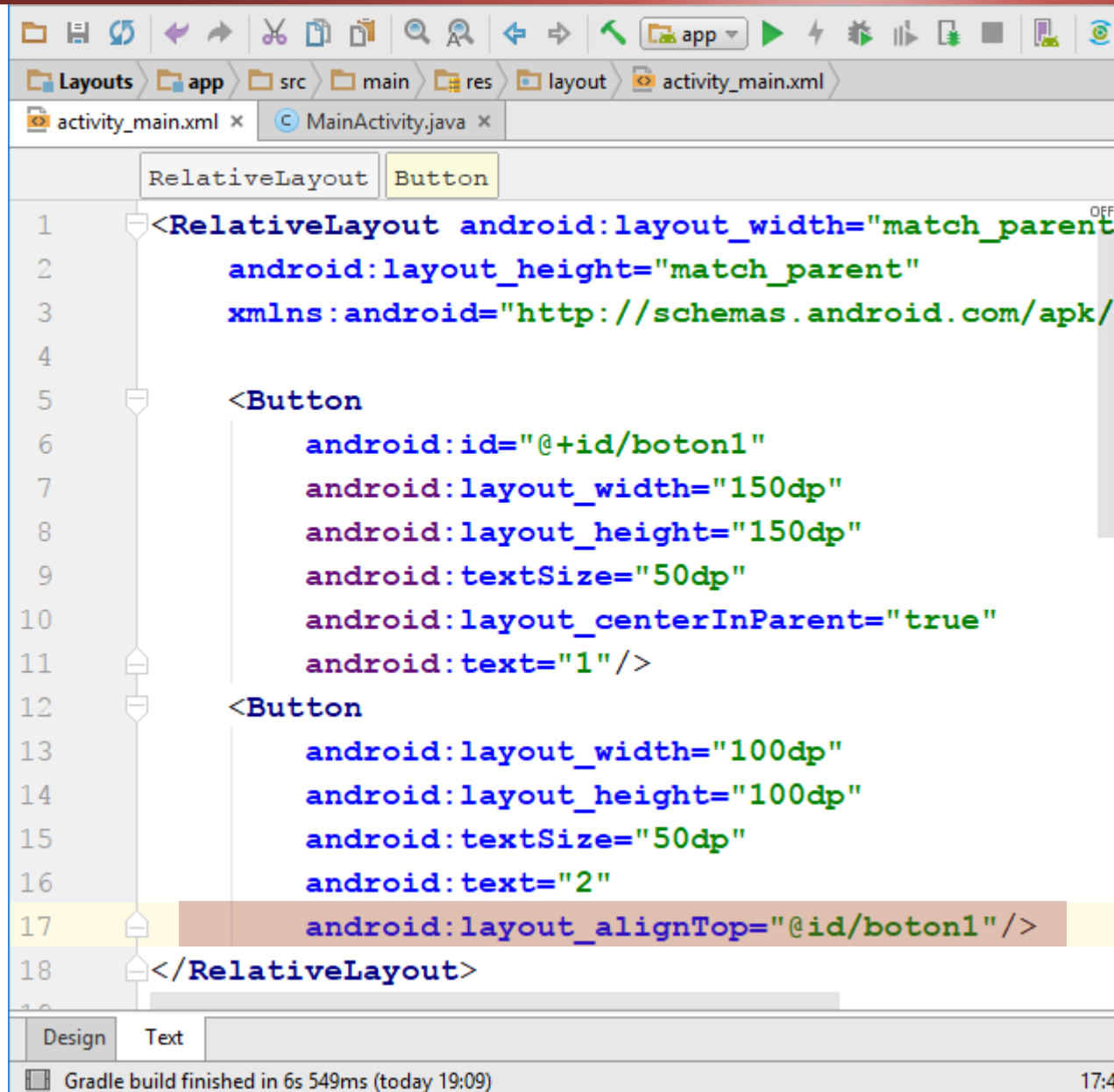
The screenshot displays the Android Studio IDE. On the left, the XML code for a `RelativeLayout` is shown in the 'Text' tab. The code defines two buttons: '1' and '2'. Button '2' is centered in the parent layout. Button '1' is positioned below button '2' using the `layout_above` attribute.

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

On the right, the 'Design' tab shows a visual representation of the layout on a mobile device screen. Two gray rectangular boxes represent the buttons. Box '2' is centered at the top, and box '1' is positioned directly below it. Dashed lines and arrows indicate the alignment and positioning of the views within the container.

At the bottom of the screen, the status bar shows 'Gradle build finished in 3s 641ms (2 minutes ago)', the time '26:1', and the encoding 'UTF-8'. The page number '22' is visible in the bottom right corner.

RelativeLayout



Probar el
resultado de la
siguiente
disposición

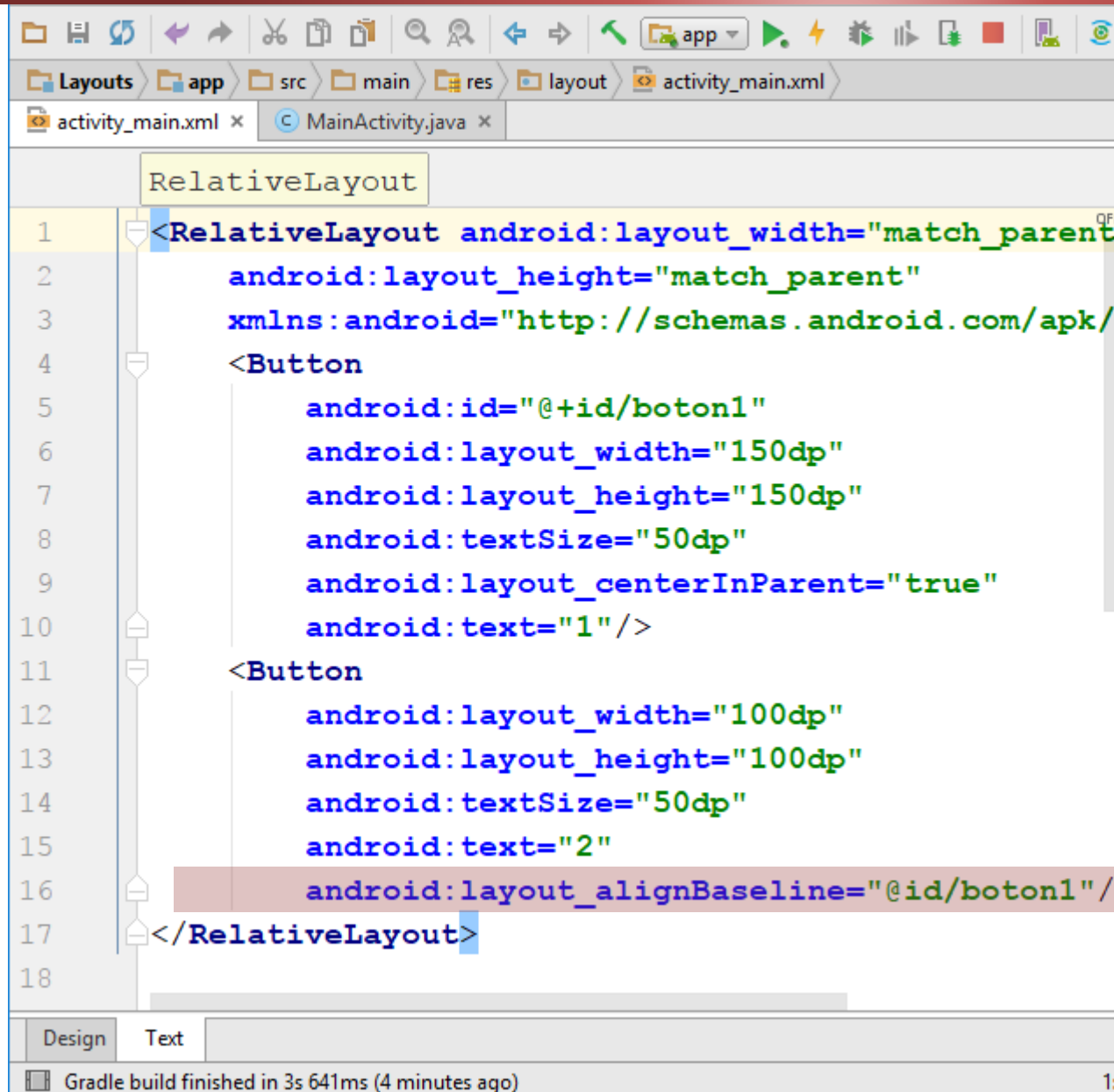
RelativeLayout

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

The screenshot displays the Android Studio IDE. On the left, the XML code for a `RelativeLayout` is shown. It contains two `Button` elements. The first button, with ID `@+id/boton1`, is centered and has a width and height of `150dp` and a text size of `50dp`. The second button is positioned below the first and has a width and height of `100dp` and a text size of `50dp`. The line `android:layout_alignTop="@id/boton1"/>` for the second button is highlighted in yellow, indicating it aligns its top edge with the top edge of the first button. On the right, the design preview shows a mobile screen with a blue header labeled 'Layouts'. Below the header, two gray squares are displayed: square '2' is on the left and square '1' is on the right. Square '1' is positioned directly below square '2', demonstrating the `layout_alignTop` property. The bottom of the screen shows the Android navigation bar. The status bar at the top indicates a time of 7:00. The bottom status bar shows 'Gradle build finished in 6s 549ms (today 19:09)', the time '17:44', and encoding 'UTF-8'.

```
1 <RelativeLayout android:layout_width="match_parent"
2   android:layout_height="match_parent"
3   xmlns:android="http://schemas.android.com/apk/
4
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 </RelativeLayout>
```


RelativeLayout



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

Design Text

Gradle build finished in 3s 641ms (4 minutes ago)

Probar el
resultado de la
siguiente
disposición

RelativeLayout

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

The screenshot displays the Android Studio IDE. On the left, the XML editor shows the following code:

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

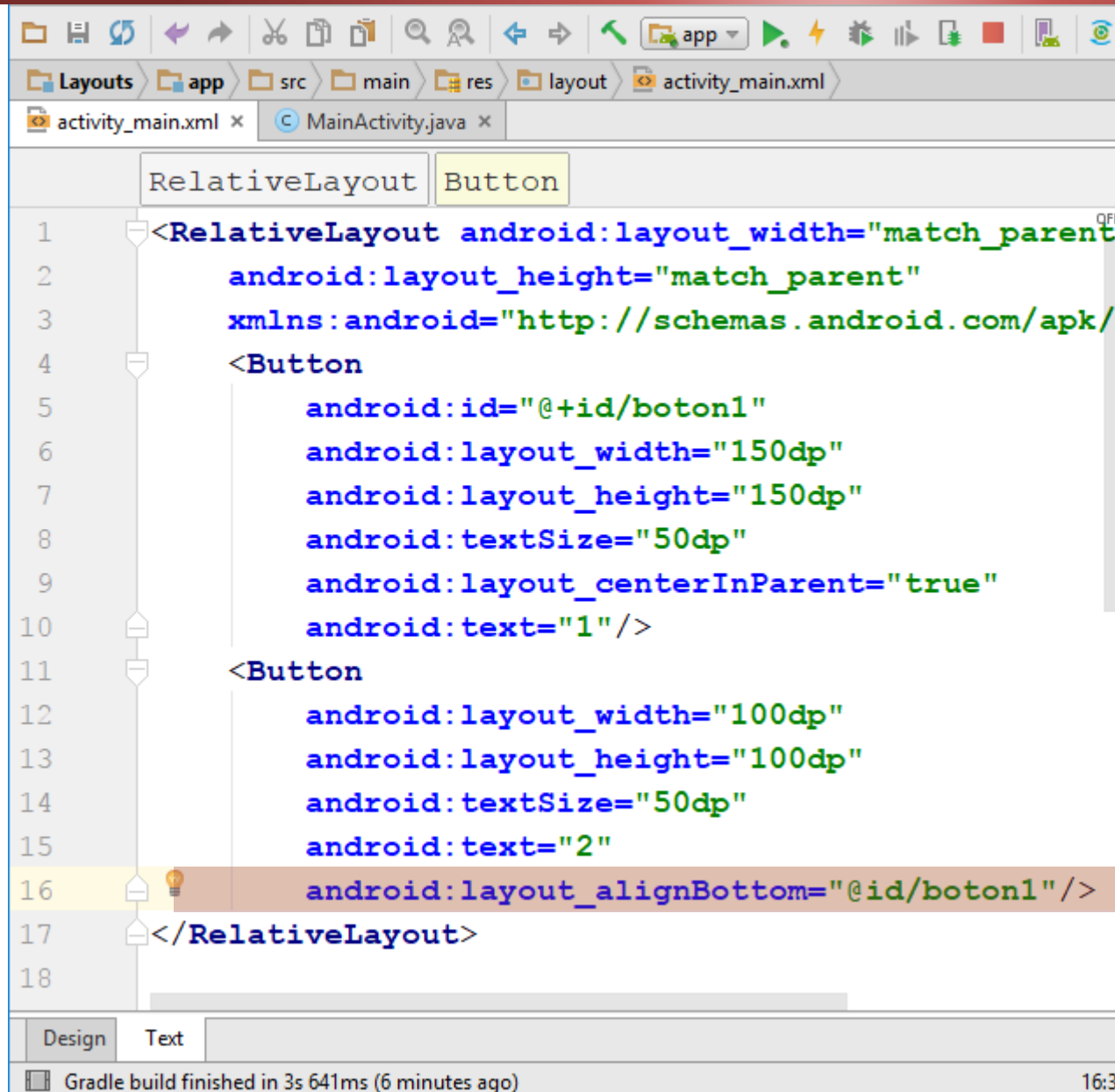
On the right, the Design view shows a visual representation of the layout. It features a blue header bar labeled "Layouts" at the top. Below it, two gray rectangular buttons are positioned. Button "1" is centered horizontally and vertically. Button "2" is positioned to the left of button "1", and its baseline is aligned with the baseline of button "1", as indicated by a horizontal dashed line. A black navigation bar is visible at the bottom of the design area. The interface includes a ruler at the top and a status bar at the bottom.

26

Gradle build finished in 3s 641ms (4 minutes ago)

1:1 CRLF UTF-8 Context: <no context>

RelativeLayout



Probar el
resultado de la
siguiente
disposición

RelativeLayout

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

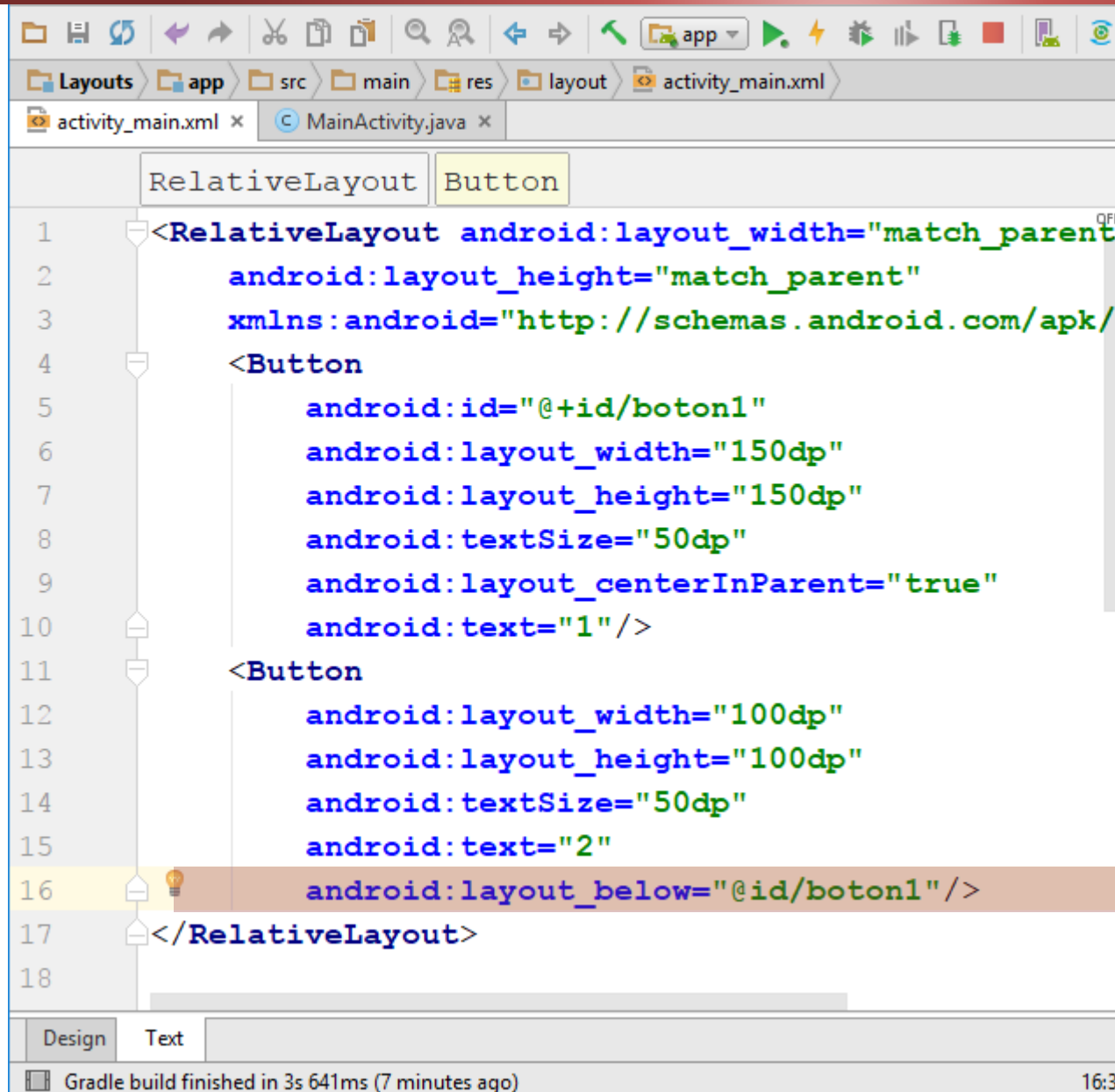
The screenshot displays the Android Studio IDE. On the left, the XML editor shows the layout definition for a `RelativeLayout`. It contains two `Button` elements. The first button, with ID `@+id/boton1`, is centered and has a height of 150dp. The second button is positioned below it and is aligned to the bottom of the first button using the `layout_alignBottom` attribute. On the right, the Design tab shows a visual representation of this layout on a mobile device screen. Two gray rectangular buttons are shown: the top one is labeled '1' and the bottom one is labeled '2'. A dashed vertical line indicates that the bottom edge of button '2' is aligned with the bottom edge of button '1'. The status bar at the top of the preview shows the time as 7:00. At the bottom of the screen, a status bar indicates the system time as 16:37 and provides information about the build process and encoding.

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

Design | Text

Gradle build finished in 3s 641ms (6 minutes ago) 16:37 CRLF UTF-8 Context: <no context>

RelativeLayout

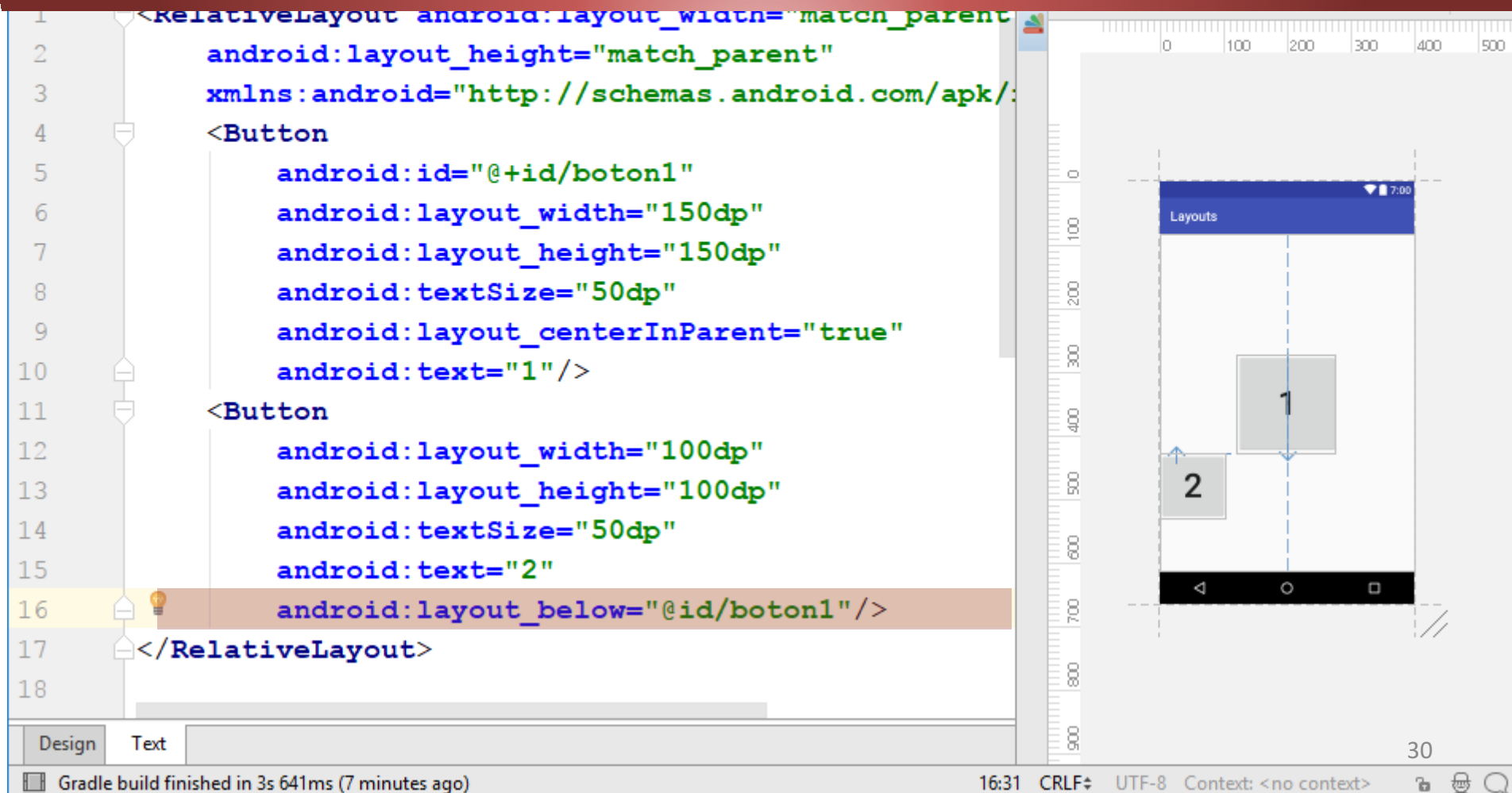


Probar el
resultado de la
siguiente
disposición

RelativeLayout

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

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

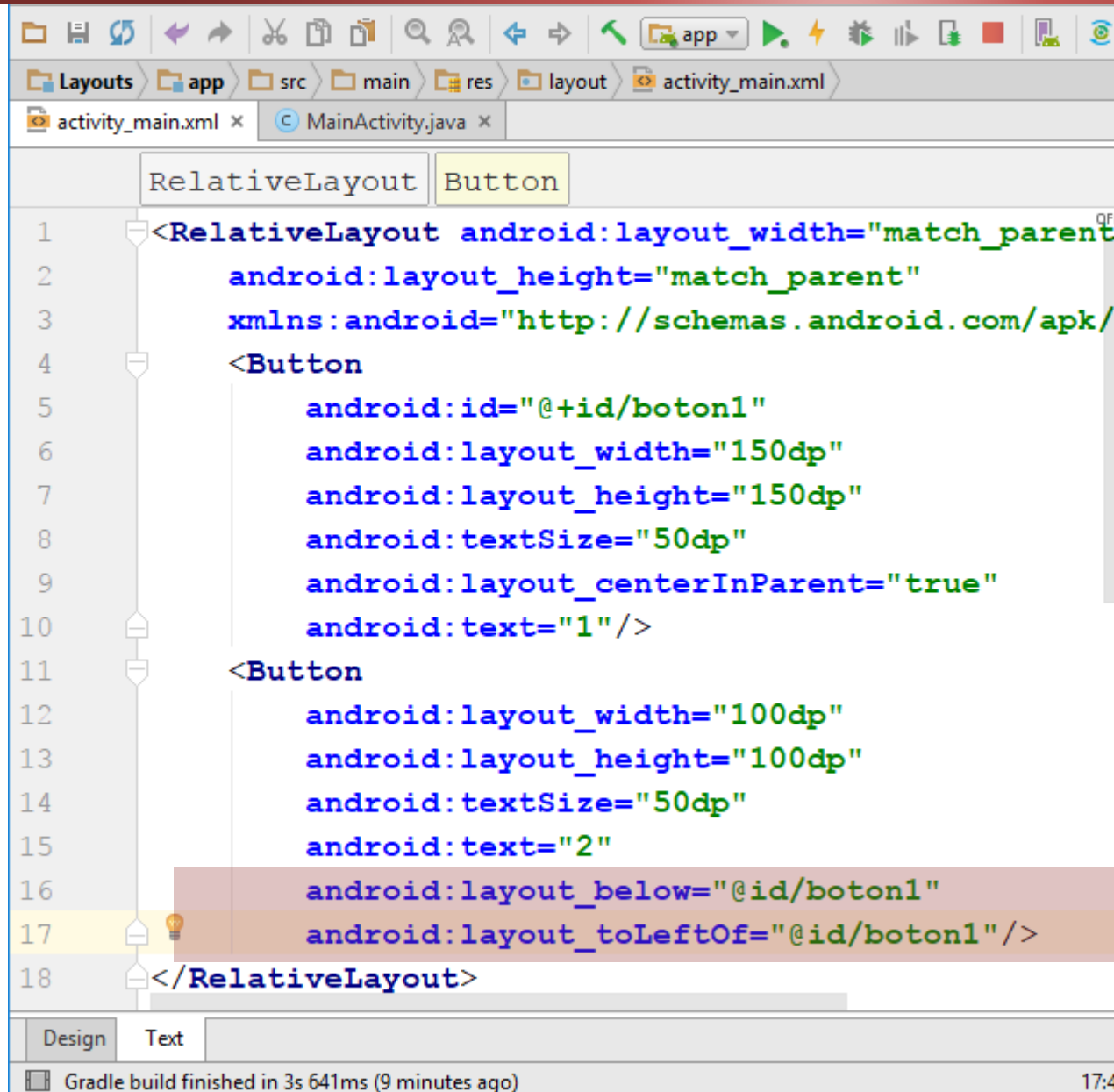


The image shows the Android Studio interface. On the left, the XML code for a `RelativeLayout` is displayed. It contains two `Button` elements. The first button, with ID `@+id/boton1`, is centered in the parent layout. The second button is positioned below the first, as indicated by the `android:layout_below="@id/boton1"` attribute. On the right, the Design view shows a visual representation of this layout on a mobile device screen. Two gray squares, labeled '1' and '2', represent the buttons. Square '1' is centered, and square '2' is positioned directly below it, demonstrating the `layout_below` property.

Design Text

Gradle build finished in 3s 641ms (7 minutes ago) 16:31 CRLF UTF-8 Context: <no context>

RelativeLayout



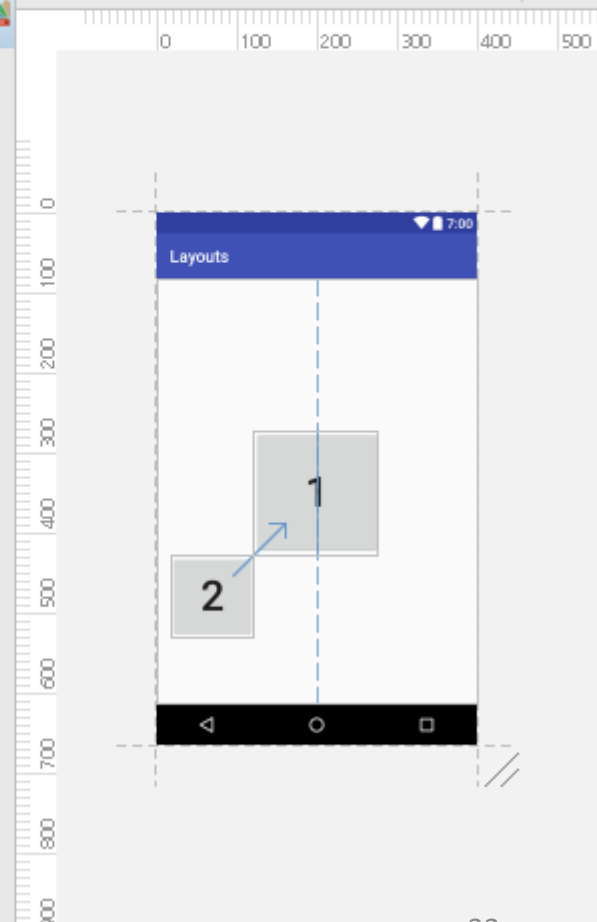
```
1 <RelativeLayout android:layout_width="match_parent"
2   android:layout_height="match_parent"
3   xmlns:android="http://schemas.android.com/apk/
4   <Button
5       android:id="@+id/boton1"
6       android:layout_width="150dp"
7       android:layout_height="150dp"
8       android:textSize="50dp"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

Probar el
resultado de la
siguiente
disposición

RelativeLayout

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

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

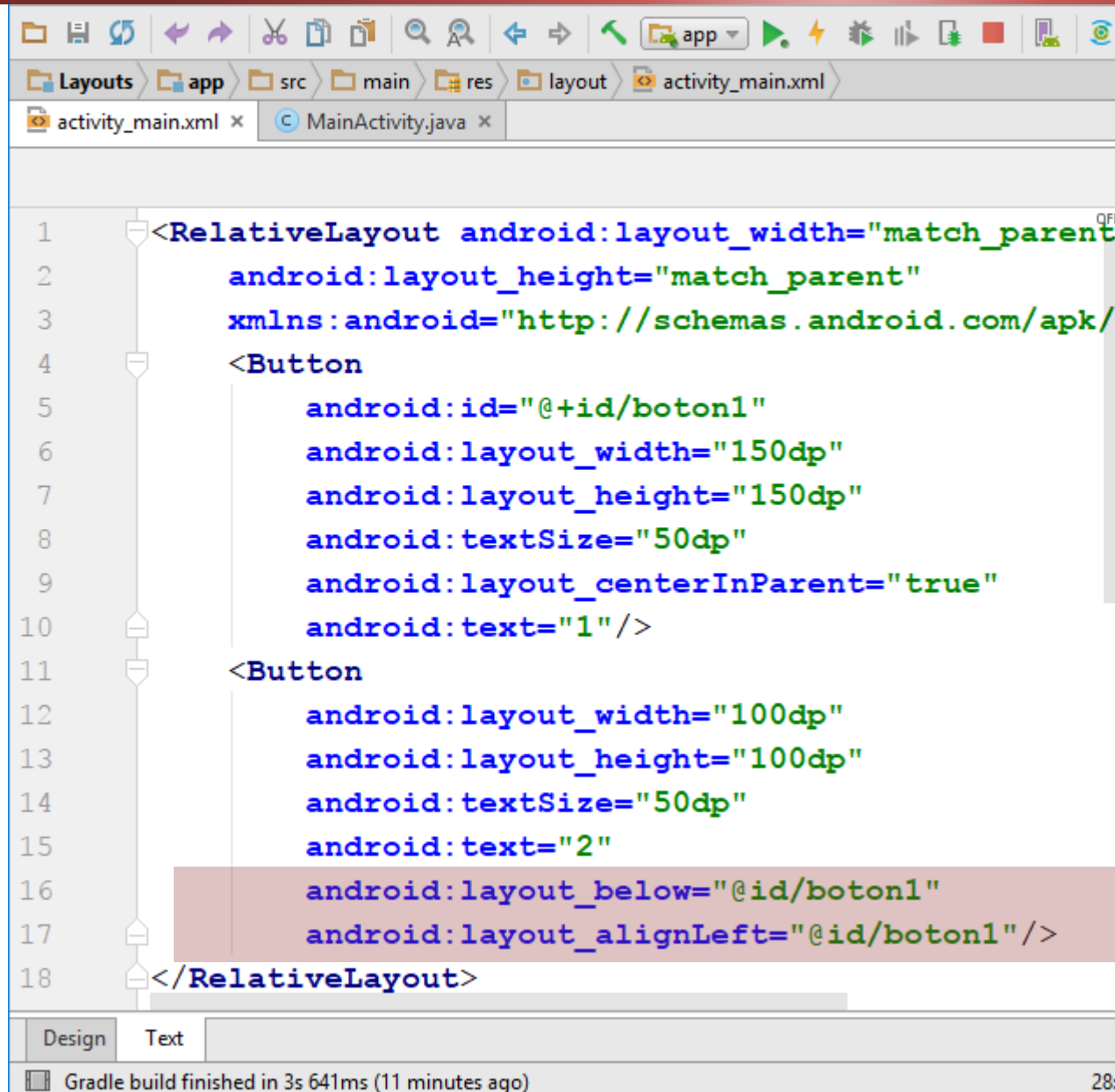


The diagram shows a mobile screen with a blue header labeled 'Layouts' and a black navigation bar at the bottom. Two gray squares are displayed: a larger square labeled '1' centered in the upper half, and a smaller square labeled '2' positioned to the left of square '1'. A dashed vertical line and an arrow indicate that the right edge of square '2' is aligned with the left edge of square '1', demonstrating the `layout_toLeftOf` property.

Design Text

Gradle build finished in 3s 641ms (9 minutes ago) 17:45 CRLF UTF-8 Context: <no context>

RelativeLayout



Probar el
resultado de la
siguiente
disposición

RelativeLayout

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

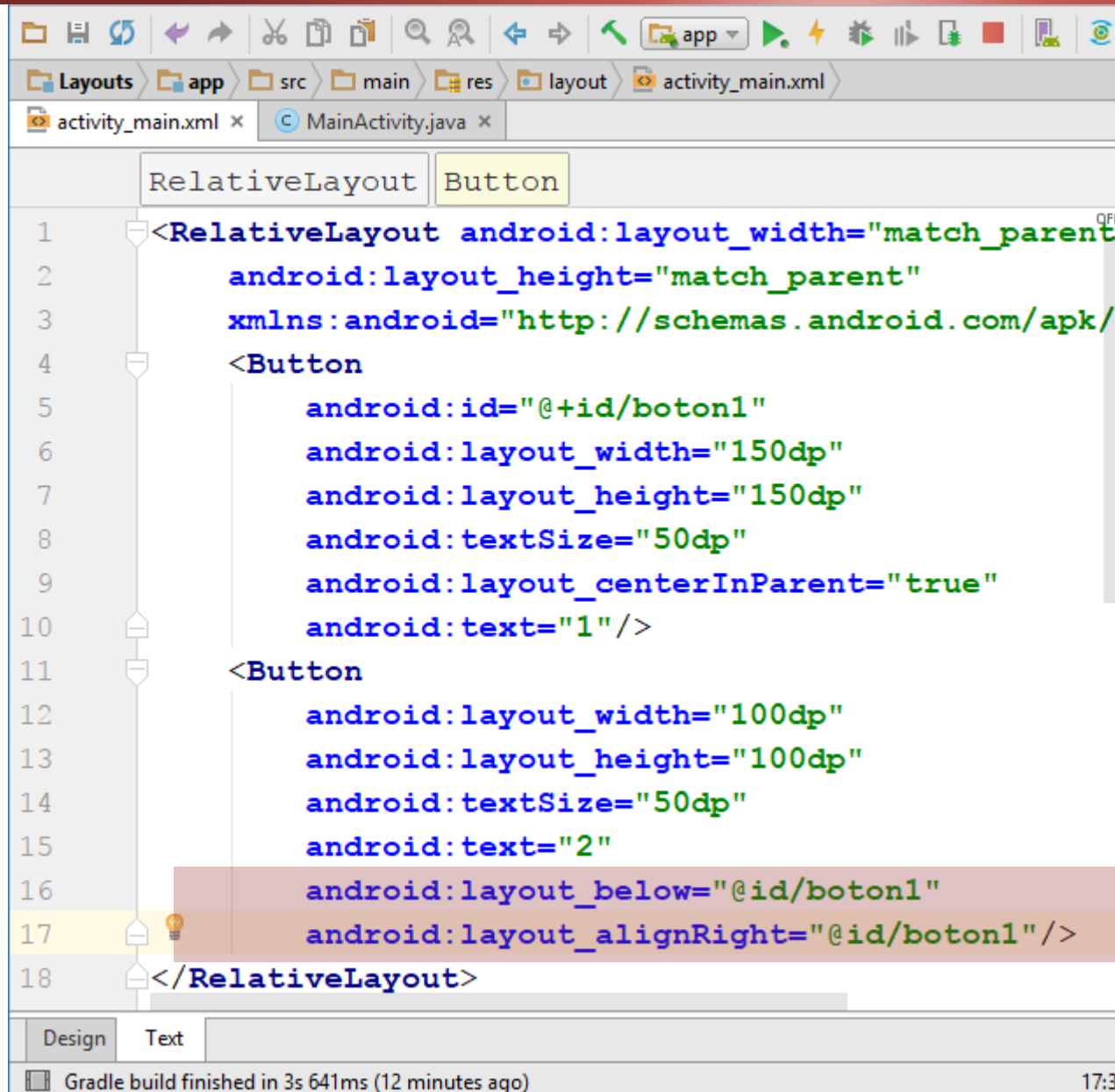
The screenshot displays the Android Studio IDE. On the left, the XML code for a `RelativeLayout` is shown in the 'Text' tab. The code defines two buttons: 'boton1' and '2'. Button '2' is positioned below and to the left of 'boton1' using the `layout_alignLeft` attribute. The 'Design' tab on the right shows a visual representation of this layout on a mobile device screen. Two gray rectangular buttons are visible: a larger one at the top and a smaller one below it, aligned to the left. A vertical dashed line indicates the alignment reference. The status bar at the bottom shows 'Gradle build finished in 3s 641ms (11 minutes ago)', the time '28:1', and the encoding 'UTF-8'.

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

34

Gradle build finished in 3s 641ms (11 minutes ago) 28:1 CRLF UTF-8 Context: <no context>

RelativeLayout



Probar el
resultado de la
siguiente
disposición

RelativeLayout

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

The screenshot displays the Android Studio IDE. On the left, the XML code for a `RelativeLayout` is shown. The code defines two buttons: `boton1` and `boton2`. `boton1` is centered in the parent layout. `boton2` is positioned below `boton1` and its right edge is aligned with the right edge of `boton1` using the `layout_alignRight` attribute.

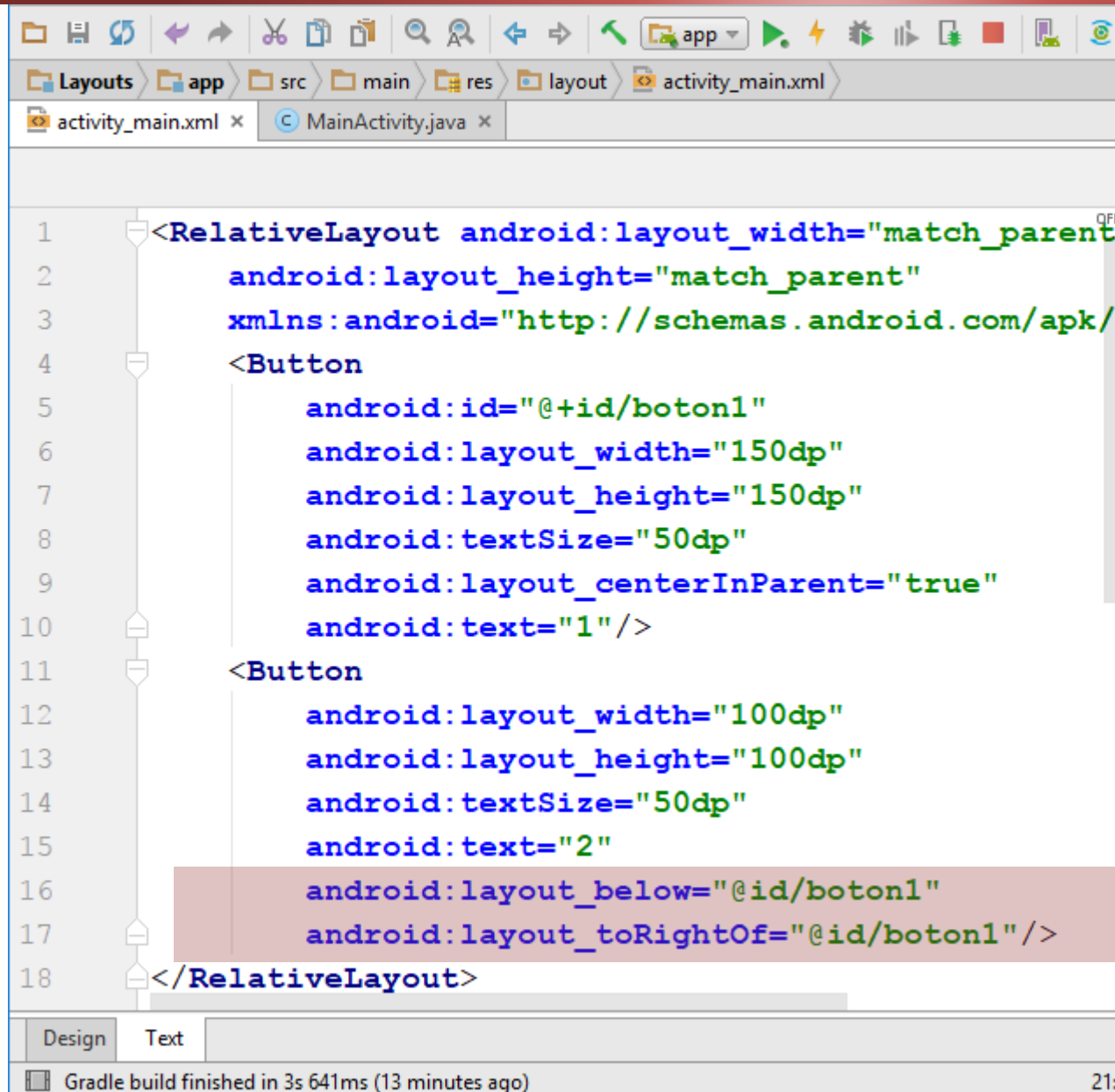
```
1 <RelativeLayout android:layout_width="match_parent"
2   android:layout_height="match_parent"
3   xmlns:android="http://schemas.android.com/apk/
4   <Button
5       android:id="@+id/boton1"
6       android:layout_width="150dp"
7       android:layout_height="150dp"
8       android:textSize="50dp"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

On the right, the design preview shows a mobile screen with two overlapping gray squares. The top square is labeled '1' and is centered. The bottom square is labeled '2' and is positioned below square 1, with its right edge aligned with the right edge of square 1. A vertical dashed line indicates this alignment. The interface includes a ruler at the top and bottom, and tabs for 'Design' and 'Text' at the bottom left.

36

Gradle build finished in 3s 641ms (12 minutes ago) 17:36 CRLF UTF-8 Context: <no context>

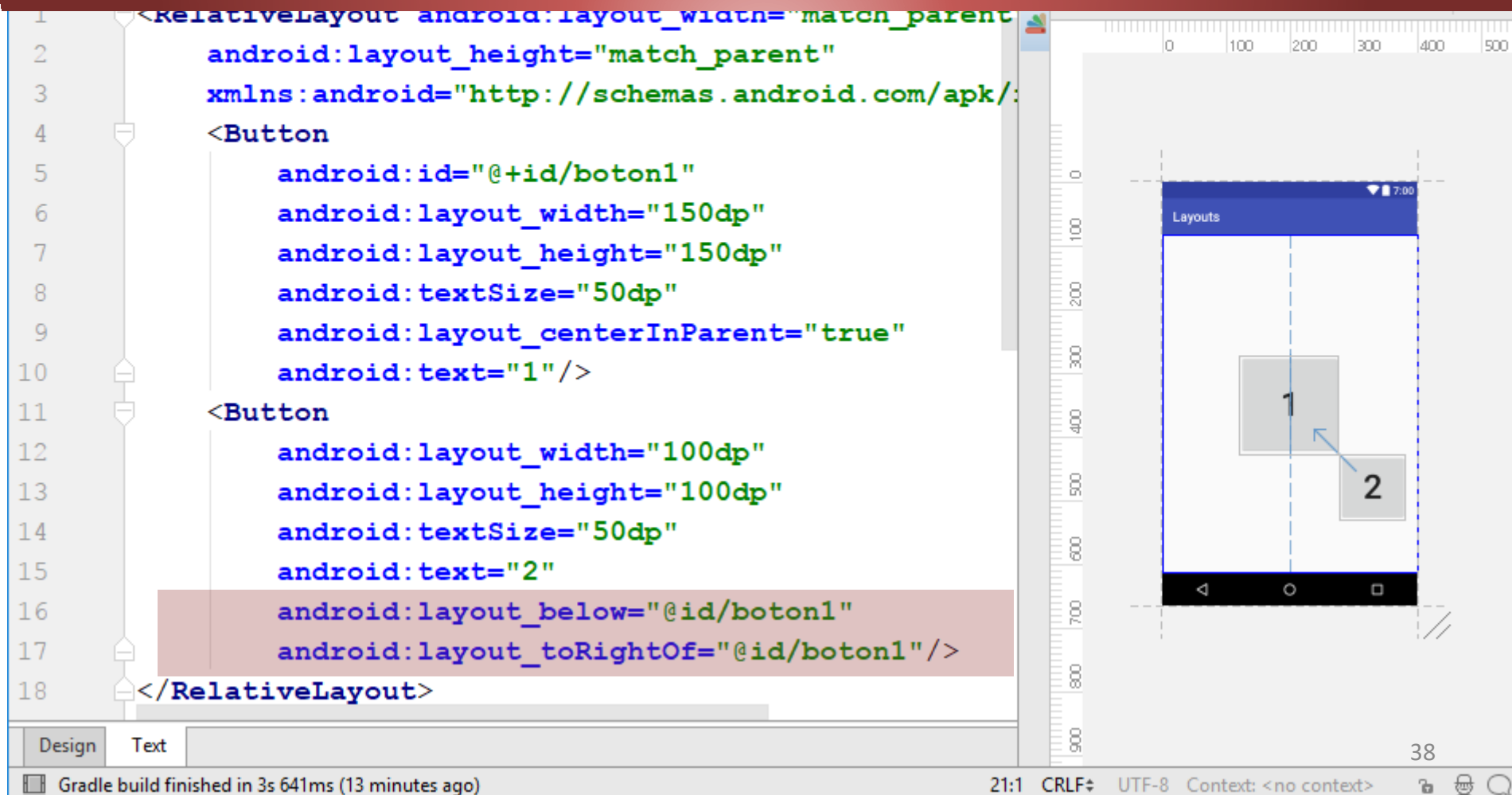
RelativeLayout



Probar el
resultado de la
siguiente
disposición

RelativeLayout

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



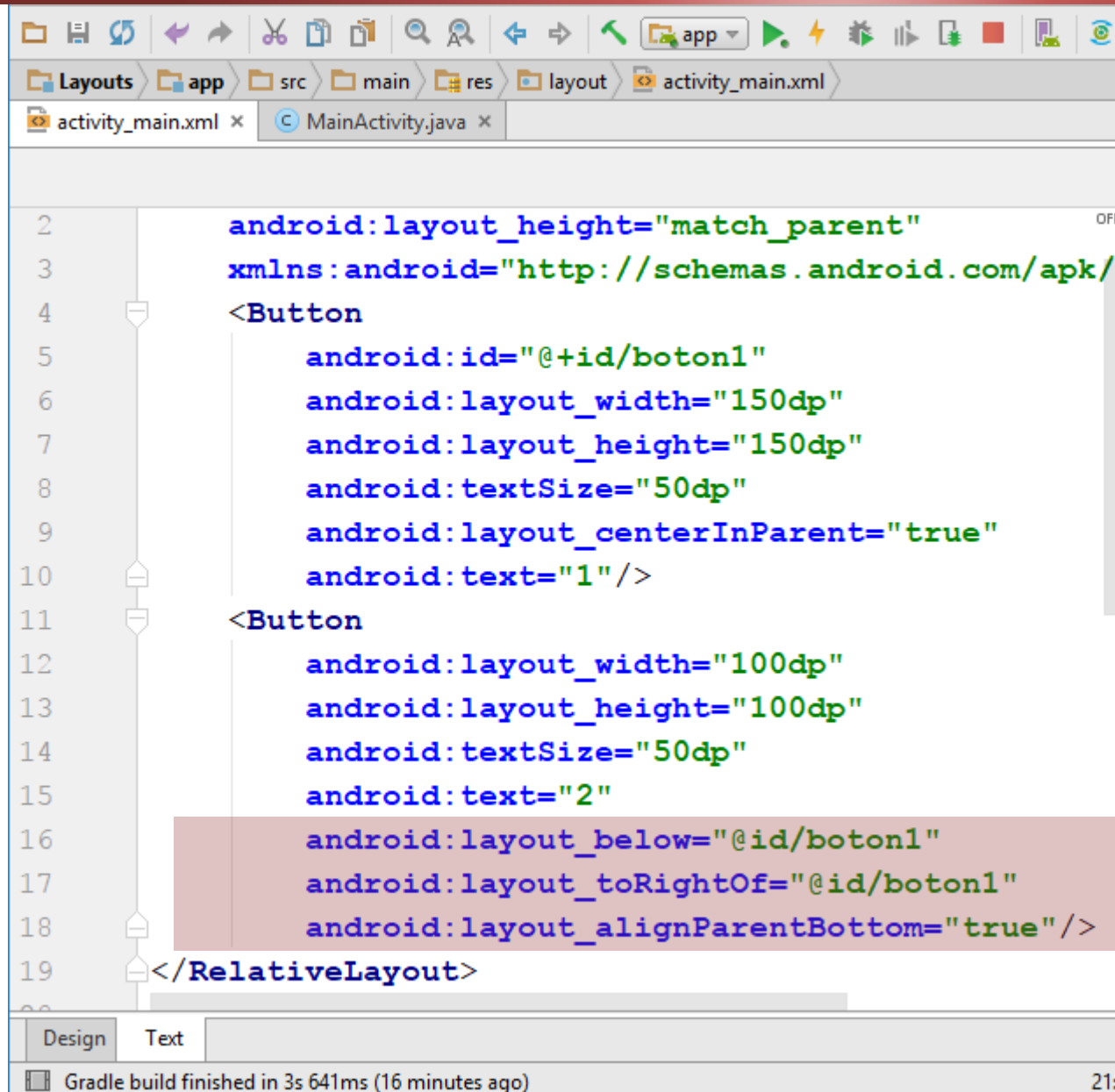
The screenshot displays the Android Studio IDE. On the left, the XML code for a `RelativeLayout` is shown in the 'Text' tab. The code defines two buttons: 'boton1' and '2'. Button '2' is positioned below and to the right of 'boton1' using the `layout_below` and `layout_toRightOf` attributes. The 'Design' tab on the right shows a visual representation of this layout on a mobile device screen. Two gray squares, labeled '1' and '2', are shown. Square '1' is on the left, and square '2' is positioned to its right and below it, with a blue arrow indicating the alignment. The status bar at the top of the preview shows 'Layouts' and a time of 7:00. The bottom status bar indicates 'Gradle build finished in 3s 641ms (13 minutes ago)', the time '21:1', and the encoding 'UTF-8'.

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

Design | Text

Gradle build finished in 3s 641ms (13 minutes ago) 21:1 CRLF UTF-8 Context: <no context>

RelativeLayout

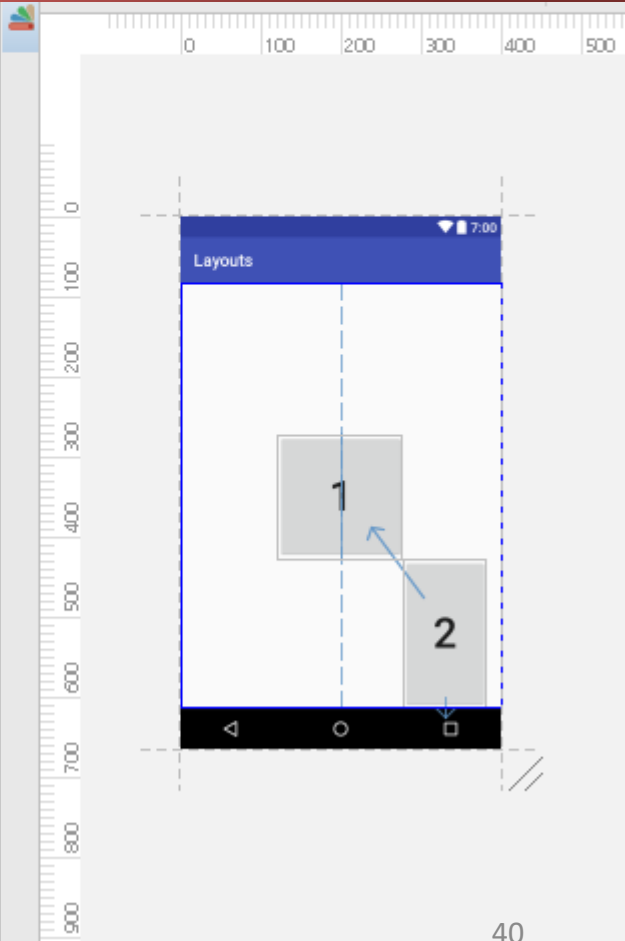


Probar el
resultado de la
siguiente
disposición

RelativeLayout

Se pueden combinar distintas disposiciones para obtener resultados muy variados

```
3 android:layout_height="match_parent"
4 xmlns:android="http://schemas.android.com/apk/
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 <Button
13     android:layout_width="100dp"
14     android:layout_height="100dp"
15     android:textSize="50dp"
16     android:text="2"
17     android:layout_below="@id/boton1"
18     android:layout_toRightOf="@id/boton1"
19     android:layout_alignParentBottom="true"/>
20 </RelativeLayout>
```



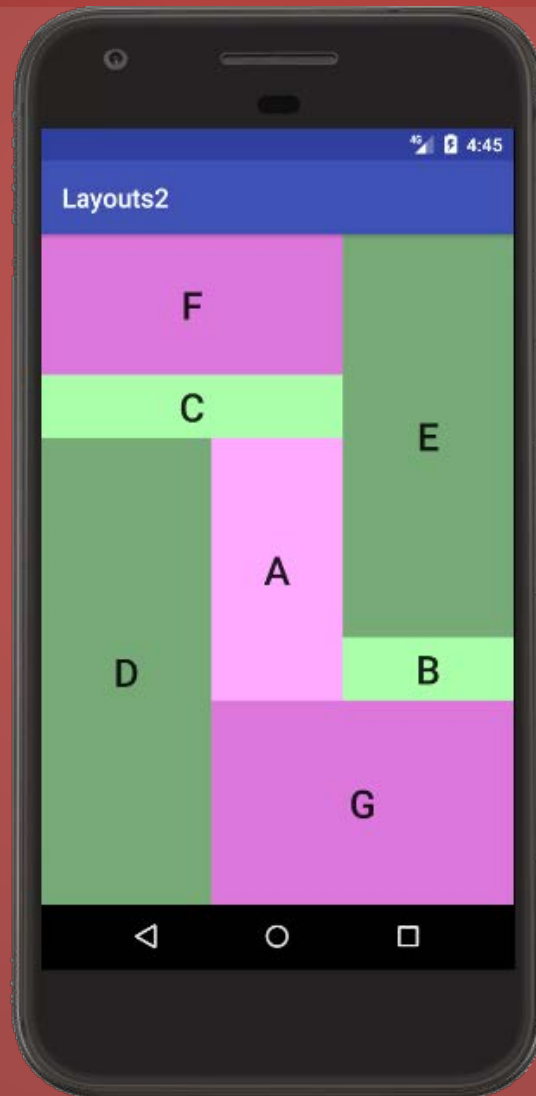
Relacionado con las alineaciones horizontales y teniendo en cuenta los idiomas **RTL** también existen:

- **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

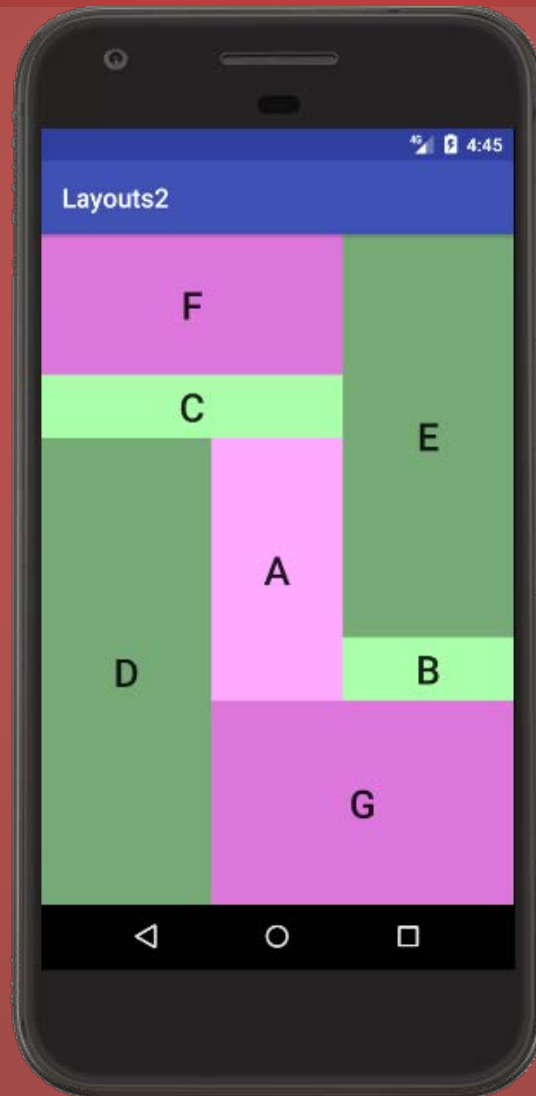
Ejercicio

Diseñar la siguiente interface (los **views** son **Buttons** a los que se les ha cambiado el color del **background**)

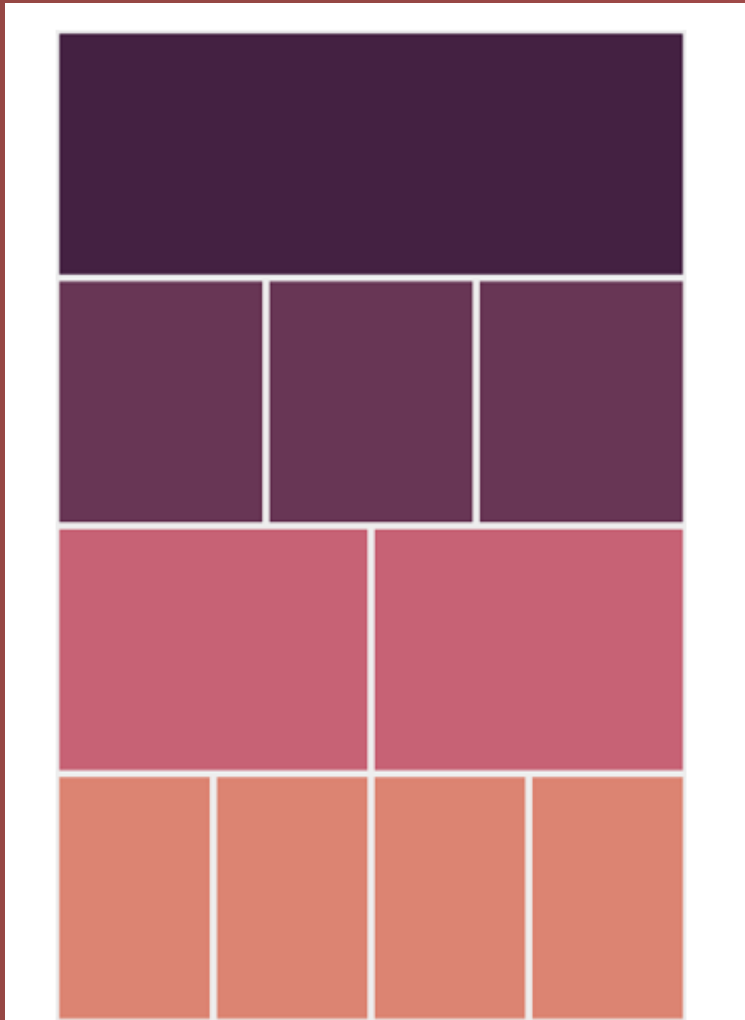


Ejercicio

Diseñar la siguiente interface (los **views** son **Buttons** a los que se les ha cambiado el color del **background**)



TableLayout

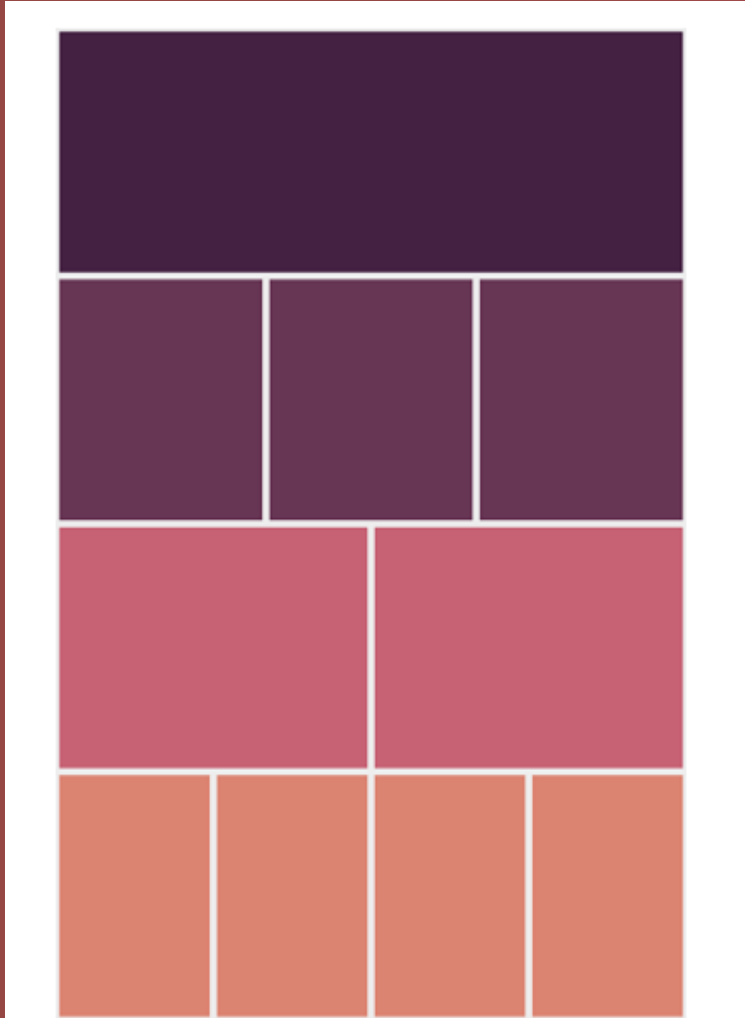


Es un **ViewGroup** que ubica a sus hijos en filas y columnas

Generalmente está compuesto de objetos **TableRow**.

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

TableLayout

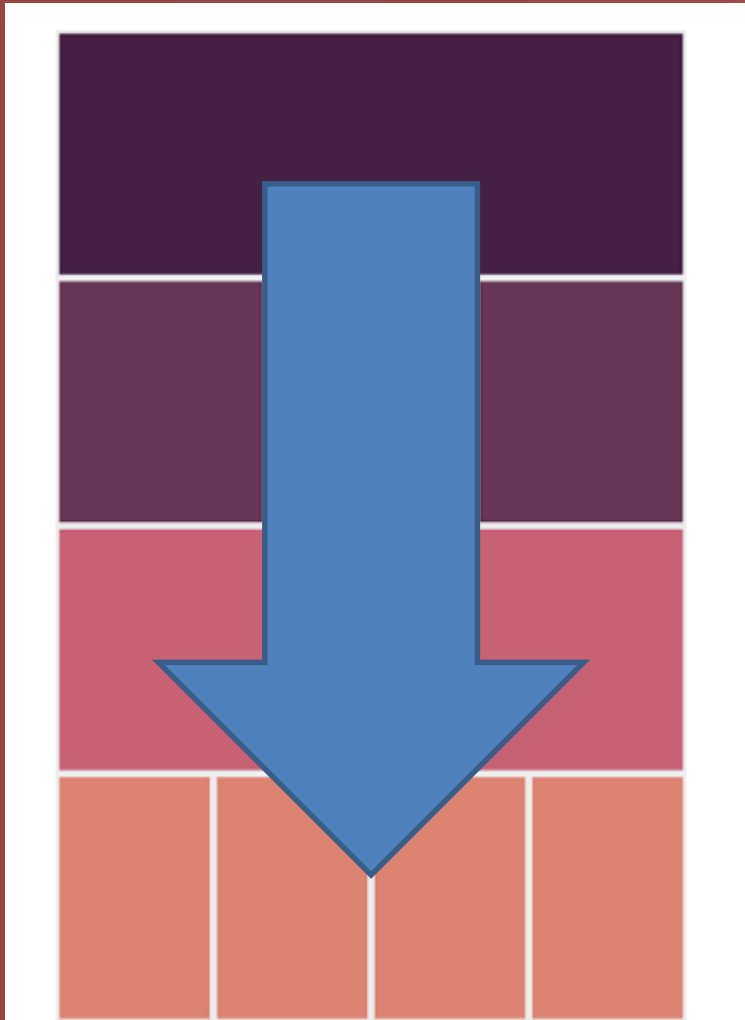


La tabla tendrá tantas columnas como aquel **TableRow** con más celdas.

Una celda podría expandirse ocupando más de una columna.

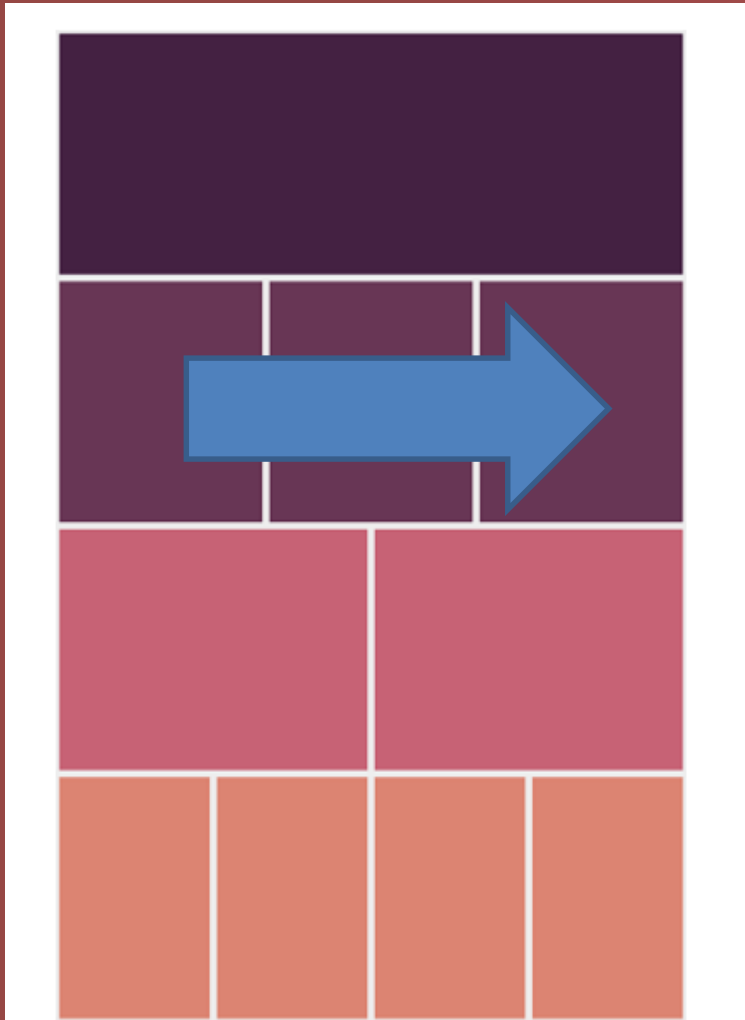
Un **TableLayout** no contempla bordes. Simplemente organiza elementos visuales

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"
            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>
```

Estira las columnas para ocupar todo el ancho del TableLayout



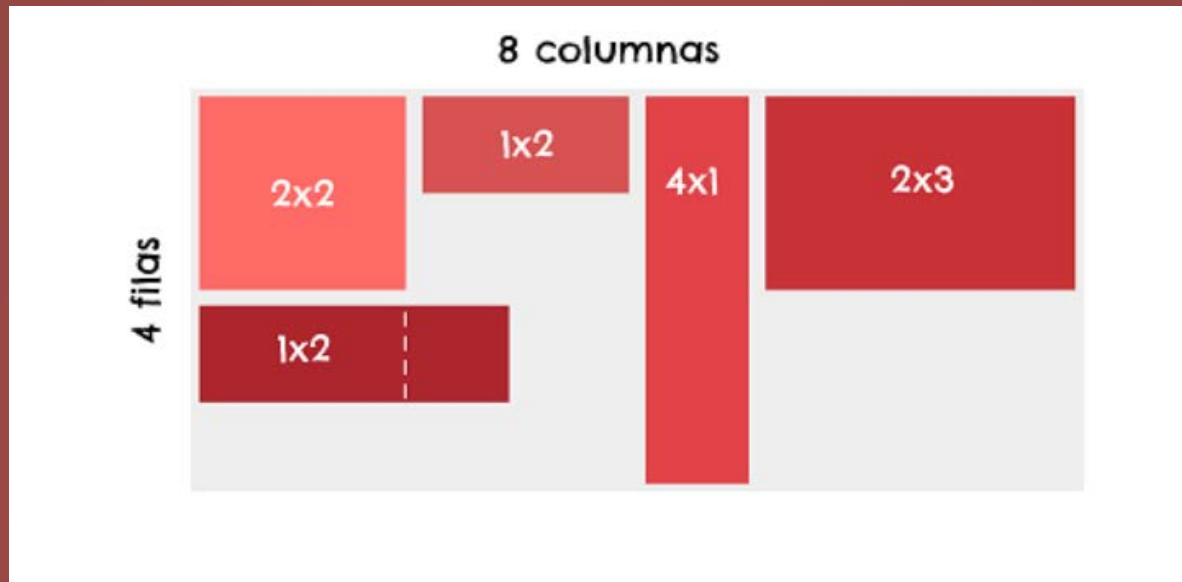
Ejemplo de TableLayout

```
<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

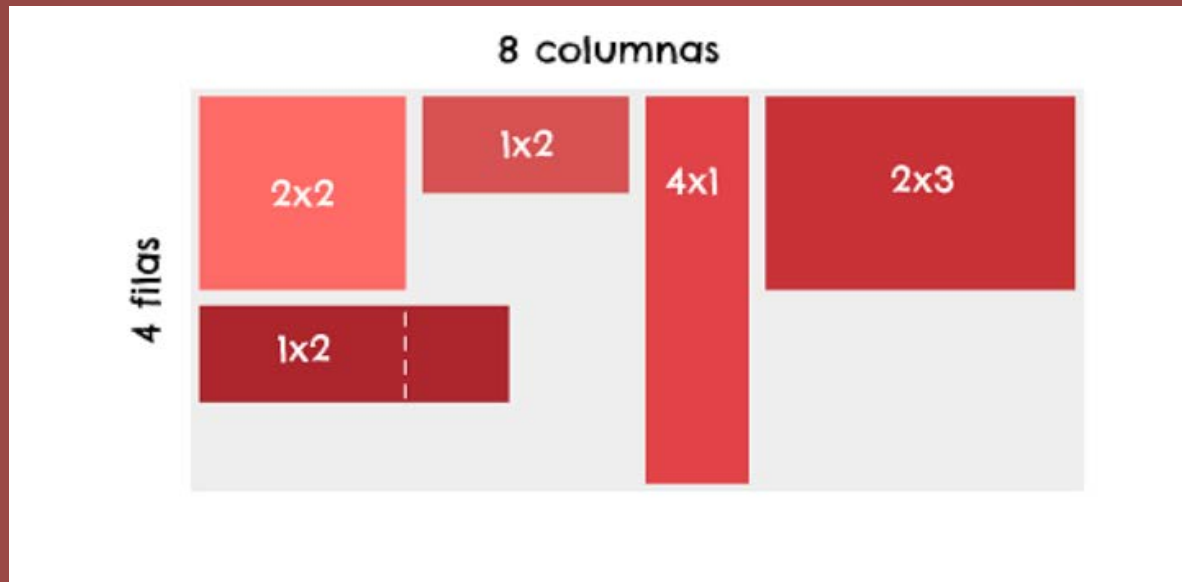


GridLayout



- Incluido a partir de Android 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

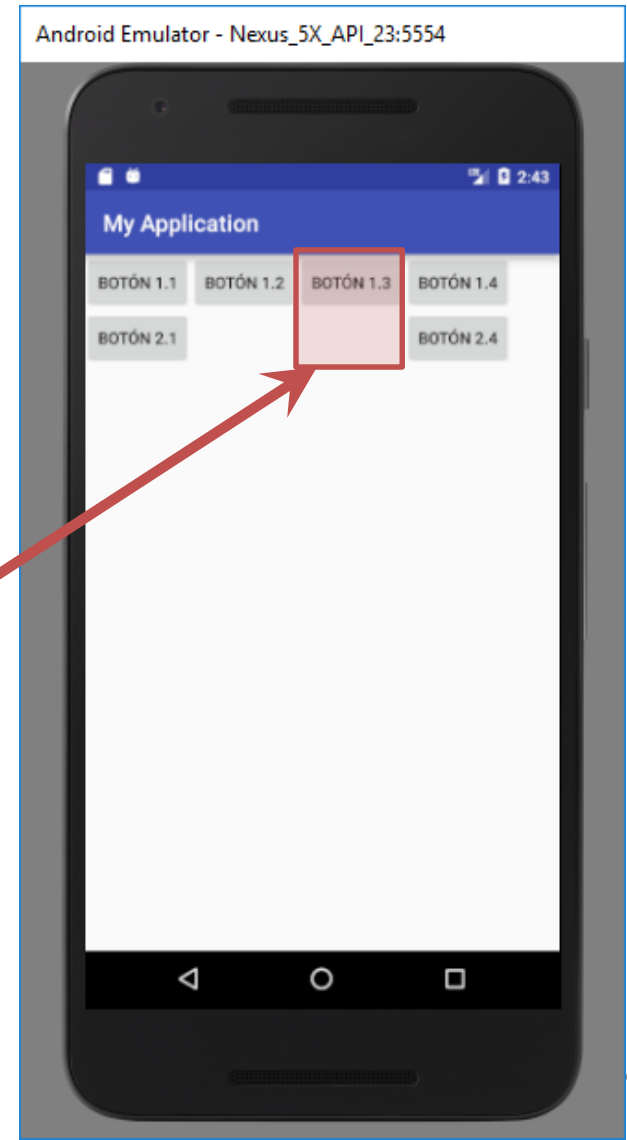
```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="4"
    android:orientation="horizontal" >

    <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

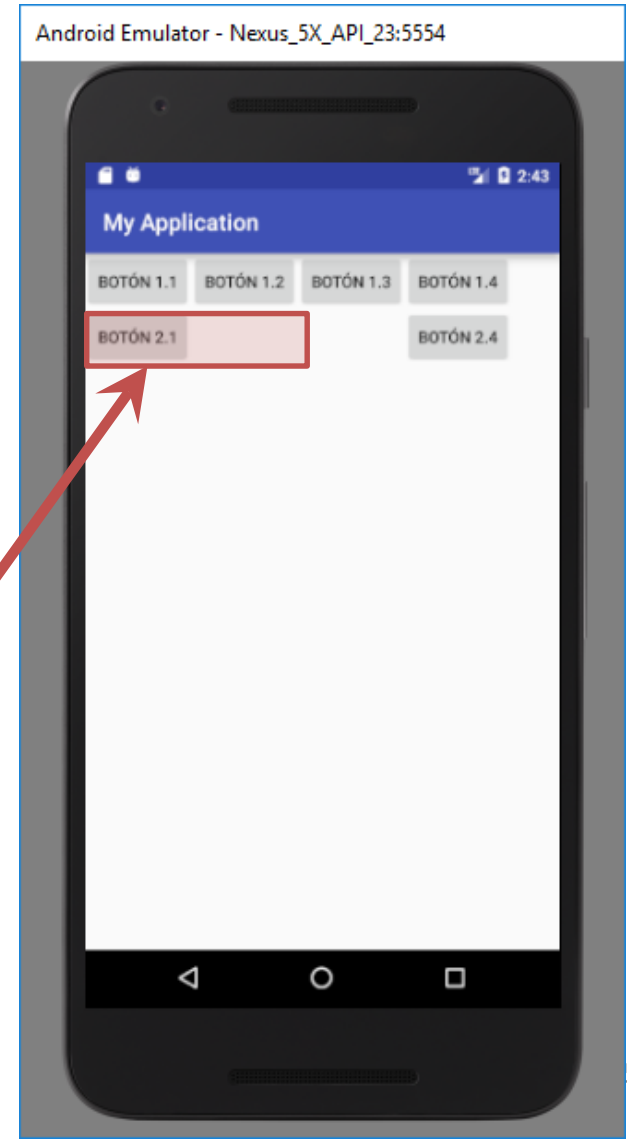
```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="4"
    android:orientation="horizontal" >

    <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

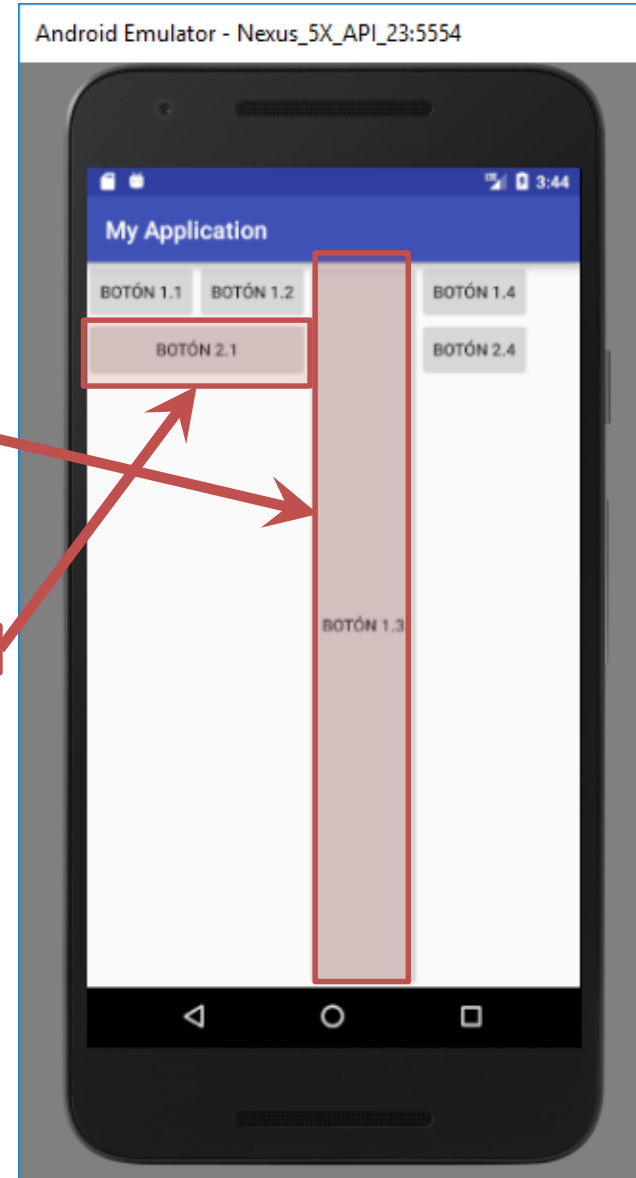
```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="4"
    android:orientation="horizontal" >

    <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

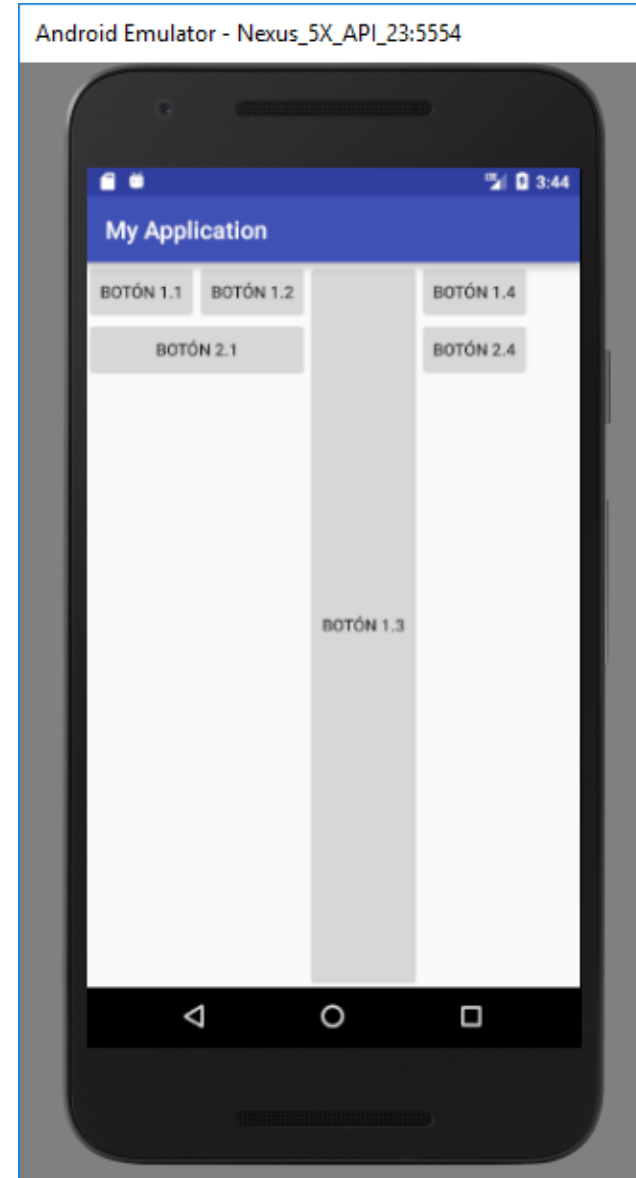
```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="4"
    android:orientation="horizontal" >

    <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?



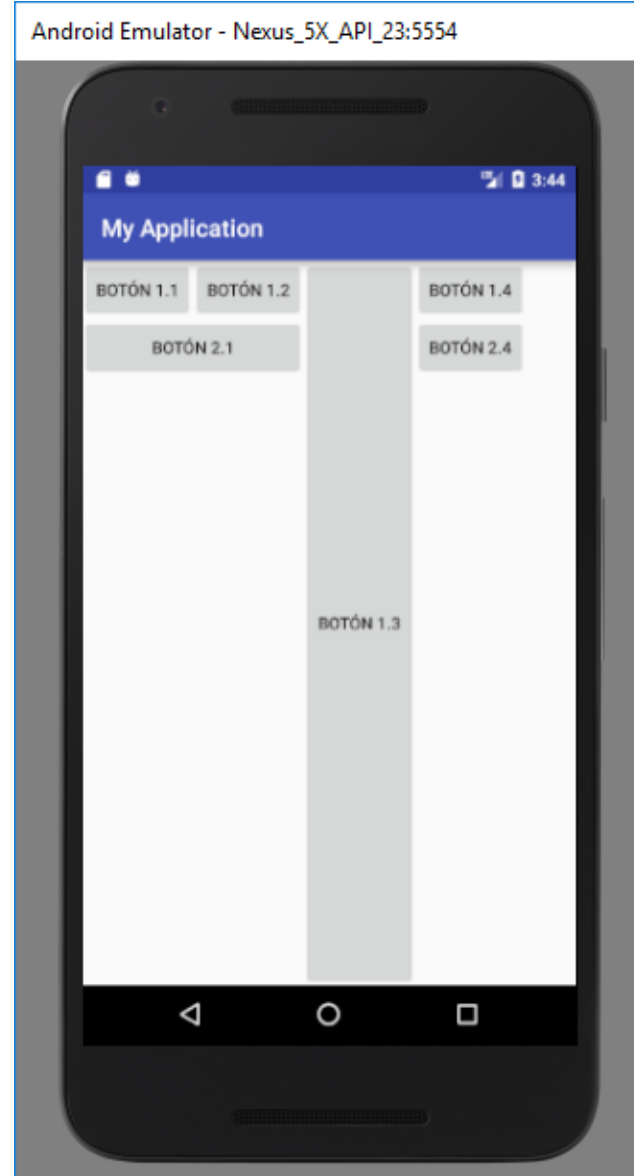
Ejemplo de GridLayout

```
<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

